

# CONTACT TRACING ENHANCES THE EFFICIENCY OF COVID-19 GROUP TESTING

Ritesh Goenka,<sup>\*1</sup> Shu-Jie Cao,<sup>\*2</sup> Chau-Wai Wong,<sup>3</sup> Ajit Rajwade,<sup>1</sup> Dror Baron<sup>3</sup>

<sup>1</sup> IIT Bombay, India, <sup>2</sup> ShanghaiTech University, China, <sup>3</sup> North Carolina State University, USA

## ABSTRACT

Group testing can save testing resources in the context of the ongoing COVID-19 pandemic. In group testing, we are given  $n$  samples, one per individual, and arrange them into  $m < n$  pooled samples, where each pool is obtained by mixing a subset of the  $n$  individual samples. Infected individuals are then identified using a group testing algorithm. In this paper, we use side information (SI) collected from contact tracing (CT) within non-adaptive/single-stage group testing algorithms. We generate data by incorporating CT SI and characteristics of disease spread between individuals. These data are fed into two signal and measurement models for group testing, where numerical results show that our algorithms provide improved sensitivity and specificity. While Nikolopoulos et al. utilized family structure to improve non-adaptive group testing, ours is the first work to explore and demonstrate how CT SI can further improve group testing performance.

**Index Terms**—Contact tracing, non-adaptive group testing, compressed sensing, overlapping group LASSO, generalized approximate message passing (GAMP).

## 1. INTRODUCTION

Widespread testing has been promoted for combating the ongoing COVID-19 pandemic. Samples are typically collected from nasal or oropharyngeal swabs, and then processed by a reverse transcription polymerase chain reaction (RT-PCR) machine. However, widespread testing is hindered by supply chain constraints and long testing times.

Pooled or *group testing* has been suggested for improving testing efficiencies [1]. Group testing involves mixing a subset of  $n$  individual samples into  $m < n$  pools. The measurement process can be expressed as  $\mathbf{y} = \mathfrak{N}(\mathbf{A}\mathbf{x})$ , where  $\mathbf{x}$  is a vector that quantifies the health status of the  $n$  individuals,  $\mathbf{A}$  is an  $m \times n$  binary pooling matrix with  $A_{ij} = 1$  if the  $j$ th individual contributes to the  $i$ th pool, else  $A_{ij} = 0$ ,  $\mathbf{y}$  is a vector of  $m$  noisy measurements or tests, and  $\mathfrak{N}$  represents a probabilistic noise model that relates the noiseless pooled results,  $\mathbf{A}\mathbf{x}$ , to  $\mathbf{y}$ . We consider two signal and noise models.

**Model M1:** A *binary noise* model used by Zhu et al. [2],

where  $\mathbf{x}$  is binary,  $\mathbf{w} = \mathbf{A}\mathbf{x}$  is an auxiliary vector, and the measurement  $y_i \in \{0, 1\}$  depends probabilistically on  $w_i$ , where  $\Pr(y_i = 1|w_i = 0)$  and  $\Pr(y_i = 0|w_i > 0)$  are probabilities of erroneous tests.

**Model M2:** A *multiplicative noise* model of the form  $\mathbf{y} = \mathbf{A}\mathbf{x} \circ \mathbf{z}$  as used in Ghosh et al. [3], where  $\circ$  represents element-wise multiplication,  $\mathbf{z}$  is a vector of  $m$  noisy elements defined as  $z_i = (1+q)^{\eta_i}$ ,  $q \in (0, 1]$  is a known amplification factor for RT-PCR,  $\eta_i \sim \mathcal{N}(0, \sigma^2)$ , and  $\sigma^2 \ll 1$  is a known parameter controlling the strength of the noise in RT-PCR. Under model **M2**,  $\mathbf{x}$  and  $\mathbf{y}$  represent viral loads in the  $n$  individuals and  $m$  pools, respectively. Assuming reasonably high viral loads in  $\mathbf{x}$ , Poisson effects in  $\mathbf{y}$  can be ignored [3].

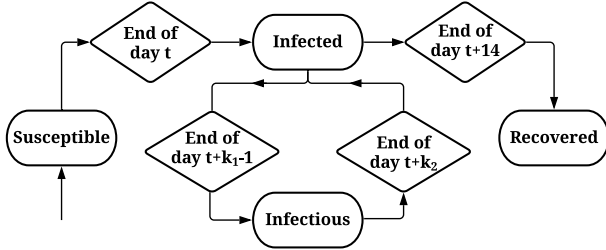
For both models, the objective is to estimate  $\mathbf{x}$  from  $\mathbf{y}$  and  $\mathbf{A}$ . We use single-stage *non-adaptive* algorithms as in [2, 3], rather than two-stage algorithms, which employ a second stage of tests depending on results from the first stage, as in Heidarzadeh and Narayanan [4] or the classical approach by Dorfman [5]. The main advantage of non-adaptive algorithms is that they save on testing time, which is quite high for RT-PCR. Algorithms for estimation of  $\mathbf{x}$  from  $\mathbf{y}$  and  $\mathbf{A}$  [3, 6] rely primarily on the *sparsity* of  $\mathbf{x}$ , which is a valid assumption for COVID-19 due to low prevalence rates [7]. Zhu et al. [2] also exploit probabilistic information such as the prevalence rate, structure in  $\mathbf{x}$ , and *side information* (SI). Finally, Nikolopoulos et al. [8] showed how SI about family-style structure can reduce the number of required tests,  $m$ .

In this paper, we show how to estimate  $\mathbf{x}$  while utilizing *contact tracing* (CT) SI, which allows one to analyze the spread of the pandemic [9]. Our contributions are twofold. First, we propose a generative model for a population of  $n$  individuals that characterizes the spread of COVID-19 by explicitly using CT SI. Second, we show that CT SI, when used appropriately, can help algorithms such as generalized approximate message passing (GAMP) [10] or LASSO variants [11, 12] better estimate  $\mathbf{x}$  from  $\mathbf{y}$  and  $\mathbf{A}$ . Our work uses more SI than Nikolopoulos et al. [8], who only considered family-style structure in binary group testing.

## 2. DATA GENERATION MODEL

In this section, we present a generative infection model incorporating CT SI, which we later use to prepare simulated data for algorithmic evaluation. We model a population of  $n$  individuals using a dynamical or time-varying graphical model that contains nodes  $\{v_i\}_{i=1}^n$  and undirected edges

<sup>\*</sup> RG and SJC have made equal contributions. AR acknowledges support from SERB Grant #10013890, IITB-WRCB Grant #DONWR04-002, and DST-Rakshak grant #DST0000-005. CWW acknowledges support from NSF Grant #2030430. The supplemental document can be found on the webpages of the last three authors.



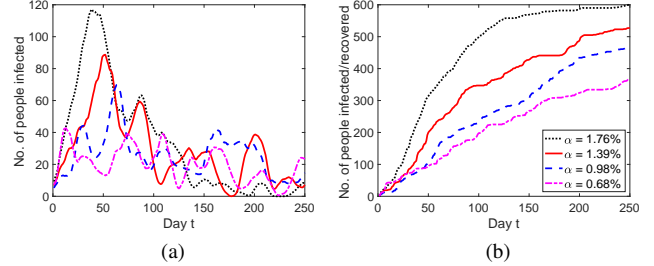
**Fig. 1.** State transition diagram for a node. A node is infectious only between days  $k_1$  and  $k_2$  (both inclusive) after getting infected. In our work, we set  $(k_1, k_2) = (3, 7)$ .

$\{e_{ij}^{(t)}\}_{i,j=1}^n$ . On a given day  $t$ , an edge  $e_{ij}^{(t)}$  between nodes  $v_i$  and  $v_j$  encodes CT SI  $(\tau_{ij}^{(t)}, d_{ij}^{(t)})$ , which can be acquired via Bluetooth-based CT applications [13]. Here,  $\tau_{ij}^{(t)}$  represents the contact duration and  $d_{ij}^{(t)}$  represents a measure of the physical proximity between two individuals. On day  $t$ , a node can be in one of the following states: *susceptible*, *infected*, *infectious*, and *recovered*. To keep the model simple, we assume that there are no reinfections, i.e., recovered is a terminal state, despite some reports of reinfection [14].

We adopt a simplified infection dynamic wherein the infectious period is preceded and followed by the infected state. We propose the following design parameters for the infection dynamics based on a World Health Organization report on COVID-19 [15]. Specifically, a node  $v_i$  remains infected but noninfectious for  $k_1 = 3$  days. On day  $t + k_1$ , the node becomes infectious and may transmit the disease to a susceptible neighboring node  $v_j$  with probability  $p_{i,j}^{(t+k_1)}$  whose construction is described below. An infectious node can potentially transmit the infection until  $k_2 = 7$  days after getting infected, and becomes noninfectious afterward. We also model a small fraction of stray infections that may occur, for example, due to sporadic contact with contaminated surfaces. Such infections only affect nodes in the susceptible state with a probability  $p_1 = 2 \times 10^{-4}$  of our choice. A state diagram for a general node is illustrated in Fig. 1. Regarding the viral load  $x_i^{(t)}$  for node  $i$  on day  $t$ , we assume  $x_i^{(t)} = 0$  if the node is susceptible or recovered. For an infected or infectious node, we make a simplified assumption that its viral load  $x_i^{(t)} \sim \text{Uniform}(1, 32768)$ ,<sup>1</sup> once drawn, remains constant throughout the combined 14-day period of infection.

Next, we model the probability  $p_{i,j}^{(t)}$  that the disease is transmitted from node  $v_i$  to  $v_j$  on day  $t$ . We view the infection times of the population throughout the pandemic as a nonhomogeneous Poisson process with time-varying rate function  $\lambda(t)$ . Consider a  $\tau_{ij}^{(t)}$ -hour contact on day  $t$  when susceptible node  $v_j$  is exposed to infectious node  $v_i$ . The average infection rate  $\lambda_{ij}(t)$  for day  $t$  is assumed to be proportional to both the viral load  $x_i^{(t)}$  and the physical proximity  $d_{ij}^{(t)}$ , namely,

<sup>1</sup>The cycle threshold for RT-PCR commonly ranges from 19 to 34 cycles [16, Fig. 3], where 34 cycles corresponds to a low initial viral load of a few molecules, and each cycle roughly doubles the viral density. Therefore, we estimate the largest possible viral load as  $2^{34-19} = 2^{15} = 32768$ .



**Fig. 2.** (a) The number of active infections, and (b) cumulative infections at different inter-clique contact levels  $\alpha$ . We chose 50-day windows for testing proposed algorithms.

$\lambda_{ij}(t) = \lambda_0 x_i^{(t)} d_{ij}^{(t)}$ , where  $\lambda_0$  is a tunable, baseline Poisson rate. The probability that  $v_j$  is infected by the end of contact period  $\tau_{ij}^{(t)}$  is therefore  $p_{i,j}^{(t)} = 1 - \exp(-\lambda_0 x_i^{(t)} d_{ij}^{(t)} \tau_{ij}^{(t)})$  for  $t \in [k_1, k_2] + t_i$ . From the standpoint of susceptible node  $v_j$ , all its neighbors  $v_k$  that are infectious contribute to its probability of getting infected on day  $t$ , namely,  $1 - \prod_k (1 - p_{k,j}^{(t)})$ .

While generating our data, we considered  $n = 1000$  nodes divided into cliques based on the distribution of family sizes in India [17, pg. 18], for a duration of  $t_{\max} = 250$  days. Fig. 2 shows the number of active infections and the cumulative number of infections at the end of each day. The clique structures were kept constant throughout the  $t_{\max}$  days, whereas inter-clique contacts corresponding to sporadic contacts between people were dynamically added and removed. The varying inter-clique contact level  $\alpha$  affects the sparsity of the underlying vector  $\mathbf{x}$  as it brings infections to new cliques/families. Pooling of samples is performed at the beginning of each day from  $t_{\text{peak}} - 24$  to  $t_{\text{peak}} + 25$ , where  $t_{\text{peak}}$  is the day with the maximum number of active infections.

### 3. PROPOSED GROUP TESTING ALGORITHMS

This section describes two classes of group testing algorithms for reconstructing the health status vector  $\mathbf{x}$  from the pooled tests,  $\mathbf{y}$ , and pooling matrix,  $\mathbf{A}$ .

**Algorithms for binary noise.** For model **M1**, Zhu et al. [2] use *generalized approximate message passing* (GAMP) [10] for group testing estimation. GAMP is comprised of two components. The first component consists of an input channel that relates a prior for  $n$  individuals' viral loads,  $\mathbf{x} = (x_i)_{i=1}^n$ , and pseudo data,  $\mathbf{v} = \mathbf{x} + \mathbf{q} \in \mathbb{R}^n$ , where the  $n$  coordinates of  $\mathbf{x}$  are correlated, and  $\mathbf{q}$  is additive white Gaussian noise with  $q_i \sim \mathcal{N}(0, \Delta)$ . We estimate  $\mathbf{x}$  from  $\mathbf{v}$  using a denoising function, often called a denoiser,

$$\hat{x}_i = g_{\text{in}}(\mathbf{v}) = \mathbb{E}[X_i | \mathbf{V} = \mathbf{v}], \quad (1)$$

where we use the convention that when both the upper and lower case versions of a symbol appear, the upper case is a random variable and the lower case its realization, and  $\mathbb{E}[X_i | \mathbf{v}]$  represents  $\mathbb{E}[X_i | \mathbf{V} = \mathbf{v}]$  when the context is clear. The second component of GAMP consists of an output channel relating the auxiliary vector  $\mathbf{w}$  to the noisy measurements  $\mathbf{y}$  as reviewed in Sec. 1. We

adopt the output channel denoiser of Zhu et al. [2],  $h_i = g_{\text{out}}(y_i; k_i, \theta_i) = (\mathbb{E}[W_i | y_i, k_i, \theta_i] - k_i)/\theta_i$ , where  $\theta_i$  is the estimated variance of  $h_i$ , and  $k_i$  is the mean of our estimate for  $w_i$ . Since  $y_i$  depends probabilistically on  $w_i$ , we have  $f(w_i | y_i, k_i, \theta_i) \propto \Pr(y_i | w_i) \exp\left[-\frac{(w_i - k_i)^2}{2\theta_i}\right]$ , where  $W_i$  is approximated as Gaussian in the derivation of GAMP.

While Zhu et al. [2] considered Bernoulli  $\mathbf{x}$ , which implies a scalar separable denoiser  $g_{\text{in}}$  for the input channel, this paper accounts for probabilistic dependencies within  $\mathbf{x}$ . Our first probabilistic model considers groups of people, for example, members of a family. Each family is modeled as entirely healthy with probability  $1 - \pi_{\text{vf}}$ , else each individual within the family is infected with probability  $\pi_{\text{ind}}$ . This model relates to our generative model of Sec. 2 by using family structure as SI. Denoting the pseudo data of family  $\mathcal{F}$  by  $\mathbf{v}_{\mathcal{F}}$ , the denoiser for the  $i$ th individual of family  $\mathcal{F}$  is given by

$$g_{\text{in}}^{\text{family}}(\mathbf{v}_{\mathcal{F}}) = \mathbb{E}[X_i | \mathcal{F} \text{ viral}, \mathbf{v}_{\mathcal{F}}] \Pr(\mathcal{F} \text{ viral} | \mathbf{v}_{\mathcal{F}}), \quad (2)$$

where  $\mathbb{E}[X_i | \mathcal{F} \text{ viral}, \mathbf{v}_{\mathcal{F}}]$  and  $\Pr(\mathcal{F} \text{ viral} | \mathbf{v}_{\mathcal{F}})$  are parameterized by  $\pi_{\text{vf}}$ ,  $\pi_{\text{ind}}$ , and  $\Delta$ . For detailed expressions, we refer readers to Sec. 1.1 of the supplemental document.

Our second probabilistic model uses CT. Consider a hypothetical widespread testing program that relies on CT SI, where all individuals are tested 8 days before the group testing program begins resulting in a good estimate of their ground-truth health status. After the program begins, probability estimates from the previous group test are used as priors for the  $n$  individuals when performing the current group test. We provide detailed analysis in Secs. 2.2–2.3 of the supplemental document on the use of prior infection status. The final form of the denoiser for the CT model is as follows:

$$g_{\text{in}}^{\text{CT}}(v_i) = \left\{ 1 + [\Pr(X_i = 1)^{-1} - 1] \exp\left[\left(v_i - \frac{1}{2}\right)/\Delta\right] \right\}^{-1}. \quad (3)$$

Here,  $\Pr(X_i = 1)$  for day  $k + 1$  can be estimated by aggregating CT information of individual  $i$  over the past 8 days, namely,  $\widehat{\Pr}^{(k+1)}(X_i = 1) = 1 - \prod_{d=k-7}^k \prod_{j=1}^n (1 - \widehat{p}_{i,j}^{(d)})$ , where  $\widehat{p}_{i,j}^{(d)}$  is the estimated probability of infection of individual  $i$  due to contact with individual  $j$ . This probability,  $\widehat{p}_{i,j}^{(d)}$ , can be determined by the CT information  $(\tau_{ij}^{(d)}, d_{ij}^{(d)})$ , as well as their infection status as follows:

$$\widehat{p}_{i,j}^{(d)} = \exp\left(-(\lambda \tau_{ij}^{(d)} d_{ij}^{(d)} \Psi_{ij}^{(d)} + \epsilon)^{-1}\right), \quad (4)$$

where  $\Psi_{ij}^{(d)} = 1 - \widehat{\Pr}^{(d)}(X_i = 0) \widehat{\Pr}^{(d)}(X_j = 0)$ ,  $\lambda$  is a Poisson rate parameter, and  $\epsilon$  is used to avoid division by zero. Note that  $\widehat{p}_{i,j}^{(d)}$  depends on  $\lambda$ , which is unknown in practice. We estimate it using a plug-in approach by Ma et al. [18]. More details are given in Sec. 1.2 of the supplemental document.

**Algorithms for multiplicative noise.** For model **M2**, recall that  $\mathbf{x}$  and  $\mathbf{y}$  represent viral loads of individual samples and pools, respectively. The core algorithm presented in [3] uses the well-known LASSO estimator,  $\widehat{\mathbf{x}}^{\text{LASSO}} = \arg \min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \rho \|\mathbf{x}\|_1$  [19], where  $\rho$  is a smoothness parameter. LASSO exploits the sparsity of  $\mathbf{x}$  but uses no SI. Despite the multiplicative nature of the noise, LASSO

yields good estimation performance [3] in terms of three commonly used measures: (i) *relative root mean squared error* (RRMSE) =  $\|\mathbf{x} - \widehat{\mathbf{x}}\|_2 / \|\mathbf{x}\|_2$ ; (ii) *false negative rate* (FNR) = #incorrectly detected negatives / #true positives; and (iii) *false positive rate* (FPR) = #incorrectly detected positives / #true negatives. Note that FNR = 1 – sensitivity and FPR = 1 – specificity.

In some cases, the  $n$  individuals in  $\mathbf{x}$  can be partitioned into  $n_1 \ll n$  disjoint groups of people, for example family members, who interact closely with each other and are thus likely to pass the virus between group members. This family-style structure leads to a situation where either all members of the group are uninfected, or a majority of members are infected. Note that the family-style structure also includes groups of coworkers, students taking a course together, and people sharing common accommodation. If reliable SI about how the  $n$  individuals are partitioned into families is available, and only a small portion of families,  $n_2 \ll n_1$ , are infected, then LASSO can be replaced by *group square-root LASSO* (SQRT-GLASSO) [11].<sup>2</sup> The latter is defined as

$$\widehat{\mathbf{x}}^{\text{SQRT-GLASSO}} = \arg \min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 + \rho \sum_{g=1}^{n_1} \|\mathbf{x}_g\|_2, \quad (5)$$

where  $\mathbf{x}_g$  consists of viral loads of people from the  $g$ th family.

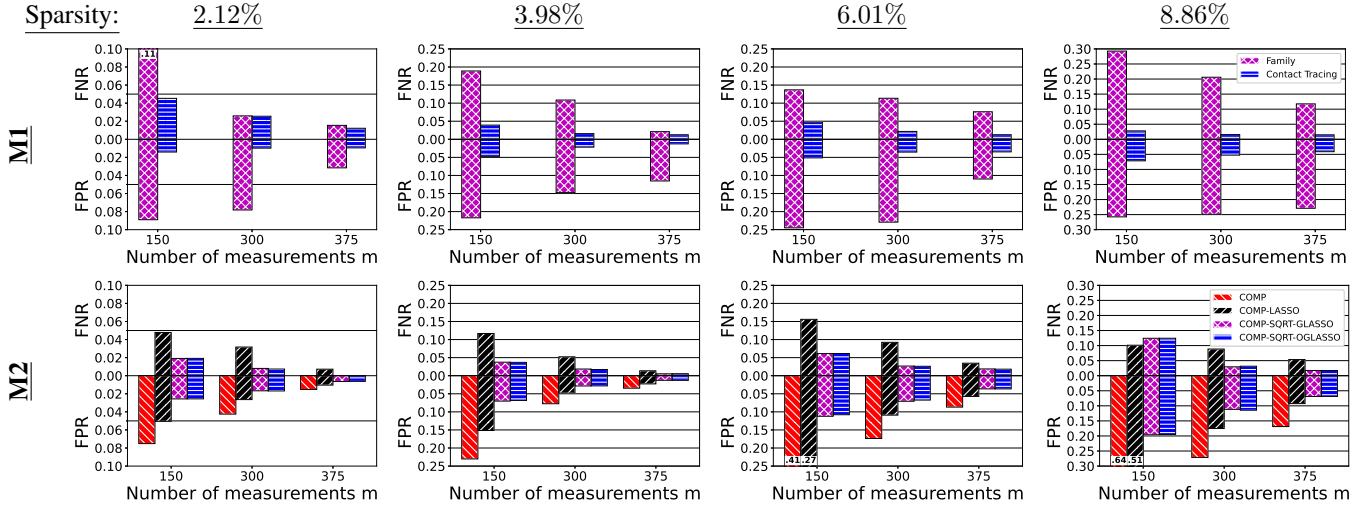
In other cases, accurate family SI may be unavailable or unreliable. Moreover, family SI does not account for contact between members of different families. In such cases, we use CT SI commonly available via Bluetooth [13] to directly *infer* family-style structure using clique detection algorithms; contacts between members of different families can also be considered to be small cliques. In particular, we use the Bron–Kerbosch algorithm [21] to find maximal cliques in the CT graph, and label each clique as a family. Note that one could generate these groups differently [22, Sec. 7], for example, decomposition into  $k$ -clique communities [23]. However, such a decomposition may partition the  $n$  individuals into  $n_3 \ll n$  family structures that *overlap* with each other, unlike the earlier case of *disjoint* families. In a scenario with overlapping families, we use the *overlapping group square-root LASSO* (COMP-SQRT-OGGLASSO) estimator [22],

$$\widehat{\mathbf{x}}^{\text{SQRT-OGGLASSO}} = \arg \min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 + \rho \Omega_{\text{overlap}}(\mathbf{x}), \quad (6)$$

where  $\Omega_{\text{overlap}}(\mathbf{x}) = \inf_{\mathbf{v} \in \mathcal{V}_G, \sum_{g \in G} \mathbf{v}_g = \mathbf{x}} \sum_{g \in G} \|\mathbf{v}_g\|_2$ ,  $G$  denotes a set of possibly overlapping groups each containing a subset of the  $n$  individuals in  $\mathbf{x}$ ,  $\mathcal{V}_G$  is a set of  $|G|$ -tuples of vectors  $\mathbf{v} = (\mathbf{v}_g)_{g \in G}$ ,  $g$  is an index for the groups in  $G$ , and  $\mathbf{v}_g \in \mathbb{R}^n$  is a vector whose support is a subset of  $g$ . Advantages of OGLASSO over GLASSO for overlapping groups are summarized in Fig. 1 and Sec. 3 of [22].

In all three algorithms, we imposed a non-negativity constraint on  $\mathbf{x}$ . Moreover, all algorithms were preceded by a step that executed *combinatorial orthogonal matching pursuit* (COMP). COMP declares all samples that contributed to a pool

<sup>2</sup>We observed that SQRT-GLASSO, which has an  $\ell_2$  data fidelity term instead of a squared  $\ell_2$  one [20], outperformed GLASSO. In contrast, conventional LASSO outperformed SQRT-LASSO.



**Fig. 3.** Performance of the proposed group testing methods **M1** (top row) with binary noise and **M2** (bottom row) with multiplicative noise at four averaged sparsity levels and three measurement levels for a population of  $n = 1000$  individuals.

with zero viral load to be negative. This COMP preprocessing step reduces the problem size and improves all three algorithms' performance, as well as speed. We henceforth refer to our algorithms as COMP-LASSO, COMP-SQRT-GLASSO and COMP-SQRT-OGGLASSO.

#### 4. NUMERICAL RESULTS

We now present numerical results obtained for the two models, **M1** and **M2**. The data was generated based on Sec. 2, and group testing inference was performed using the algorithms proposed in Sec. 3. We generated datasets using four levels of cross-clique contacts, leading to four averaged sparsity levels, 2.12%, 3.98%, 6.01%, and 8.86%, for  $\mathbf{x}$ . At each sparsity level, we perform pooling experiments using Kirkman triple matrices as proposed in [3] for three measurement setups,  $m \in \{150, 300, 375\}$ . Measurement vectors  $\mathbf{y}$  for **M1** were generated using probabilities for erroneous binary tests,  $\Pr(y_i = 1|w_i = 0) = 0.001$  and  $\Pr(y_i = 0|w_i > 0) = 0.02$ , per Hanel and Thurner [24]. Vectors  $\mathbf{y}$  for **M2** were generated by setting the parameter reflecting the strength of noise in RT-PCR to  $\sigma^2 = 0.01$ . Fig. 3 shows the performance of the proposed algorithms in terms of FNR and FPR<sup>3</sup> averaged across the inference results obtained for the time window of 50 days described in Sec. 2.

For model **M1**, we tested the family denoiser (2) and the CT denoiser (3). Fig. 3 shows that the CT denoiser outperforms the family denoiser in all settings. Both algorithms yield lower (better) FNR and FPR as the number of measurements,  $m$ , increases. Moreover, the CT denoiser's error rates are below 0.05, except for the challenging cases where the sparsity level is 8.86% and  $m \in \{150, 300\}$ .

<sup>3</sup>For **M1**, we chose to report the FPR and FNR pair such that the sum of the two error rates is minimized. The complete ROC curves are shown in Sec. 2.1 of the supplemental document. For **M2**, we reported the error rates by thresholding the estimated viral load using  $\tau = 0.2$ . We noticed that the error rates do not change much when  $\tau$  varies between 0 and 1.

For model **M2**, we tested four algorithms: COMP, COMP-LASSO, COMP-SQRT-GLASSO, and COMP-SQRT-OGGLASSO. The results show that both COMP-SQRT-GLASSO and COMP-SQRT-OGGLASSO outperform COMP-LASSO in terms of FNR and FPR, which shows the benefit of using CT SI. Note that COMP-SQRT-OGGLASSO performs on par with COMP-SQRT-GLASSO, even though the former infers everything on the fly from CT SI without explicit access to family SI. The COMP algorithm by itself produces no false negatives (corresponding to FNR = 0), but many false positives. Further, all four algorithms yield lower (better) FNR and FPR as  $m$  increases or the averaged sparsity level decreases. Finally, we remark that COMP-SQRT-OGGLASSO outperforms COMP-SQRT-GLASSO for more general CT graphs consisting of slightly incomplete cliques. We refer the readers to Sec. 3 of the supplemental document for details.

Our algorithms, when presented with SI, reduce the FNR and FPR, despite not knowing which individuals are infected within each infected family (around 70% of the individuals are infected within an infected family on average). Note that none of the algorithms for model **M2** make use of previous inference results, whereas the CT denoiser for model **M1** uses such information. This distinction makes the two approaches applicable in different scenarios, namely, the CT denoiser can be used for a CT and testing program where the same population is tested at regular intervals, e.g., warehouse employees, whereas COMP-SQRT-OGGLASSO is useful when a population has to be tested only once. Furthermore, while model **M1** performs well in the presence of erroneous binary tests, it does not yield viral load estimates as COMP-SQRT-OGGLASSO does. Viral load estimates could prove to be useful, since there is a positive correlation between mortality and viral loads [25, 26].

**Acknowledgment:** The authors thank Junan Zhu for allowing them to use his implementation of GAMP with SI in their implementation of the family and CT denoisers.

## 5. REFERENCES

- [1] B. Abdalhamid *et al.*, “Assessment of specimen pooling to conserve SARS CoV-2 testing resources,” *American Journal of Clinical Pathology*, vol. 153, 2020.
- [2] J. Zhu, K. Rivera, and D. Baron, “Noisy pooled PCR for virus testing,” <https://arxiv.org/abs/2004.02689>, 2020.
- [3] S. Ghosh, R. Agarwal, M. A. Rehan, S. Pathak, P. Agrawal, Y. Gupta, S. Consul, N. Gupta, R. Goyal, A. Rajwade, and M. Gopalkrishnan, “A compressed sensing approach to group-testing for COVID-19 detection,” <https://arxiv.org/abs/2005.07895>, 2020.
- [4] A. Heidarzadeh and K. Narayanan, “Two-stage adaptive pooling with RT-qPCR for COVID-19 screening,” <https://tinyurl.com/y2d8lt9m>, 2020.
- [5] R. Dorfman, “The detection of defective members of large populations,” *Annals of Mathematical Statistics*, vol. 14, no. 4, p. 436–440, 1943.
- [6] N. Shental *et al.*, “Efficient high throughput SARS-CoV-2 testing to detect asymptomatic carriers,” <https://tinyurl.com/ydctgdy>, 2020.
- [7] D. Benatia, R. Godefroy, and J. Lewis, “Estimating COVID-19 prevalence in the united states: A sample selection model approach,” <https://doi.org/10.1101/2020.04.20.20072942>.
- [8] P. Nikolopoulos, T. Guo, C. Fragouli, and S. Diggavi, “Community aware group testing,” <https://arxiv.org/abs/2007.08111>.
- [9] Center for Disease Control and Prevention, “Contact tracing for COVID-19,” <https://tinyurl.com/y9zdlhv>.
- [10] S. Rangan, “Generalized approximate message passing for estimation with random linear mixing,” in *IEEE Int. Symp. Inf. Theory*, 2011, pp. 2168–2172.
- [11] M. Yuan and Y. Lin, “Model selection and estimation in regression with grouped variables,” *J. Royal Stat. Soc. Series B*, vol. 68, no. 1, 2006.
- [12] D. Jacot, G. Greub, K. Jaton, and O. Opota, “Viral load of SARS-CoV-2 across patients and compared to other respiratory viruses,” *Microbes and Infection*, 2020.
- [13] A. Hekmati, G. Ramachandran, and B. Krishnamachari, “CONTAIN: Privacy-oriented contact tracing protocols for epidemics,” <https://arxiv.org/abs/2004.05251>.
- [14] W. Haseltine, “What COVID-19 reinfection means for vaccines,” <https://tinyurl.com/y6cq2mhj>.
- [15] World Health Organization, “Coronavirus disease 2019 (COVID-19) situation report–73,” <https://tinyurl.com/ybnbky8m>.
- [16] B. Buchan and *et al.*, “Distribution of SARS-CoV-2 PCR cycle threshold values provide practical insight into overall and target-specific sensitivity among symptomatic patients,” *American Journal of Clinical Pathology*, 2020.
- [17] United Nations, “Household size and composition around the world 2017,” <https://tinyurl.com/vo7hrlv>.
- [18] Y. Ma, J. Tan, N. Krishnan, and D. Baron, “Empirical Bayes and full Bayes for signal estimation,” in *Inf. Theory App. Workshop*, San Diego, CA, Feb. 2014, pp. 994–1001.
- [19] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical Learning with Sparsity: The LASSO and Generalizations*. CRC Press, 2015.
- [20] A. Belloni, V. Chernuzhukov, and L. Wang, “Square-root LASSO: Pivotal recovery of sparse signals via conic programming,” *Biometrika*, vol. 98, no. 4, pp. 791–806, 2011.
- [21] C. Bron and J. Kerbosch, “Algorithm 457: Finding all cliques of an undirected graph,” *Commun. ACM*, vol. 16, no. 9, p. 575–577, 1973. [Online]. Available: <https://doi.org/10.1145/362342.362367>
- [22] L. Jacob, G. Obozinski, and J.-P. Vert, “Group LASSO with overlap and graph LASSO,” in *Int. Conf. Mach. Learning*, 2009.
- [23] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, “Uncovering the overlapping community structure of complex networks in nature and society,” *Nature*, vol. 435, pp. 814–818, 2005.
- [24] R. Hanel and S. Thurner, “Boosting test-efficiency by pooled testing strategies for SARS-CoV-2,” *arXiv preprint arXiv:2003.09944*, 2020.
- [25] L. F. Westblade, G. Brar, L. C. Pinheiro, D. Paidoussis, M. Rajan, P. Martin, P. Goyal, J. L. Sepulveda, L. Zhang, G. George, D. Liu, S. Whittier, M. Plate, C. B. Small, J. H. Rand, M. M. Cushing, T. J. Walsh, J. Cooke, M. M. Safford, M. Loda, and M. J. Satlin, “SARS-CoV-2 viral load predicts mortality in patients with and without cancer who are hospitalized with COVID-19,” *Cancer Cell*, 2020. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1535610820304815>
- [26] E. Pujadas, F. Chaudhry, R. McBride, F. Richter, S. Zhao, A. Wajnberg, G. Nadkarni, B. S. Glicksberg, J. Houldsworth, and C. Cordon-Cardo, “SARS-CoV-2 viral load predicts COVID-19 mortality,” *Lancet Respiratory Medicine*, vol. 8, no. 9, p. e70, Sep. 2020. [Online]. Available: [https://doi.org/10.1016/S2213-2600\(20\)30354-4](https://doi.org/10.1016/S2213-2600(20)30354-4)
- [27] D. Chicco and G. Jurman, “The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation,” *BMC Genomics*, vol. 21, 2020.

## 1. ALGORITHMIC DETAILS OF M1

### 1.1. Family Denoiser

We now formalize a family-based infection mechanism that can be used in designing group testing algorithms for improving the detection accuracy. We define  $\mathcal{M}_{\mathcal{F}}$  as the set of indices of all members of family  $\mathcal{F}$ . We say that  $\mathcal{F}$  is *viral* when there exists viral material in the family. Next, define the infection probability of individual  $i$  within viral family  $\mathcal{F}$  as  $\pi_{\text{ind}} = \Pr(X_i = 1 \mid \mathcal{F} \text{ viral})$ , for all  $i \in \mathcal{M}_{\mathcal{F}}$ , and  $\pi_{\text{vf}} = \Pr(\mathcal{F} \text{ viral})$ . Note that the infection status of individuals in a viral family are conditionally independent and identically distributed (i.i.d.).

Under our definition, family  $\mathcal{F}$  being viral need not be attributed to any individual  $i \in \mathcal{M}_{\mathcal{F}}$ . After all, viral material can be on an infected pet or contaminated surface. For this model, once the family is viral, the virus spreads independently with a fixed probability  $\pi_{\text{ind}}$ . Of course, our simplified model may not accurately reflect reality. That said, without a consensus in the literature on how coronavirus spreads, it is unrealistic to create a more accurate model. On the other hand, our model is plausible, and we will see that it is mathematically tractable.

We further assume that individuals cannot be infected unless the family is viral, i.e.,  $\Pr(X_i = 1 \mid \mathcal{F} \text{ not viral}) = 0$ . The family structure serves as SI and allows the group testing algorithm to impose the constraint that people living together have strongly correlated health status.

Next, we derive the exact form of the denoiser (1) by incorporating the family-based infection mechanism. Denote the pseudodata of the members of family  $\mathcal{F}$  as  $\mathbf{v}_{\mathcal{F}} = (v_i)_{i \in \mathcal{M}_{\mathcal{F}}}$ , the family-based denoiser for  $i$ th individual can be decomposed as follows:

$$\begin{aligned} g_{\text{in}}^{\text{family}}(\mathbf{v}_{\mathcal{F}}) &= \mathbb{E}[X_i \mid \mathbf{v}_{\mathcal{F}}] & (7a) \\ &= \Pr(X_i = 1 \mid \mathbf{v}_{\mathcal{F}}) & (7b) \\ &= \Pr(X_i = 1, \mathcal{F} \text{ viral} \mid \mathbf{v}_{\mathcal{F}}) & (7c) \\ &= \Pr(\mathcal{F} \text{ viral} \mid \mathbf{v}_{\mathcal{F}}) \Pr(X_i = 1 \mid \mathbf{v}_{\mathcal{F}}, \mathcal{F} \text{ viral}), & (7d) \end{aligned}$$

where the first term of (7d) is

$$\begin{aligned} &\Pr(\mathcal{F} \text{ viral} \mid \mathbf{v}_{\mathcal{F}}) \\ &= \frac{f(\mathbf{v}_{\mathcal{F}}, \mathcal{F} \text{ viral})}{f(\mathbf{v}_{\mathcal{F}}, \mathcal{F} \text{ viral}) + f(\mathbf{v}_{\mathcal{F}}, \mathcal{F} \text{ not viral})}. \end{aligned} \quad (8)$$

The two quantities in (8) can be further expanded as

$$\begin{aligned} &f(\mathbf{v}_{\mathcal{F}}, \mathcal{F} \text{ not viral}) & (9a) \\ &= (1 - \pi_{\text{vf}}) f(\mathbf{v}_{\mathcal{F}} \mid \mathcal{F} \text{ not viral}) & (9b) \\ &= (1 - \pi_{\text{vf}}) \prod_{i \in \mathcal{M}_{\mathcal{F}}} \mathcal{N}(v_i; 0, \Delta), & (9c) \end{aligned}$$

and

$$\begin{aligned} &f(\mathbf{v}_{\mathcal{F}}, \mathcal{F} \text{ viral}) = \pi_{\text{vf}} f(\mathbf{v}_{\mathcal{F}} \mid \mathcal{F} \text{ viral}) & (10a) \\ &= \pi_{\text{vf}} \sum_{\mathbf{x}_k \in \Omega_{\mathcal{F}}} \prod_{i \in \mathcal{M}_{\mathcal{F}}} & (10b) \\ &\quad \left[ f(v_i \mid X_i = x_{k,i}) \Pr(X_i = x_{k,i} \mid \mathcal{F} \text{ viral}) \right], \end{aligned}$$

where  $\mathcal{N}(x; \mu, \sigma^2) := \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ , and  $\Omega_{\mathcal{F}} = \{0\dots00, 0\dots10, \dots, 1\dots11\}$  is a power set comprised of  $2^{|\mathcal{M}_{\mathcal{F}}|}$  distinct infection patterns for family  $\mathcal{F}$ . The second term of (7d) can be simplified as follows:

$$\begin{aligned} &\Pr(X_i = 1 \mid \mathbf{v}_{\mathcal{F}}, \mathcal{F} \text{ viral}) & (11a) \\ &= \Pr(X_i = 1 \mid v_i, \mathcal{F} \text{ viral}) & (11a) \\ &= \Pr(X_i = 1, v_i \mid \mathcal{F} \text{ viral}) / \Pr(v_i \mid \mathcal{F} \text{ viral}) & (11b) \\ &= \frac{\pi_{\text{ind}} \mathcal{N}(v_i; 1, \Delta)}{\pi_{\text{ind}} \mathcal{N}(v_i; 1, \Delta) + (1 - \pi_{\text{ind}}) \mathcal{N}(v_i; 0, \Delta)} & (11c) \\ &= \left( 1 + \frac{1 - \pi_{\text{ind}}}{\pi_{\text{ind}}} \cdot \frac{\mathcal{N}(v_i; 0, \Delta)}{\mathcal{N}(v_i; 1, \Delta)} \right)^{-1} & (11d) \\ &= \left( 1 + (\pi_{\text{ind}}^{-1} - 1) \exp\left[\left(v_i - \frac{1}{2}\right)/\Delta\right] \right)^{-1}. & (11e) \end{aligned}$$

### 1.2. Contact Tracing Denoiser

While family structure SI characterizes part of the spread of the disease, individual members of a family will presumably all come in close contact with each other, hence CT SI will include cliques for these individuals. Additionally, CT SI describes inter-family contacts. Therefore, CT SI can characterize the spread of the disease more comprehensively than family SI. To exploit the CT SI, we encode it for each individual  $i$  into the prior probability of infection,  $\Pr(X_i = 1)$ , and use the following scalar denoiser:

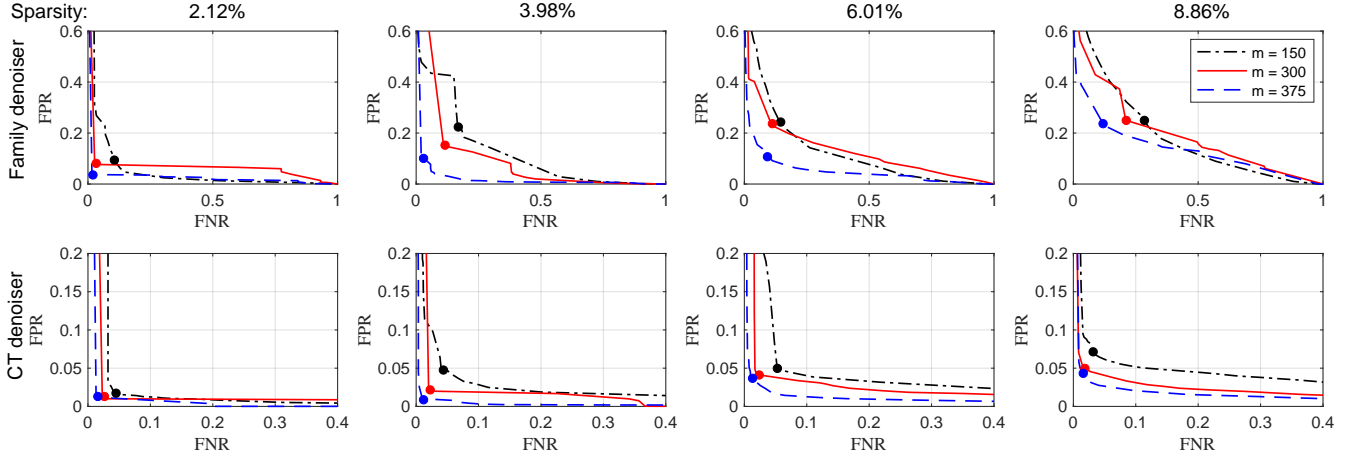
$$\begin{aligned} &g_{\text{in}}^{\text{CT}}(v_i) & (12a) \\ &= \mathbb{E}[X_i \mid v_i] = \Pr(X_i = 1 \mid v_i) & (12a) \\ &= f(v_i \mid X_i = 1) \Pr(X_i = 1) / f(v_i) & (12b) \\ &= \left\{ 1 + [\Pr(X_i = 1)^{-1} - 1] \exp\left[\left(v_i - \frac{1}{2}\right)/\Delta\right] \right\}^{-1}. & (12c) \end{aligned}$$

Here,  $\Pr(X_i = 1)$  for day  $k + 1$  can be estimated by aggregating CT information of individual  $i$  over a so-called *SI period* from day  $k - 7$  to day  $k$  as follows

$$\widehat{\Pr}^{(k+1)}(X_i = 1) = 1 - \prod_{d=k-7}^k \prod_{j=1}^n (1 - \widehat{p}_{i,j}^{(d)}), \quad (13)$$

where  $\widehat{p}_{i,j}^{(d)}$  is the estimated probability of infection of individual  $i$  due to contact with individual  $j$ . This probability,  $\widehat{p}_{i,j}^{(d)}$ , can be determined by the CT information  $(\tau_{ij}^{(d)}, d_{ij}^{(d)})$ , as well as their infection status as follows:

$$\widehat{p}_{i,j}^{(d)} = \exp\left(-(\lambda \tau_{ij}^{(d)} d_{ij}^{(d)} \Psi_{ij}^{(d)} + \epsilon)^{-1}\right), \quad (14)$$



**Fig. 4.** Performance of M1 in terms of ROC when family denoiser (top row) and CT denoiser (bottom row) are used. Columns correspond to averaged sparsity levels ranging from 2.12% to 8.86%. Within each plot, the performance under three measurement levels for a population of  $n = 1000$  individuals is compared. The dot on each curve corresponds to an operating point that minimizes the sum of FPR and FNR. The CT denoiser significantly outperforms the family denoiser with error rates mostly below 0.05. The estimation problem is more challenging when fewer measurements are used at a higher sparsity level.

where  $\Psi_{ij}^{(d)} = 1 - \widehat{\Pr}^{(d)}(X_i = 0) \widehat{\Pr}^{(d)}(X_j = 0)$ ,  $\lambda$  is an unknown Poisson rate parameter, and  $\epsilon$  is used to avoid division by zero. We estimate  $\lambda$  with maximum likelihood (ML) using the pseudodata of all individuals, i.e.,

$$\widehat{\lambda}^{\text{ML}} = \arg \max_{\lambda} \prod_{i=1}^n f(v_i | \lambda), \quad (15)$$

where  $f(v_i | \lambda) = f(v_i | X_i = 1) \Pr(X_i = 1 | \lambda) + f(v_i | X_i = 0) \Pr(X_i = 0 | \lambda)$ . Once  $\widehat{\lambda}^{\text{ML}}$  is obtained, it is plugged into (14) for calculating the prior probability [18]. Note that this plug-in strategy is also used for two other denoisers, namely,  $\lambda = \rho$  for  $g_{\text{in}}^{\text{Bernoulli}}(v_i)$  and  $\lambda = (\pi_{\text{vf}}, \pi_{\text{ind}})$  for  $g_{\text{in}}^{\text{family}}(\mathbf{v})$ .

## 2. ADDITIONAL RESULTS FOR M1

In Sec. 4 of the main paper, we reported the performance of M1 in a compact way, due to space limitations, by choosing a representative operating point on an ROC curve instead of using the whole curve. In this section, we provide complete ROC curves that correspond to the top row of Fig. 3 of the main paper. Fig. 4 illustrates M1's performance for family and CT denoisers at different measurement and sparsity levels. The dot on each curve corresponds to the operating point that minimizes the total error rate, i.e., the sum of FPR and FNR, as reported in Sec. 4 of the main paper. The closer a dot is to the origin of the FPR–FNR plane, the better the performance it reflects. Comparing the ROC curves in the top row to those in the bottom row, we note that the CT denoiser significantly outperforms the family denoiser at all sparsity levels. The CT denoiser, with most of its FNR and FPR < 5%, can achieve as low as 15% of the total error rate of the family

denoiser. Across different sparsity levels, the algorithm performs less accurately as the sparsity level increases. In each plot, lower measurement rates make it more challenging for the group testing algorithm.

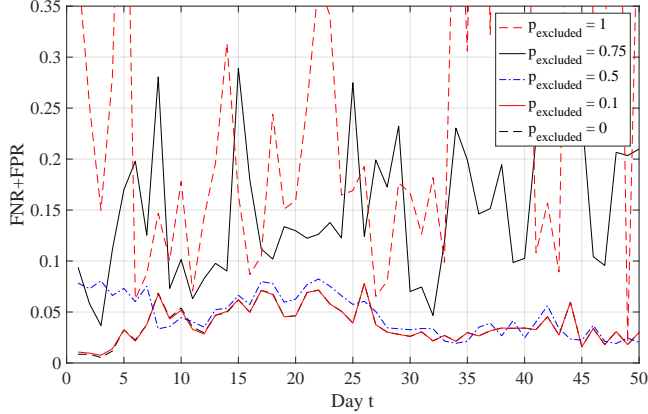
We also examine the stability of the thresholds corresponding to the operating points we selected to report results in Fig. 3 of the main paper. Our empirical results reveal that at a particular sparsity level, the variation of the threshold due to different design matrices or denoisers is less than 0.003. As the sparsity level increases from 2.12% to 8.86%, the threshold only drops from 0.160 to 0.137. Hence, the threshold for minimizing the total error rate is insensitive to the testing conditions.

## 3. ADDITIONAL EXPERIMENTS FOR M1

### 3.1. Using Prior Knowledge of the Infection Status

In this subsection, we examine the advantage that prior knowledge of the population's infection status in the startup phase provides our proposed algorithm for the M1 binary model. As stated in Sec. 3 of the main paper and in (13), our algorithm iteratively uses the updated probability of infection,  $\widehat{\Pr}(X_i = 1)$ , estimated from an SI period of 8 immediately preceding days. Note that for days  $k < 8$ , we had to use the ground-truth infection status of each individual in the startup phase to generate the results reported in Sec. 4 of the main paper. However, ground-truth infection data from the startup phase may provide our approach an unfair advantage over the algorithms proposed for M2. Below, we investigate whether this advantage is significant.

We examine how varying the amount of startup information impacts our algorithm's quality. Specifically, we ran



**Fig. 5.** Performance of M1 when a proportion,  $p_{\text{excluded}}$ , of the population’s health states in the startup phase is unknown. The curves reveal that in the absence of up to 50% prior knowledge of the infection status of the population, the accuracy of M1 is close to that when complete startup information is available.

domly replace a portion,  $p_{\text{excluded}} \in \{0, 0.1, 0.5, 0.75, 1\}$ , of the population’s infection status by an estimated probability of infection, e.g., 5%, for a setup that has a true averaged sparsity level of 7.2%. Using a probability instead of a binary value, 0 or 1, gives the algorithm soft probabilistic information instead of hard ground-truth style information. Fig. 5 shows that even with 50% prior knowledge of the infection status of individuals, our detection accuracy for M1 is close to that when using complete prior information after ramping up for 8 days. The averages of the total error rates across time for increasing  $p_{\text{excluded}}$  are 0.038, 0.039, 0.046, 0.148, and 0.407, respectively. We also tried to replace the startup infection status by an estimated probability of infection of 10%, but only observed negligible performance differences. The results show that the CT algorithm is robust to the absence of up to 50% of startup infection information.

### 3.2. Duration of Startup SI Period

We investigated the impact of the duration of the startup SI on estimation performance. In principle, the longer the SI duration, the more accurate we expect the results to be. There is a trade-off between the accuracy of our algorithm and the startup SI infection status information that needs to be pre-collected before the initialization of the testing algorithm. In our experiment, we tested three startup SI durations, namely, 4 days, 8 days, and 12 days. Our experimental results (omitted for brevity) show that the estimation accuracy is somewhat insensitive to the duration of the SI period. Hence, for the experiments conducted for this paper, we chose 8 days as the SI period.

## 4. AN ADDITIONAL EXPERIMENT FOR M2

### 4.1. Data Generation

For this experiment, we use a different and slightly more general contact tracing graph to simulate the spread of infection. Recall that the adjacency matrix of the contact graph has a block diagonal structure with sizes of cliques coming from the distribution of family sizes in India [17, pg. 18]. However, in this case, we allow two consecutive (according to the order in which cliques appear along the diagonal of the contact matrix) non-trivial cliques (i.e., cliques with more than one node) to have an overlap of one node with probability half. This assumption is reasonable since the concept of family encompasses more general groups such as people at the same workplace, students studying in the same classroom, etc. Furthermore, we remove  $\alpha = 5\%$  of the edges from this block diagonal structure, thus converting the existing cliques into “almost-cliques.” This modified block diagonal structure is kept constant over time while the cross-clique contacts are updated every day. Except for the changes in the underlying contact tracing graph, the rest of the data generation method is the same as that described in Sec. 2 of the main paper.

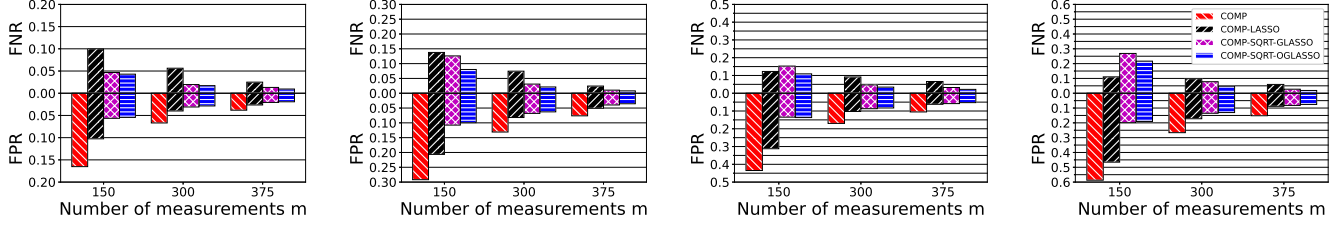
### 4.2. Inference

We use the four algorithms (including COMP) for multiplicative noise described in Sec. 3 of the main paper. However, instead of using maximal cliques as groups in COMP-SQRT-OGLASSO, we use the decomposition of the contact tracing graph into overlapping 3-clique communities [23]. An algorithm for detecting  $k$ -clique communities can be found in Sec. 1 of [23, Supplementary Notes]. The first step of this algorithm involves finding the maximal cliques in the contact graph, for which we use the Bron-Kerbosch algorithm [21]. In the next step, we detect 3-clique communities and label each of those as groups. Further, we also label as groups the maximal cliques that are not part of any of these communities, in order to ensure that every contact is taken into account. The advantage of using 3-clique communities over just maximal cliques is that the former is able to capture “almost cliques,” i.e., cliques with a small fraction of absent pairwise contacts.

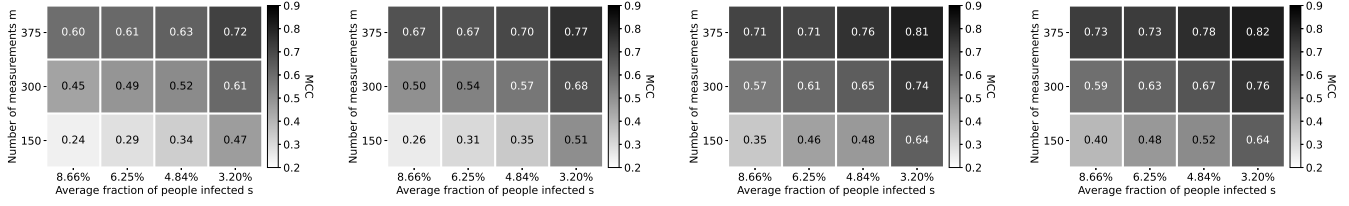
### 4.3. Numerical Results

We present the results in a format similar to that in Sec. 4 of the main paper, but for the contact graph described in Sec. 4.1. Fig. 6 shows the mean values (across 50 signals) of the false negative rate (FNR) and false positive rate (FPR) obtained for four different sparsity levels. The sparsity levels were obtained by varying the amount of cross-clique contacts. We remark that the length of each bar in Fig. 6,  $\text{FNR} + \text{FPR}$ , is equal to  $1 - \text{Youden's Index}$ . Further, we plot heat maps (Fig. 7) to compare the performance of the four algorithms under consideration—the intensity of gray corresponds to the mean value (across 50 signals) of the Matthews Correlation

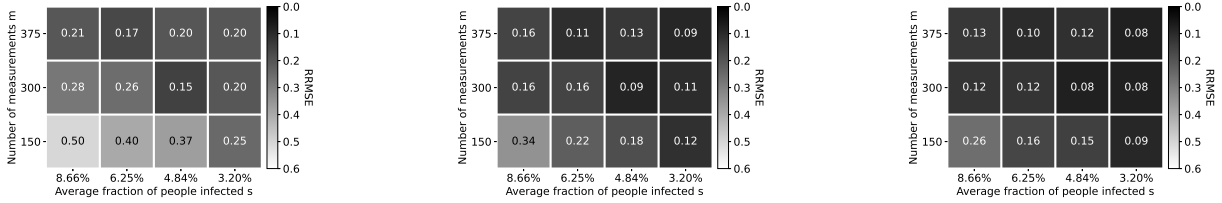




**Fig. 6.** Figure showing mean FNR and FPR values for the contact graph from Section 4.1, for mean sparsity levels of 3.20%, 4.84%, 6.25%, 8.66% (from left to right).



**Fig. 7.** Figure showing mean MCC values obtained using COMP, COMP-LASSO, COMP-SQRT-GLASSO, COMP-SQRT-OGGLASSO (from left to right).



**Fig. 8.** Figure showing mean RRMSE values obtained using COMP-LASSO, COMP-SQRT-GLASSO, COMP-SQRT-OGGLASSO (from left to right).

Coefficient (MCC). The MCC is defined as

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}, \quad (16)$$

and has been proposed as a comprehensive metric to evaluate the performance of binary classification algorithms [27]. Its values range from  $-1$  to  $+1$ , where a value closer toward  $+1$  is desirable. The RRMSE values can be seen from the heat maps in Fig. 8 (we do not provide a heat map for COMP since it does not estimate viral loads).

## 5. ADDITIONAL RESULTS FOR M2

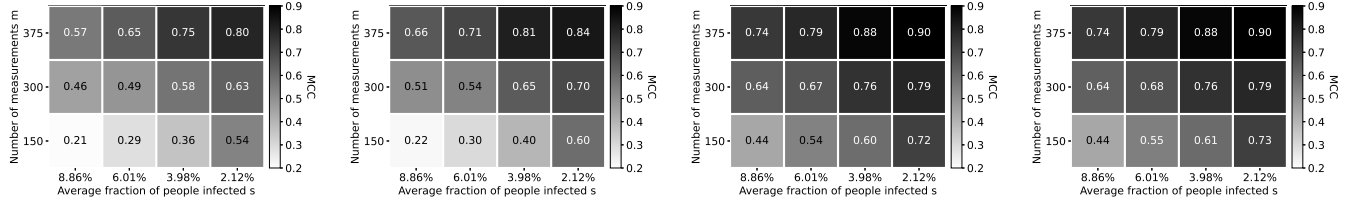
For model **M2**, we present a comparison of the four algorithms for the experiment described in Sec. 4 of the main paper. Fig. 9 and Fig. 10 show a comparison of the performance of the algorithms under consideration in terms of mean MCC and mean RRMSE values, respectively. Further, we remark that the true viral loads of the false negatives yielded

by COMP-LASSO variants are concentrated toward lower values. For instance, only about 29% of the false negatives given by COMP-SQRT-OGGLASSO had viral load values greater than  $2^{12} = 4096$ .

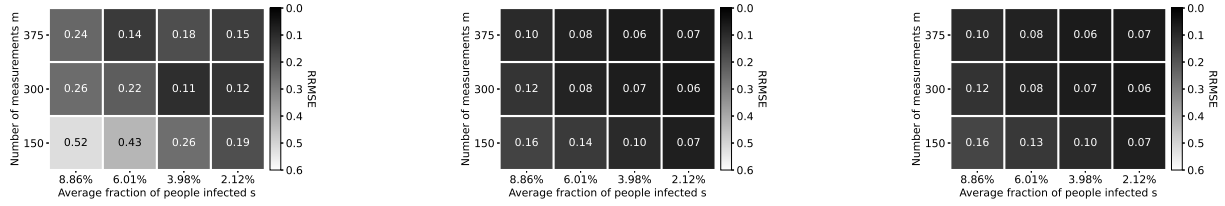
## 6. SENSING MATRIX DESIGN

As mentioned in the main paper, we use Kirkman triple matrices as sensing matrices for performing pooling. A Kirkman triple (binary) matrix  $\mathbf{A}$  can be partitioned into  $3n/m$  sub-matrices of dimensions  $m \times m/3$ , each of which contains exactly one nonzero entry in each row and three nonzero entries in each column. Further, the dot product of any two columns of the matrix  $\mathbf{A}$  should not exceed 1. For a given value of  $n$ ,  $m (< n)$  must satisfy the following conditions:

1.  $m$  must be of the form  $3n_1$ , where  $n_1$  divides  $n$ , since the number of sub-matrices and the number of columns in each sub-matrix must be integers.



**Fig. 9.** Figure showing mean MCC values obtained using COMP, COMP-LASSO, COMP-SQRT-GLASSO, COMP-SQRT-OGGLASSO (from left to right).



**Fig. 10.** Figure showing mean RRMSE values obtained using COMP-LASSO, COMP-SQRT-GLASSO, COMP-SQRT-OGGLASSO (from left to right).

2.  $\binom{3}{2} \cdot n \leq m(m-1)/2$  since a triple contains  $\binom{3}{2}$  pairs and a pair must belong to at most one triple.

For  $n = 1000$ , the only values of  $m$  which satisfy the above constraints are 120, 150, 300, 375, 600, and 750. We construct Kirkman triple matrices with  $m = 150, 300, 375$  and use them in our experiments. The matrices are constructed based on a few simple rules:

1. The indices of ones in each column form an arithmetic progression (AP).
2. The matrix has a block structure and the common difference of the AP remains constant throughout each block. Furthermore, the sum of all columns in a block yields the vector consisting of all ones.
3. The common difference values  $\{d_B : \mathbf{B} \text{ is a block}\}$  are chosen such that the multi-set  $\{r \cdot d_B : r \in \{n : n \in \mathbb{N}, n < 3\}, \mathbf{B} \text{ is a block}\}$  has no duplicate values.

Fig. 11 shows the structure of a  $375 \times 1000$  Kirkman triple matrix obtained using the above approach. Let  $\mathbf{B}$  be any block and let  $d_B$  denote the common difference of the AP for block  $\mathbf{B}$  as indicated in the figure. Then, the  $i$ th column of  $\mathbf{B}$  is given by

$$\mathbf{B}_i = \sum_{j=1}^{3-1} e_{\beta_i + j d_B}, \quad \beta_i = \text{mod}(i, d_B) + 3d_B \lceil i/d_B \rceil, \quad (17)$$

where  $\lceil \cdot \rceil$  denoted the greatest integer function and  $e_j$  denotes the  $j$ th standard basis vector. Clearly, any block  $\mathbf{B}$  must have dimensions  $3n_B \times n_B$ , where  $d_B$  divides  $n_B$ .

As  $m$  decreases, it becomes harder to design matrices satisfying all three rules specified earlier. However, it is possible to relax the third rule in such cases and still obtain a matrix satisfying the required constraints. For example, our  $150 \times 1000$  Kirkman triple matrix does not obey the third rule. We further remark that one may design balanced matrices with a different number of (say  $k$ ) ones in each column such that the dot product of every pair of columns is bounded by 1, using the above approach. Such matrices would arise from the Steiner systems  $S(2, k, m)$  [just as Kirkman matrices arise from  $S(2, 3, m)$ ]. For example, it is straightforward to design a  $400 \times 1000$  matrix with  $k = 4$  using our approach.

1	5	25	125	59	60	61	62
				66	65	64	63

**Fig. 11.** A  $375 \times 1000$  Kirkman triple matrix obtained using our approach. The number written within each block is equal to the corresponding common difference value and the blocks without number markings are zero matrices. The blocks having less than  $m = 375$  rows have dimensions equal to  $3d_B \times d_B$ .