No time to waste: practical statistical contact tracing with few low-bit messages

Romijnders, R.; Asano, Y.M.; Louizos, C.; Welling, M.

Publication date
2023
Document Version
Final published version
Published in
Proceedings of Machine Learning Research
License
Other

No time to waste: practical statistical contact tracing with few low-bit messages

Rob Romijnders
University of Amsterdam

Yuki M. Asano
University of Amsterdam

Christos Louizos
Qualcomm AI Research[1]

Max Welling
University of Amsterdam

Abstract

Pandemics have a major impact on society and the economy. In the case of a new virus, such as COVID-19, high-grade tests and vaccines might be slow to develop and scarce in the crucial initial phase. With no time to waste and lockdowns being expensive, contact tracing is thus an essential tool for policymakers. In theory, statistical inference on a virus transmission model can provide an effective method for tracing infections. However, in practice, such algorithms need to run decentralized, rendering existing methods – that require hundreds or even thousands of daily messages per person – infeasible. In this paper, we develop an algorithm that (i) requires only a few (2-5) daily messages, (ii) works with extremely low bandwidths (3-5 bits) and (iii) enables quarantining and targeted testing that drastically reduces the peak and length of the pandemic. We compare the effectiveness of our algorithm using two agent-based simulators of realistic contact patterns and pandemic parameters and show that it performs well even with low bandwidth, imprecise tests, and incomplete population coverage.

1 Introduction

Pandemics like COVID-19 are disastrous for society. With the development of vaccines taking months, if not years, an early understanding of virus spread is critical. Drastic interventions like lockdowns can be successful but have devastating economic and societal consequences (Kaye et al., 2021; Boden et al., 2021; Vindegaard and Benros, 2020). Contact tracing – whereby virus transmission among an individual's contacts is traced – provides a more elegant alternative to prevent, or at least monitor, the rise of a pandemic (Li and Saad, 2021).

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206.

With incomplete and imperfect tests, statistical contact tracing can provide a future- and action-oriented approach Baker et al. (2021) and enable effective mitigating strategies, such as preventative testing, and research on the virus spread dynamics (Carinci, 2020). However, the difficulty lies in developing practical decentralized algorithms, as the sensitive information of an individual's disease status should not be known to a central entity.

The two key factors to consider for such algorithms are communication costs and privacy. While privacy also depends heavily on implementation and encryption, here we focus on reducing communication costs at a given model performance. Previous works established that statistical contact tracing can far outperform post-infection contact tracing (Herbrich et al., 2020; Baker et al., 2021), but still requires large amounts of communication. Decentralized algorithms typically work in "rounds", in each of which every device sends update "messages" to the ones it has been in contact with (e.g. measured via close-range bluetooth) and subsequently computes internal updates. Since each message between decentralized entities requires encryption and synchronization, *the number of messaging rounds should be low.* However, the Gibbs sampling method of Herbrich et al. (2020) requires up to thousands of daily messages, and even existing belief propagation (BP) algorithms require up to a hundred every day (Baker et al., 2021).

This paper presents an algorithm that requires only *five* messages per contact per day, given the same performance. With each of our algorithm's messages quantized to only 4 bits, it requires *less than sending a one character ASCII message* to every contact each day. This aligns with existing contact tracing software that advocate for four or fewer bits per update (Alsdurf et al., 2020; Apple and Google, 2020). However, because contact graphs can have loops (`A` contact with `B`, `B` contact with `C`, `C` contact with `A`), inference is not easy, and we analyse the behavior of our inference algorithms on realistic human contact patterns, comparing with the Gibbs sampler of Herbrich et al. (2020) and belief propagation (BP) Baker et al. (2021) and extensively ablate its properties. Finally, we evaluate our algorithm on the OpenABM-Covid19 simulator (Hinch et al., 2021). This simulator has been developed to test various COVID-

19 containment policies. It is an agent-based model (ABM) that uses more than 150 parameters reflecting, among others, information about different household, age and contact characteristics. Deploying our method out-of-the-box on this extensive simulation, we find the preventative effect is even more pronounced, yielding two orders of magnitude reduction in peak infection rate given the same amount of communication.

Overall, this paper makes three contributions:

1. An inference algorithm that requires only up to five daily messages per contact, while successfully predicting individual virus spread;
2. An analysis of the difficulty of statistical modeling of the presented problem and why standard Gibbs sampling has poor performance;
3. Comparison of our algorithm against two published algorithms (Herbrich et al., 2020; Baker et al., 2021), and demonstration of suppression of pandemic virus spread at lower communication costs, established on two realistic simulators.

Code is available at:
`http://github.com/QUVA-Lab/nttw`.

2 Related work

Our work develops novel inference algorithms for inferring individual virus states in a pandemic. As such, related work comprises two fields: virus spread modeling and statistical inference.

Virus spread modeling We emphasize that we study only decentralized algorithms in this paper. Nevertheless, related work has also centralized approaches. For example, (Biazzo et al., 2021) uses neural networks to sample infection histories, Lorch et al. (2004) uses Bayesian optimization to infer and learn parameters. However, the analysis is more aimed at post-hoc analysis rather than individual-based prequential solutions. Finally, Wood et al. (2020) advocate the use of the probabilistic programming approach, but currently the centralized probabilistic program has no option for decentralization (smartphones). Worth mentioning are two other approaches that assume known infection status and investigate the learning of parameters (which could be an outer loop to our approach, and we defer to future work) (Vineetha Warriyar et al., 2020; Myers and Leskovec, 2010; Mathioudakis et al., 2011).

The decentralized setup avoids security questions of centralized storage, which are discussed in (Park et al., 2020; Grantz et al., 2020; Troncoso et al., 2020; Raskar et al., 2020). Moreover, we take example from the adoption of other approaches to virus spread modeling (Bay et al., 2020; Chan et al., 2020; Cho et al., 2020; Bestvina, 2020).

Other approaches for statistics-based individual virus spread modeling are: Alsdurf et al. (2020) that provides minimal detail and results but sketches a high level algorithm; Bestvina (2020) that suggests a propagation model of heuristic scores but provides no statistical derivation; and finally Baker et al. (2021) that formulate a belief propagation approach on a different graph that we will compare to in Section 5.

Statistical inference Formulating virus spread as a large probabilistic graphical model (PGM) (Pearl, 1989), traditional inference algorithms are at our disposal. We study two major families of inference algorithms: sampling-based approaches, and deterministic approaches. Sampling-based approaches such as Gibbs sampling draw samples from the posterior. However, iterative sampling algorithms often yield correlated samples and thus many steps need to be made in order to reduce the posterior variance Robert and Casella (2004). We will argue later that the required amount of messages to obtain posterior estimates is unpractical for contact tracing. (Herbrich et al., 2020) derives a blocked Gibbs sampler, where each block are the random variables per timestep corresponding to one user. However, even a blocked Gibbs sampler requires hundreds or even thousands of messages for 'good' inference. We will compare with the inference algorithm from this paper and use their synthetic data simulator.

The second class of inference algorithms stems from belief propagation (BP; c.f. Bishop (2007)). BP algorithms will send messages in the form of beliefs about neighboring random variables. Two works on virus spread modeling have proposed implementations of belief propagation (Herbrich et al., 2020; Baker et al., 2021). However, Herbrich et al. (2020) provided no experimental results on BP. The approach in (Baker et al., 2021) shows improvement over baselines of traditional contact tracing, but requires twenty to a hundred messages, which we deem impractical. Finally, the algorithm prescribes messages over a domain that is hard to quantize (studied in Section 5).

Generalizing the above, a key related work to ours developed a new inference paradigm (Rosen-Zvi et al., 2005). The authors of this work showed how both BP, Gibbs sampling, and mean-field variational inference all relate to a set of equations named DLR equations (see (Rosen-Zvi et al., 2005) and definitions therein). From this generalization, follows a new, deterministic inference algorithm, Factorized Neighbors (FN; described in Section 2 of Rosen-Zvi et al. (2005)). While other improvements of belief propagation are known, (Yuille, 2002; Welling and Teh, 2001), we focus on the Factorized Neighbor algorithm due to its simplicity and decentralized nature.

[1]Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc. and/or its subsidiaries.

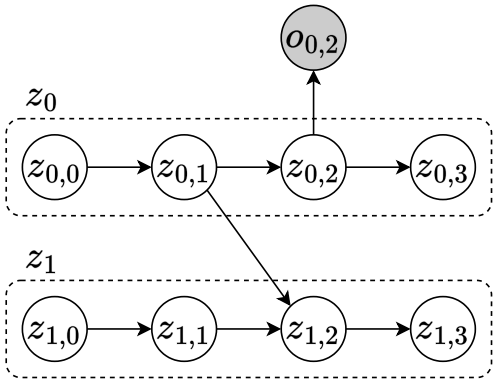

Figure 1: Example of a PGM with two users, z_0 and z_1 and one observation, $o_{0,2}$. User 0 has a directed contact with user 1 on day 1. This contact could potentially transmit the virus, causing user 1 to switch from S state to E state on day 2.

3 Data and model

3.1 A probabilistic model for virus spread

This section discusses the model of virus spread and possible simulators to obtain realistic data. The states, Susceptible, Exposed, Infected, Recovered (SEIR) (Kermack and McKendrick, 1927; Anderson and May, 1992), are considered in a model formulated according to Herbrich et al. (2020). Each individual can be, on each given day, in one of the four SEIR states.

Each individual on each day corresponds to a random variable (being one of the SEIR states). Daily transitions and virus transmissions will be modeled with conditional probability distributions. This collection of random variables and conditional distributions can be drawn as a probabilistic graphical model (PGM) (Pearl, 1989). The PGM will have two types of edges, corresponding to the two types of conditional distributions. The first corresponds to the noisy tests, where we follow the same observation distribution, $p(o_{u,t}|z_{u,t})$, as Herbrich et al. (2020) and use the same false positive and false negative rates for the tests. The other type of edges corresponds to the conditional distribution of daily transitions and the influence of contacts. These require more attention as they will complicate the inference procedure. Moreover, these edges may form loops which cause optimization issues for some algorithms.

An example for one such model is in Figure 1, where user 0, z_0, *has contact with* user 1 on day 1, and has a test on the day after (observed variables are shaded by convention (Koller and Friedman, 2009)). Note that contacts are directional in all our experiments.

The conditional probability distributions corresponding to daily transitions follow Herbrich et al. (2020) and are repeated here briefly:

$$P(z_{u,t+1}|\mathcal{Z}_t) = \begin{cases} f(u,t,\mathcal{Z}_t) & \text{if } z_t = S,\ z_{t+1} = S \\ 1 - f(u,t,\mathcal{Z}_t) & \text{if } z_t = S,\ z_{t+1} = E \\ 1 - g & \text{if } z_t = E,\ z_{t+1} = E \\ g & \text{if } z_t = E,\ z_{t+1} = I \\ 1 - h & \text{if } z_t = I,\ z_{t+1} = I \\ h & \text{if } z_t = I,\ z_{t+1} = R \\ 1 & \text{if } z_t = R,\ z_{t+1} = R \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Here, variables g and h are dynamics parameters obtained from other studies. The function f is defined as

$$f(u,t,\mathcal{Z}_t) = (1-p_0)(1-p_1)^{|\{(v,u,t)\in\mathcal{D}:z_{v,t}=I\}|}. \tag{2}$$

$\mathcal{D}$ is the set of all contacts, and $\{(v,u,t) \in \mathcal{D} : z_{v,t} = I\}$ is the set of infected contacts. Mind that contacts are directional, so (v,u,t) exists in the set when user v has a directive contact to u at timestep t (and thus influences the SEIR status for user u on day $t+1$). Each infected contact decreases the probability that user u stays in state S with a factor $1-p_1$. In our experiments, we set p_1 at 0.3 to have a realistic pandemic within the 50 days that a simulation runs. p_0 is the probability of external infection, which is set at 0.001.

3.2 Data simulator

Experiments make use of two simulators for individual-based virus spread data.

The *CRISP simulator* models society as a combination of cliques, where more contacts happen within the clique than between cliques (Herbrich et al., 2020). Contact patterns follow either a uniform distribution (each individual has a similar amount of contacts), or a power-law (roughly 20% of individuals have 80% of the amount of contacts, roughly 80% of individuals have 20% of contacts). A virus transmission can happen via a contact following the noisy-or model in Equation (1). Further details can be found in (Herbrich et al., 2020) and corresponding open-source code.

The *OpenABM-Covid19* simulator uses more than 150 parameters, stratifying among age groups, households and simulating eleven disease states (Hinch et al., 2021). The default parameter set focuses on the United Kingdom, modeling up to a million individuals. Parameters are based on nine population surveys and other academic research. We use the parameters as provided, similar to Baker et al. (2021). Experiments run for 10,000 individuals and parameters are scaled appropriately (included in open-source code). Further experimental details are in Appendix B.

Other relevant simulators are (Alsdurf et al., 2020; Lorch et al., 2004; Mehrjou et al., 2021; Deb et al., 2020), but we believe the OpenABM simulator combines a well researched parameter set, with a practical and fast open-source implementation.

Both simulations run with a population of 10,000 people. With the CRISP, 15% of the population could be (noisily) tested, and at most 5% can be quarantined daily. Disease dynamics are set $p_0 = \frac{1}{1000}$ and $p_1 = \frac{1}{10}$, and the simulation is run for 50 days, unless stated otherwise. With the OpenABM simulator, the simulation runs for 100 days. Tested individuals (at most 15%) are chosen as users with highest infectiousness score on the previous day. At most 10% of people can be quarantined. This testing policy follows the conditional testing of related literature (Baker et al., 2021).

4 Inference

Four inference algorithms will be investigated: Gibbs sampling, two variants of belief propagation (BP), and the Factorized Neighbors (FN) algorithm. The goal of the inference algorithm constitutes calculating the posterior distribution over $\{S, E, I, R\}$ for a particular individual on a particular day, given the test observations. In other words, the sentence *'What is the probability of infection for user 9 on day 13?'* translates to $p(z_{9,13} = I|\mathcal{O})$. $z_{u,t}$ indicates the random variable with domain $\{S, E, I, R\}$ for user u on time t. z_u will be shorthand for the collection of random variables among all time steps: $\{z_{u,t}\}_{t=0}^{T-1}$. Finally, $\mathcal{O}$ is the set of all observations.

4.1 Gibbs sampling and belief propagation

Gibbs sampling and Belief propagation follow the formulations in Herbrich et al. (2020). The essential formulae will be repeated here for clarity. Gibbs sampling utilizes blocked Gibbs sampling, where each block comprises the random variables of one particular user, z_u. One Gibbs step resamples the block for each user separately according to:

$$p(z_u|\hat{z}_{\neg u}, \mathcal{O}). \tag{3}$$

$\hat{z}_{\neg u}$ indicates the Gibbs sample for all users except user u, and $\mathcal{O}$ is the set of all observations (e.g., COVID-19 tests). Gibbs sampling is run with ten burn-in steps.

Whereas Gibbs sampling communicates samples among users, BP (and later FN) communicate real-valued messages. Belief propagation can be thought of as a decentralized way to implement sum-product inference (Koller and Friedman, 2009) (when the graph is acyclic), or optimizing the Bethe free energy in a loopy graph (Yedidia et al., 2000).

$$\mu_{f_s \to z_{u,t}}(z_{u,t}) = \sum_{z_s} f_s(z_s, z_{u,t}) \prod_{k \in \text{Nb}(f_s) \setminus z_{u,t}} \mu_{z_k \to f_s} \tag{4}$$

$$\mu_{z_{u,t} \to f_s}(z_{u,t}) = \prod_{k \in \text{Nb}(z_{u,t}) \setminus f_s} \mu_{f_k \to z_{u,t}} \tag{5}$$

In the above, f_s are factors (conditional distributions). Each factor corresponds to an instance of the transition distribution in Equation (1). $z_{u,t}$ are the SEIR states of user u at time t, z_s are random variables that share the scope of f_s and z_k are all variables that send messages to f_s. Finally, the marginal posterior belief for a variable follows from:

$$\beta_{z_{u,t}} = \prod_{k \in \text{Nb}(z_{u,t})} \mu_{f_k \to z_{u,t}}. \tag{6}$$

Two previous works established implementations for the actual messages (Baker et al., 2021; Herbrich et al., 2020), and we will compare with both. For (Baker et al., 2021), we will compare with the open-sourced code in Section 5.1 and refer to this formulation as SIB.

4.2 Factorized Neighbors

The discussed Gibbs sampling and Belief propagation have two problems: a) Gibbs sampling is slow to mix and needs thousands of daily messages which is unpractical (c.f. Section 5.2), b) implementations for BP exist, but some still require up to hundreds of daily messages (c.f. Section 5.1). We establish a more practical algorithm from another view of decentralized inference. Rosen-Zvi et al. (2005) generalized Gibbs sampling, Belief Propagation, and mean-field variational inference. From the generalization, the authors deduce another algorithm, Factorized Neighbors (FN). Although their paper only shows the FN algorithm on an undirected model, an Ising model, we derive the update equations for the particular inference problem in our model (c.f. Equation 1). The formulae yield a decentralized inference algorithm with messages representing local beliefs of infection.

At a high level, FN comprises a set of fixed point equations. It is known that Belief Propagation and Gibbs sampling, under specific circumstances, arrive at a fixed point of the same set of equations. FN iterates by updating beliefs of nodes by marginalizing conditional distributions with the beliefs of neighboring nodes:

$$\begin{aligned} b_u(z_u) &= \sum_{z_{N(u)}} P(z_u|z_{N(u)}, \mathcal{O}) B_{N(u)}(z_{N(u)}) \\ &= E_{B_{N(u)}(z_{N(u)})}[P(z_u|z_{N(u)}, \mathcal{O})]. \end{aligned} \tag{7}$$

Here, $B_{N(u)}(z_{N(u)})$ is the belief over the neighboring nodes of u. This belief factorizes into a product of the neighboring beliefs $B_{N(u)}(z_{N(u)}) = \prod_{v \in N(u)} b_v(z_v)$, which yields the Factorized Neighbor algorithm its name.

Our main contribution is deriving efficient computation for this update equation. A naive computation of the expected value in Equation (7) grows exponentially in the number of neighboring nodes, because the neighborhood belief, $B_{N(u)}$ has a domain of $\mathcal{O}(3^{T \cdot |N(u)|})$ sequences. The amount of neighboring nodes, $|N(u)|$, can be quite significant: some users can accrue tens or even hundreds of contacts in a day by for example, visiting a sports game or music concert. With a user having 100 contacts, a realistic number in popular events (Rutten et al., 2022), the computation would take $4^{100} = 1.6 \cdot 10^{60}$ flops, which is obviously unfeasible. (A similar reasoning explains why variational inference is infeasible, see Appendix A.2)

The central assumption in Factorized Neighbors inference, to our advantage, is that beliefs over neighboring nodes are modeled as a factored distribution. The expected value of the noisy-or construction (Koller and Friedman, 2009), in Equation 2 then turns into a product of expectations. For clarity, the following derivation marks the product in green and beliefs in blue:

$$
\begin{aligned}
& E_{B_{N(u)}(z_{N(u)})}\Big[\\
& \quad p(z_{v,\tau+1} = S | z_{v,\tau} = S, \{z_{v_c,\tau}\}_{c=0}^{C-1})\Big] \\
& = E_{B_{N(u)}(z_{N(u)})}\left[(1-p_0)\prod_{c=0}^{C-1}(1-p_1)^{\mathbf{1}[z_{v_c,\tau}]}\right] \\
& = (1-p_0)\prod_{c=0}^{C-1} E_{b_c(z_{v_c,\tau})}\left[(1-p_1)^{\mathbf{1}[z_{v_c,\tau}]}\right] \qquad (8)
\end{aligned}
$$

Simplicity is the final advantage of the Factorized Neighbor algorithm. The transitions in a SEIR system, formulated by Herbrich et al. (2020), constitute a Markov transition, where each next node is independent of the past given the present. The expected value in Equation (7) takes this Markov transition under the expectations of (infected) contacts. This also constitutes the main difference with BP and might explain why FN is more robust to noisy tests and stale updates (for example, from other smartphones with an empty battery or temporarily lacking internet access).

4.3 Quantizing messages

The messages in a practical algorithm need to be quantized and we use uniform mid-rise quantization. Specifically, with $\lfloor . \rfloor$ the floor function, c bits and $\kappa = 2^c$ quantization levels, values are quantized to $f(x) = \frac{1}{\kappa} \cdot \lfloor x \cdot \kappa + \frac{1}{2} \rfloor$.

For both BP and FN, the backward messages comprise of four real-valued numbers, which incur $4 \cdot c$ bits for 2^c levels of quantization. However, we will find that backward messages are superfluous, and only forward messages are necessary (Results Section 5.3). These comprise one real valued number and thus incur c bits for 2^c levels of quantization. For the iterative algorithms, communication load is calculated as number of message rounds times the about of bits per contact. In Gibbs sampling, both forward and backward message have 1 bit (corresponding to Equations 26 and 29 in Section 4.2 of (Herbrich et al., 2020)). Therefore, each Gibbs sample comprises two bits per contact.

5 Experimental Results

5.1 Pandemic mitigation

First, we compare various algorithms in terms of their communication load and their potential for mitigating a pandemic. Figure 2 shows these results. Four inference algorithms (Gibbs, BP, SIB, FN (ours)) are compared against a random baseline whilst varying the communication load involved. The x-axis indicates multiples of sending around 1 message of 1 bit for each day and each contact. In the case of BP and FN, each algorithm sends five daily messages per contact, and the amount of quantization is varied; in the case of Gibbs, each message is a Gibbs sample (which is only 1 bit, see Subsection 4.3), and the number of messages is varied. The y-axis indicates 'peak infection rate', which is the highest daily proportion of infected users throughout simulation – the most deleterious outcome for economy and society given limited healthcare capacities.

Two observations follow from Figure 2: FN works best under low communication, as can be seen by the strictly better performance of the blue curve compared to the others for the left side of the plot. We speculate that the expectations in Equation 7 are more stable under lower communication amounts than the message-based approach of Equation (4). Figure 3 shows multiple algorithms at a communication load of 20 bits, for 150 days. Note that multiple apps of the COVID-19 pandemic advocate for low-bit messages (Alsdurf et al., 2020; Apple and Google, 2020; CoEpi, 2020) and the improvements of FN in this low-bit regime show its promising capabilities.

5.2 Analysing the mixing of Gibbs chains

Another stark observation from Figure 2 is that Gibbs sampling does worse than either FN and BP – even in the high communication load settings. To better understand this, we next analyse the mixing of Gibbs chains and find that the blocked Gibbs sampler mixes poorly. Therefore, one chain might not reflect the entire posterior. Appendix A.5 shows examples of estimated marginals from different chains and how these reflect different modes. For example, a user

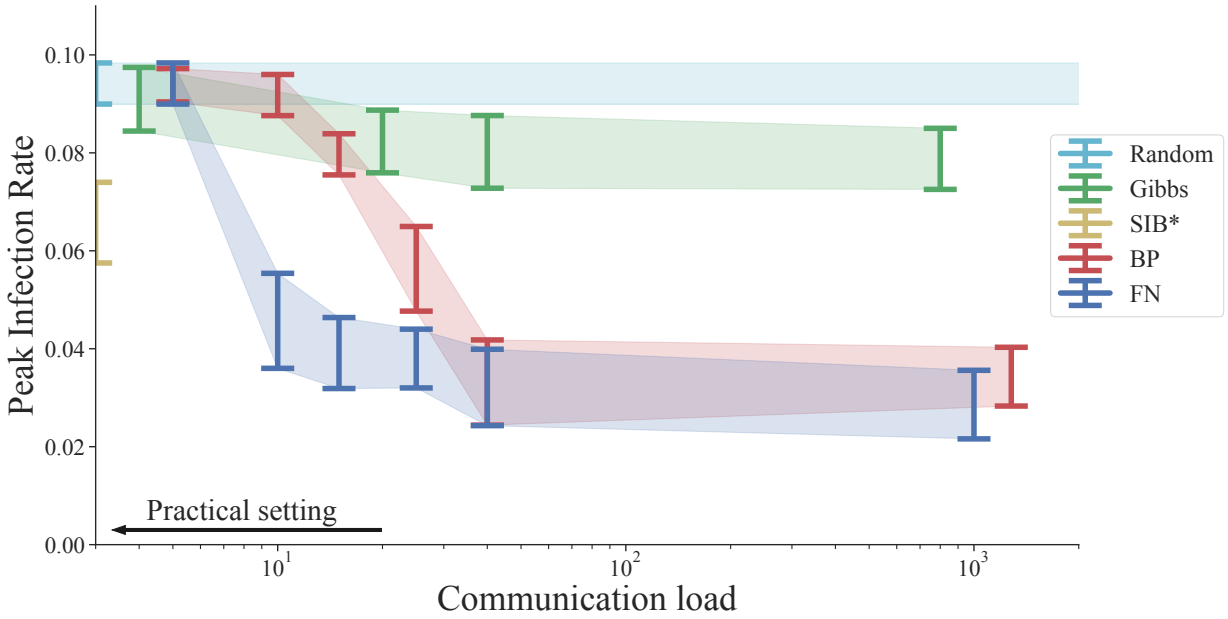

Figure 2: Comparing epidemic mitigation at increasing communication budget. Both BP and FN send five daily messages per contact, which we deem practically feasible. Communication load on the x-axis indicates multiples of an algorithm that sends 1 message of 1 bit per day per contact. * the SIB algorithm is unquantized and thus plotted on the y-axis for reference. Error bars are calculated over four random seeds and shown as the population mean $\pm$ one standard deviation.

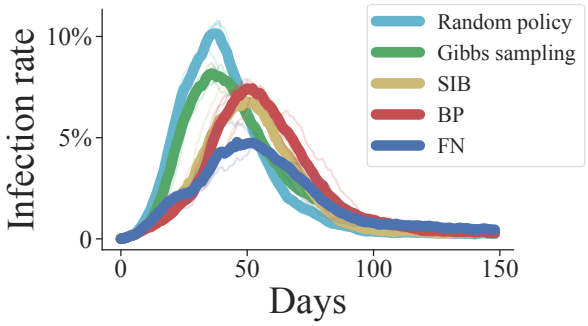

Figure 3: Infection rates on the CRISP simulator. BP and FN send five messages of four bits each. Gibbs sampling takes ten samples, which require two bits each, also corresponding to a communication load of 20 bits.

can have two or more different contacts before an infection. Different Gibbs chains might have different estimates for which contact 'initiated the infection'. To quantitatively assess the mixing time of a Gibbs sampler, we run two chains on a contact graph of 1000 users (following the data generator of (Herbrich et al., 2020), c.f. Section 3). Figure 4 shows the mean absolute error between marginals estimated from two chains. Both axes have a logarithmic scale and we observe a slope of $-\frac{1}{3}$, meaning that to be one order of magnitude closer in divergence, one needs three orders of magnitude more samples. However, more than a thousand messages would be far to the right of Figure 2, meaning a highly unpractical amount of message rounds. In Appendix A.6 we provide further analysis as to *why* Gibbs sampling mixes slowly and find that the spectral gap of the Gibbs transition kernel decreases as the graph becomes more loopy.

5.3 Note on backward messages

Researching the inference methods for our model in the context of realistic settings, we found that half of the messages in the decentralized BP and FN algorithm are unnecessary. Each contact in the model, visualized in the PGM of Figure 1, expends two messages, a forward and a backward message. For BP, the forward message goes from the random variable of the 'sending' user to the factor of the 'receiving' user (c.f. eq. (4)), and the backward message goes in the other direction (in the same round). For FN, the forward message consists of the belief for infection of the 'sending' user at time of contact. The backward message consists of the change of beliefs in the transition $S \to E$ for the 'receiving' user (c.f. Appendix A.1).

However, empirically, the backward messages regress to uniform distribution for most messages. More specifically, the messages (that distribute over four states, $\{S, E, I, R\}$), are equal to 0.25 in 99.9% of the backward messages. As such, we set the backward message to the uniform distribution and do not calculate nor send it in any experiment to save a significant portion of the communication budget.

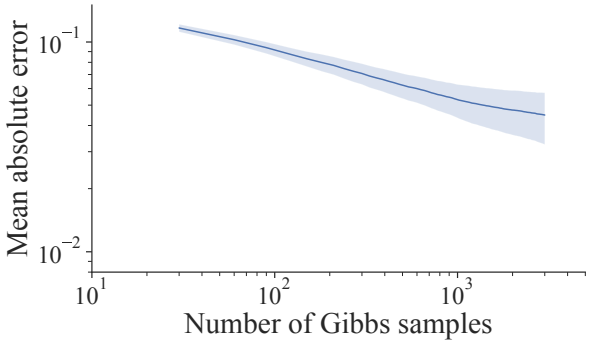

Figure 4: Error between Gibbs chains to show slow mixing. Bringing estimates from Gibbs chains closer together by almost an order of magnitude (in Mean Absolute Error of estimated marginals) requires three orders of magnitude more samples. Note that in reality, one can take only five or ten samples due to synchronization and security overhead.

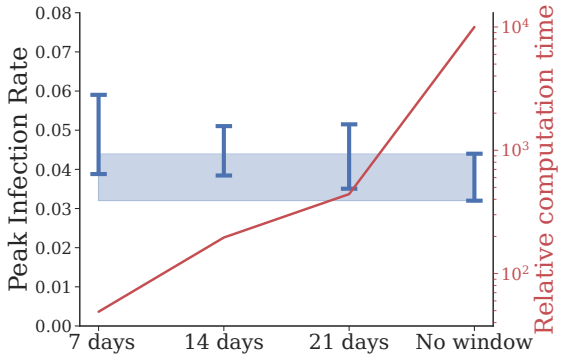

Figure 5: Running statistical inference in a simulator when making predictions using a limited number of days. The simulation runs fifty days, so a window of 21 days provides a significant saving of compute at a similar resulting peak infection rate.

An explanation for why the backward messages are uniform is the following: a backward message informs the sending user of its 'belief of infectiousness' (interpretation differs slightly between Gibbs, BP and FN). However, only if a receiving user were sure to be uninfected before the contact and sure to be infected after the contact would the backward message be informative. This scenario is unlikely as a) users typically have multiple contacts, which dilutes the informativeness, b) tests have false positives and false negatives, which dilutes the certainty of the local states, and c) tests are rather scarce, so a user rarely has a test right before and after a contact. We tested these three factors and concluded that each factor regresses the message to the uninformative uniform distribution.

5.4 Windowed inference

Running inference for contact tracing on longer time windows requires more compute. Approaches like Gibbs sampling from Herbrich et al. (2020) grow as $\mathcal{O}(T^3)$, where T is the number of days in a simulation. Message based approaches like BP and FN will also grow linearly in the number of days. Therefore, we investigate a heuristic where inference is only done over a short time window. For each step, the marginals obtained from the posterior previously are set as a prior for the window. Similar to Baker et al. (2021), a window of 21 days is used. This corresponds to setting the posterior of twenty days in the past as prior for the next window.

Figure 5 shows the resulting peak infection rates when inference is done over shorter windows. Even if computation grows linear with the number of days, cumulative runtime for a simulation grows quadratically. Therefore running simulations using a 21-day window provides a significant saving of compute at minimal performance reduction.

5.5 Robustness

Most computerized simulations happen in idealized circumstances. However, when a real pandemic hits, circumstances are far from ideal. In this Section, we compare three algorithms on two robustness scenarios: noisy test observations and stale updates from unavailable devices.

For the noisy tests, we consult European guidelines on testing for the COVID-19 pandemic and find that some tests had as high as a 25% false positive rate (for Disease prevention and control, 2021; Stohr et al., 2022). Moreover, the same guidelines stated that tests would be admissible with false negative rates as high as 3%. As such, we seek to simulate our inference algorithms under this noisy scenario and change the observation model such that false positives and false negatives are made. False positive rates (fpr) vary from 1%, to 10%, up to 25%, and false negative rates (fnr) vary from 0.1%, to 1%, up to 3%. (The first scenario of noisiness, fpr 1% and fnr 0.1%, correspond to the same values as Herbrich et al. (2020); further details are in Appendix and open-source code).

For the stale phones, we model a scenario where a group of 'stale phones' only updates messages half the time. In the experiment, the group of 'stale phones' increases from 10% to 20% up to 50%.

Results for the robustness experiments are displayed in Table 1 for the CRISP simulator and Table 2 for the OpenABM-Covid19 simulator. This result is obtained by running inference with five daily messages per contact, each with four bits (thus corresponding to a value of 20 on the x-axis of Figure 2). Both the 'noisy test' and 'stale phone' scenarios result in higher peak infection rates. However, compared to BP, FN has lower peak infection rates in most scenarios where tests are noisy and phones have stale updates.

Robustness setup	BP	FN
Normal scenario		
(fpr 0.1%; fnr 0.01%)	7.7 ±0.4	5.7 ±0.8
Noisy test scenario		
(fpr 1%; fnr 0.1%)	8.2 ±0.4	5.9 ±0.8
(fpr 10%; fnr 1%)	8.8 ±0.4	8.4 ±0.4
(fpr 25%; fnr 3%)	9.1 ±0.4	8.2 ±0.6
Stale phone scenario		
10% stale phones	8.0 ±0.4	6.1 ±0.4
20% stale phones	8.3 ±0.3	6.3 ±0.5
50% stale phones	8.2 ±0.3	6.9 ±0.3
Random policy	9.4 ±0.4	

Table 1: Testing the robustness of inference algorithms in the noisy circumstances of a new pandemic and society has *no time to waste*. In the noisy testing scenario, false positive rate (fpr) and false negative rate (fnr) go up to 25% and 3%, respectively. In the stale phone scenario, up to 50% of users might have stale updates in half of the time. Numbers correspond to percentages, population mean ± one standard deviation.

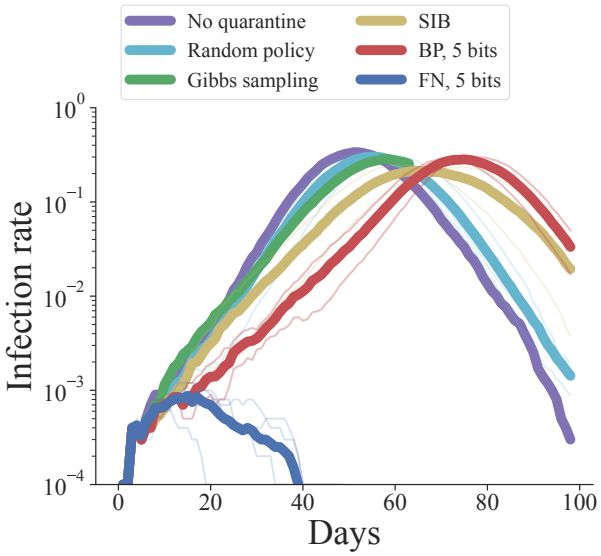

Figure 6: Infection rates on the OpenABM simulator. Parameters for the simulator follow population surveys in the United Kingdom and other academic research. Compared to BP or Gibbs sampling, FN results in lower infection rate. Thin lines indicate single realizations, thick lines correspond to population averages.

5.6 OpenABM-Covid19 simulator

Next, we ask the question *Do these results translate to a more realistic simulator?* The simulator used so far, (Herbrich et al., 2020), is quite simple, modeling society only as a collection of stochastic blocks. In contrast, the OpenABM-Covid19 simulator (Hinch et al., 2021) stratifies dynamics in nine age categories, six household categories, and models up to eleven different disease states. Our experiments follow the parameter settings of previous literature (Baker et al., 2021). Both BP and FN are used in this simulation with five 4-bit daily updates per contact (in contrast to Baker et al. (2021) and Herbrich et al. (2020) which use hundreds or thousands of daily updates per contact).

Figure 6 shows the results for various inference algorithms on the OpenABM-Covid19 simulator. We compare a simulation with random quarantining and simulations that use one of four inference algorithms. Each light line indicates an evaluation with a different random seed – thick lines indicate averages among multiple random seeds. The three inference algorithms use a similar communication budget: Gibbs takes ten samples, where each contact requires two bits, BP and FN each use five rounds of four bits each. For comparison, the SIB approach of Baker et al. (2021) is included, but note that this model is not quantized, thus requires a significantly larger amount of bits and update rounds, and has been optimized for a different testing and quarantining policy.

All algorithms improve upon a simulation without quarantine applied. Also, on average, all statistical inference algorithms result in lower infection rates than a random quarantine policy. This result seconds earlier work that statistical inference provides value to pandemic monitoring. While Gibbs, BP, and FN all use similar communication budgets, results show that FN obtains the lowest infection rates. Combined with the simplicity of Equation 7, we argue that FN is a practical inference algorithm for low-communication contact tracing.

Robustness setup	BP	FN
Normal scenario		
(fpr 0.1%; fnr 0.01%)	16.9 [13.8,17.4]	0.06 [0.05,0.1]
Noisy test scenario		
(fpr 1%; fnr 0.1%)	23.7 [22.2,24.8]	19.8 [0.8,21.3]
(fpr 10%; fnr 1%)	23.1 [22.7,25.0]	22.0 [21.2,23.2]
(fpr 25%; fnr 3%)	22.8 [22.0,24.1]	25.1 [23.8,25.9]
Stale phone scenario		
10% stale phones	20.6 [20.2,21.3]	0.08 [0.06,0.1]
20% stale phones	23.8 [23.3,24.1]	0.1 [0.09,0.9]
50% stale phones	26.2 [26.0,26.8]	6.3 [0.1,15.8]
Random	27.3 [26.7,27.6]	

Table 2: Testing the robustness of inference algorithms in the noisy circumstances. This table reproduces Table 1 for the Open-ABM simulator. Reported are the median and 20-80th percentile of twenty random runs.

6 Discussion

It is a curious human trait that they will only spring to action on existential threats like climate change and deadly pandemics when it is two minutes to twelve. We believe that at the beginning of a new pandemic that is potentially much more lethal than COVID-19, before the arrival of vaccines, our first and most effective line of defense is information technology. A question we have to ask is "why are we not preparing better for such an event through the development of intelligent contact tracing technology?". This paper continues important work that was initiated at the beginning of the COVID-19 pandemic but has lost momentum (Baker et al., 2021; Herbrich et al., 2020; Hinch et al., 2021; Vindegaard and Benros, 2020). We think there is no time to waste in developing this technology.

This work contributes to the literature by developing a practical algorithm that requires few daily messages. Requiring only few messages enables all the necessary security for such sensitive information. Moreover, a practical algorithm that requires only a few bits does not burden the communication network. Finally, its robustness enables inference in many scenarios, such as noisy tests and stale updates from unavailable phones. The proposed algorithm has been shown effective with only five messages of fewer than four bits, adhering to standards set by other contact tracing apps (Alsdurf et al., 2020; Apple and Google, 2020) and symptom-tracking apps (TCN, 2020; CoEpi, 2020). Comparing this communication cost to conventional often-used chat apps, our algorithm requires *less* communication than sending five times one alphabetical character to each contact of the recent three weeks.

Limitations Despite extensive evaluation, our research has two limitations: Simulations run for a population of 10 thousand people. Though related literature (Herbrich et al., 2020), also evaluates on 10 thousand people, experiments at millions or more might reveal new patterns and insights. Another limitation of our work is that contacts are assumed to be given. In practice, contact datasets could have false positives, and research is needed to see how this impacts inference.

Future work This research is an important topic, and we see two apparent avenues for future directions: conditional testing and differential privacy. With conditional testing, policies for test assignments could be improved or learned from data. Current experiments follow established work and rank users according to the posterior probability of infection. However, with scarce tests, testing an already infected person might provide less information compared to another individual where the prediction is more uncertain. Such a policy could be learned from data.

The final point addresses privacy concerns. Already, this paper studies decentralized algorithms where no central entity accrues information about individuals. However, also on privacy aspects we see potential improvements. When a user has few contacts, its disease score could be a direct reflection of select other individuals, thus diminishing their privacy. As such, the update function (e.g. Equation 4 or 7) could be made differentially private (Dwork and Roth, 2014), which we see as an important next step.

Acknowledgements

This work is financially supported by Qualcomm Technologies Inc., the University of Amsterdam and the allowance Top consortia for Knowledge and Innovation (TKIs) from the Netherlands Ministry of Economic Affairs and Climate Policy.

References

Alsdurf, H., Bengio, Y., Deleu, T., Gupta, P., Ippolito, D., Janda, R., Jarvie, M., Kolody, T., Krastev, S., Maharaj, T., Obryk, R., Pilat, D., Pisano, V., Prud'homme, B., Qu, M., Rahaman, N., Rish, I., Rousseau, J., Sharma, A., Struck, B., Tang, J., Weiss, M., and Yu, Y. W. (2020). COVI white paper. *arXiv*.

Anderson, R. M. and May, R. M. (1992). *Infectious diseases of humans: dynamics and control*. Oxford university press.

Apple and Google (2020). Privacy-preserving contact tracing. *apple.com/covid19/contacttracing/, (last accessed August 2022)*.

Baker, A., Biazzo, I., Braunstein, A., Catania, G., Dall'Asta, L., Ingrosso, A., Krzakala, F., Mazza, F., Mézard, M., Muntoni, A. P., et al. (2021). Epidemic mitigation by statistical inference from contact tracing data. *PNAS*.

Bay, J., Kek, J., Tan, A., Hau, C. S., Yongquan, L., Tan, J., and Quy, T. A. (2020). Bluetrace: A privacy-preserving protocol for community-driven contact tracing across borders. *Government Technology Agency-Singapore, Tech. Rep*.

Bestvina, I. (2020). Viratrace: Contact tracing infection risk estimate. *github.com/ViraTrace/InfectionModel (last accessed August 2022)*.

Biazzo, I., Braunstein, A., Dall'Asta, L., and Mazza, F. (2021). Epidemic inference through generative neural networks. *arXiv*.

Bishop, C. M. (2007). *Pattern recognition and machine learning*. Springer.

Boden, M., Zimmerman, L., Azevedo, K. J., Ruzek, J. I., Gala, S., Magid, H. S. A., Cohen, N., Walser, R., Mahtani, N. D., Hoggatt, K. J., et al. (2021). Addressing

the mental health impact of covid-19 through population health. *Clinical Psychology Review*.

Carinci, F. (2020). Covid-19: preparedness, decentralisation, and the hunt for patient zero. *British Medical Journal*.

Chan, J., Foster, D., Gollakota, S., Horvitz, E., Jaeger, J., Kakade, S., Kohno, T., Langford, J., Larson, J., Sharma, P., et al. (2020). Pact: Privacy sensitive protocols and mechanisms for mobile contact tracing. *arXiv*.

Cho, H., Ippolito, D., and Yu, Y. W. (2020). Contact tracing mobile apps for COVID-19: privacy considerations and related trade-offs. *arXiv*.

CoEpi (2020). Coepi: Community epidemiology in action. *www.coepi.org/, (last accessed August 2022)*.

Deb, T., Roy, A., Genc, S., Slater, N., Mallya, S., Kass-Hout, T. A., and Hanumaiah, V. (2020). the covid-19 simulator and machine learning toolkit for predicting covid-19 spread. *https://github.com/aws-samples/covid19-simulation, (last accessed August 2022)*.

Dwork, C. and Roth, A. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*.

for Disease prevention and control, E. C. (2021). Considerations on the use of self-tests for covid-19 in the eu/eea. *ECDC technical report, 17 March 2021*.

Grantz, K. H., Meredith, H. R., Cummings, D. A., Metcalf, C. J. E., Grenfell, B. T., Giles, J. R., Mehta, S., Solomon, S., Labrique, A., Kishore, N., et al. (2020). The use of mobile phone data to inform analysis of covid-19 pandemic epidemiology. *Nature communications*.

Herbrich, R., Rastogi, R., and Vollgraf, R. (2020). CRISP: A probabilistic model for individual-level COVID-19 infection risk estimation based on contact data. *arXiv*.

Hinch, R., Probert, W. J. M., Nurtay, A., Kendall, M., Wymant, C., Hall, M., Lythgoe, K. A., Cruz, A. B., Zhao, L., Stewart, A., Ferretti, L., Montero, D., Warren, J., Mather, N., Abueg, M., Wu, N., Legat, O., Bentley, K., Mead, T., Van-Vuuren, K., Feldner-Busztin, D., Ristori, T., Finkelstein, A., Bonsall, D. G., Abeler-Dörner, L., and Fraser, C. (2021). Openabm-covid19 - an agent-based model for non-pharmaceutical interventions against COVID-19 including contact tracing. *PLoS Computational Biology*.

Kaye, A. D., Okeagu, C. N., Pham, A. D., Silva, R. A., Hurley, J. J., Arron, B. L., Sarfraz, N., Lee, H. N., Ghali, G. E., Gamble, J. W., et al. (2021). Economic impact of covid-19 pandemic on healthcare facilities and systems: International perspectives. *Best Practice and Research Clinical Anaesthesiology*.

Kermack, W. O. and McKendrick, A. d. G. (1927). A contribution to the mathematical theory of epidemics. *Proceedings of the royal society of London*.

Koller, D. and Friedman, N. (2009). *Probabilistic Graphical Models - Principles and Techniques*. MIT Press.

Li, B. and Saad, D. (2021). Impact of presymptomatic transmission on epidemic spreading in contact networks: A dynamic message-passing analysis. *Physical Review E*.

Liu, J. S. (1996). Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statistics and computing*.

Lorch, L., Kremer, H., Trouleau, W., Tsirtsis, S., Szanto, A., Schölkopf, B., and Gomez-Rodriguez, M. (2004). Quantifying the effects of contact tracing, testing, and containment measures in the presence of infection hotspots. *ACM Transactions on Spatial Systems and Algorithms*.

Mathioudakis, M., Bonchi, F., Castillo, C., Gionis, A., and Ukkonen, A. (2011). Sparsification of influence networks. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Mehrjou, A., Soleymani, A., Abyaneh, A., Schölkopf, B., and Bauer, S. (2021). Pyfectious: An individual-level simulator to discover optimal containment polices for epidemic diseases. *arXiv*.

Myers, S. A. and Leskovec, J. (2010). On the convexity of latent social network inference. In *NeurIPS*.

Park, S., Choi, G. J., and Ko, H. (2020). Information Technology–Based Tracing Strategy in Response to COVID-19 in South Korea—Privacy Controversies. *Journal of the American Medical Association*.

Pearl, J. (1989). *Probabilistic reasoning in intelligent systems*. Morgan Kaufmann.

Raskar, R., Schunemann, I., Barbar, R., Vilcans, K., Gray, J., Vepakomma, P., Kapa, S., Nuzzo, A., Gupta, R., Berke, A., et al. (2020). Apps gone rogue: Maintaining personal privacy in an epidemic. *arXiv*.

Robert, C. P. and Casella, G. (2004). *Monte Carlo Statistical Methods*. Springer.

Rosen-Zvi, M., Jordan, M. I., and Yuille, A. L. (2005). The DLR hierarchy of approximate inference. *UAI*.

Rutten, P., Lees, M. H., Klous, S., Heesterbeek, H., and Sloot, P. (2022). Modelling the dynamic relationship between spread of infection and observed crowd movement patterns at large scale events. *Nature Scientific Reports*.

Sakai, Y. and Hukushima, K. (2016). Eigenvalue analysis of an irreversible random walk with skew detailed balance conditions. *Physical Review E*.

Stohr, J. J., Zwart, V. F., Goderski, G., Meijer, A., Nagel-Imming, C. R., Kluytmans-van den Bergh, M. F., Pas,

S. D., van den Oetelaar, F., Hellwich, M., Gan, K. H., et al. (2022). Self-testing for the detection of sars-cov-2 infection with rapid antigen tests for people with suspected covid-19 in the community. *Clinical Microbiology and Infection*.

TCN (2020). Tcn protocol. *github.com/TCNCoalition/TCN, (last accessed August 2022)*.

Troncoso, C., Payer, M., Hubaux, J.-P., Salathé, M., Larus, J., Bugnion, E., Lueks, W., Stadler, T., Pyrgelis, A., Antonioli, D., et al. (2020). Decentralized privacy-preserving proximity tracing. *arXiv*.

Vindegaard, N. and Benros, M. E. (2020). Covid-19 pandemic and mental health consequences: Systematic review of the current evidence. *Brain, Behavior, and Immunity*.

Vineetha Warriyar, K., Almutiry, W., and Deardon, R. (2020). Individual-level modelling of infectious disease data: Epiilm. *arXiv*.

Welling, M. (2004). On the choice of regions for generalized belief propagation. In *UAI*.

Welling, M. and Teh, Y. W. (2001). Belief optimization for binary networks: A stable alternative to loopy belief propagation. In *UAI*.

Wood, F., Warrington, A., Naderiparizi, S., Weilbach, C., Masrani, V., Harvey, W., Ścibior, A., Beronov, B., and Nasseri, A. (2020). Planning as inference in epidemiological models. *arXiv*.

Yedidia, J. S., Freeman, W. T., and Weiss, Y. (2000). Generalized belief propagation. In *NeurIPS*.

Yuille, A. L. (2002). CCCP algorithms to minimize the bethe and kikuchi free energies: Convergent alternatives to belief propagation. *Neural Computation*.

A Additional results

This supplementary material discusses additional topics to our paper titled *No time to waste: practical statistical contact tracing with few low-bit messages*. Sections A.1 and A.2 discuss derivations of the algorithms presented, Section A.3 presents additional results on low-update scenarios, and Sections A.4, A.5, and A.6 discuss three other analyses of Gibbs sampling in the contact tracing problems.

A.1 A detailed derivation of the Factorized Neighbor algorithm

This section presents a detailed derivation of the FN update equations, focusing on the backward messages. It will turn out that backward messages in both BP and FN follow a similar quadratic form for their calculation. These quadratic forms are compared between BP and FN to provide insight into how these algorithms incorporate information from contacts.

A.1.1 Notation

We briefly summarize notation of the algorithm and update equations: Each user, u, at each timestep, t, is represented with a random variable $z_{u,t}$ with domain $\{S, E, I, R\}$, indicating the Susceptible, Exposed, Infected and Recovered states. Likewise, the random variable, $o_{u,t}$, is the observation made of user u at timestep t. The indicator function $\mathbf{1}[z_u, \tau]$ yields 1 if user u is infected, in state I, at timestep τ, and 0 otherwise. Random variables for algorithms that operate on multiple timesteps are summarized with the notation $z_u = \{z_{u,t}\}_{t=0}^{T-1}$. Thus random variable $z_u \in \{S, E, I, R\}^T$.

The beliefs of a single variable $z_{u,t}$ or z_u are denoted with lower-case, e.g. $b_u(z_u)$. The joint belief over a set is denoted with upper-case, e.g. $B_{\{u,z\}}(z_u, z_v)$. Of special interest is the joint belief $B_{N(u)}(z_{N(u)})$. Here $N(u)$ indicates the neighbouring nodes of user u, also called the Markov blanket. Then the random variable $z_{N(u)} = \{z_v\}_{v \in N(u)}$.

A.1.2 Backward messages of FN

We start to write out the backward message in the contact on day 1 between user 0 and user 1. The derivation uses the extended PGM in Figure 12. Note that z_u is used as shorthand for the block of variables $\{z_{u,t}\}_{t=0}^{T-1}$.

$$
\begin{aligned}
b_0(z_0) &= E_{B_{N(0)}}\left[p(z_0|z_{N(0)}, \mathcal{O})\right] && (9)\\
&= E_{b_v}\left[p(z_0|z_v, \mathcal{O})\right] E_{b_1, b_v}\left[\frac{p(z_1|z_0, z_v, \mathcal{O})}{p(z_1|z_v, \mathcal{O})}\right] && (10)\\
&= E_{b_v}\left[p(z_0|z_v, \mathcal{O})\right] E_{b_1, b_v}\left[\frac{p(z_{1,2}|z_{1,1}, z_0, z_v, \mathcal{O})}{p(z_{1,2}|z_{1,1}, z_v, \mathcal{O})}\right] && (11)\\
&= E_{b_v}\left[p(z_0|z_v, \mathcal{O})\right] E_{b_1, b_v}\left[\frac{\sum_{z_c} p(z_{1,2}, z_c|z_{1,1}, z_0, \mathcal{O})}{\sum_{z_0', z_c'} p(z_{1,2}, z_c'|z_{1,1}, z_0', \mathcal{O}) p(z_0'|z_v, \mathcal{O})}\right] && (12)\\
&\approx E_{b_v}\left[p(z_0|z_v, \mathcal{O})\right] \underbrace{E_{b_1, b_v, b_c}\left[\frac{p(z_{1,2}|z_c, z_{1,1}, z_0, \mathcal{O})}{E_{b_c(z_c')}\left[\sum_{z_0'} p(z_{1,2}|z_{1,1}, z_0', z_c', \mathcal{O}) p(z_0'|z_v, \mathcal{O})\right]}\right]}_{\text{Message from user 1 to user 0}} && (13)\\
&\approx E_{b_v}\left[p(z_0|z_v, \mathcal{O})\right] \underbrace{E_{b_1, b_c}\left[\frac{p(z_{1,2}|z_{1,1}, z_0, z_c, \mathcal{O})}{E_{b_0(z_0'), b_c(z_c')} p(z_{1,2}|z_{1,1}, z_0', z_c', \mathcal{O})}\right]}_{\text{Message from user 1 to user 0}}. && (14)
\end{aligned}
$$

Here $B_{N(0)}$ are the beliefs over neigbors of user 0. Under the Factorized Neighbor assumptions, these are factored as $B_{N(0)} = b_v(z_v) b_1(z_1)$.

From Equation 11 to 12, the user c is included as the factors for user 1 depend on user c (c.f. Figure 12). However, in decentralized inference, neither does user c know about user 0, nor does user 1 know about user v, as they are not direct neighbors. Therefore, both posteriors are approximated with beliefs, yielding a practical and tractable calculation. Note that the authors of Rosen-Zvi et al. (2005) originally formulated FN for undirected factors in an Ising model, and thus did not have this approximation.

In Equation 14, the expectation over b_1 can be computed as a quadratic form. The elements within square brackets form a matrix and depend only on $z_{1,1}$, $z_{1,2}$, and $b_{c,1}$. Therefore, by writing the beliefs of user 1 as vectors, $\boldsymbol{b_{1,1}}$ and $\boldsymbol{b_{1,2}}$, the calculation becomes:

$$b_0(z_0) = E_{b_v}\left[p(z_0|z_v, \mathcal{O})\right] \boldsymbol{b_{1,1}^T A}(z_0, b_0, b_c)\boldsymbol{b_{1,2}}. \tag{15}$$

Table 3 compares the above matrix, $\boldsymbol{A}(z_0, b_0, b_c)$, to a similar quadratic form used for backward messages in BP, derived by Herbrich et al. (2020) and restated in A.1.3.

Transition	$\boldsymbol{A}(z_0, \mu)$ in BP	$\boldsymbol{A}(z_0, b_0, b_c)$ in FN
Infeasible transition	0	0
$S \to E$	$E_{\mu_{z_{c,1}\to f_{1,2}}}\left[p(z_{1,2} = E\|z_{1,1} = S, z_{0,1}, z_{c,1})\right]$	$\frac{E_{b_c(z_c)}[p(z_{1,2}=E\|z_{1,1}=S,z_0,z_c)]}{E_{b_0(z_0')}[E_{b_c(z_c)}[p(z_{1,2}\|z_{1,1},z_0',z_c)]]}$
$S \to S$	$E_{\mu_{z_{c,1}\to f_{1,2}}}\left[p(z_{1,2} = S\|z_{1,1} = S, z_{0,1}, z_{c,1})\right]$	$\frac{E_{b_c(z_c)}[p(z_{1,2}=S\|z_{1,1}=S,z_0,z_c)]}{E_{b_0(z_0')}[E_{b_c(z_c)}[p(z_{1,2}\|z_{1,1},z_0',z_c)]]}$
$E \to E$	g	1
$I \to I$	h	1
$R \to R$	1	1

Table 3: Elements of the quadratic products in Equations 14 and 19. The backward message for both BP and FN is proportional to a quadratic form, whose elements this table compares. In shorthand, the BP message is proportional to $\boldsymbol{\mu^T A \mu}$, with $\boldsymbol{\mu}$ message vectors; the FN message is proportional to $\boldsymbol{b^T A b}$, with $\boldsymbol{b}$ belief vectors. Details are in Appendix A.1. For brevity, all conditioning on $\mathcal{O}$ is omitted.

A.1.3 Comparison to backward messages of BP

For comparison, we write the backward messages for BP in the same graph, depicted in Figure 1. The backward message prescribed by BP also comprises a quadratic form. Note that these derivations were made by (Herbrich et al., 2020), and we restate the equations here to highlight the connection with our newly derived FN equations.

We adhere to factor graph notation from (Koller and Friedman, 2009) and write forward messages from factor to variable.

$$\mu_{f_{1,0}\to z_{1,0}}(z_{1,0}) = p(z_{1,0}) \tag{16}$$
$$\mu_{z_{1,0}\to f_{1,1}}(z_{1,0}) = \mu_{f_{1,0}\to z_{1,0}}(z_{1,0}) \tag{17}$$
$$\mu_{f_{1,1}\to z_{1,1}}(z_{1,1}) = \boldsymbol{A}^T \mu_{z_{1,0}\to f_{1,1}} \tag{18}$$
$$\vdots \quad \vdots$$

Here, $p(z_{1,0})$ is the prior over user 0 for day 0; $f_{u,t}$ is the factor for user u on day t, i.e. $p(z_{u,t}|z_{u,t-1}, \mathcal{O}, \{z_{c,t-1}\})$; and A is the transition matrix of the Markov chain.

Implementing Equation 4, the backward message prescribed by BP will be then:

$$\mu_{f_{1,2}\to z_{0,1}}(z_{0,1}) = \sum_{z_{1,1}}\sum_{z_{1,2}}\sum_{z_{c,1}} p(z_{1,2}|z_{1,1}, z_{0,1}, z_{c,1})\mu_{z_{c,1}\to f_{1,2}}(z_{c,1})\mu_{z_{1,1}\to f_{1,2}}(z_{1,1})\mu_{z_{1,2}\to f_{1,2}}(z_{1,2}).$$

The summation over the messages from contacts can be written as an expectation. From this equation, the quadratic form will arise as follows:

$$\mu_{f_{1,2}\to z_{0,1}}(z_{0,1}) = \sum_{z_{1,1}}\sum_{z_{1,2}} E_{\mu_{z_{c,1}\to f_{1,2}}}[p(z_{1,2}|z_{1,1}, z_{0,1}, z_{c,1})]\mu_{z_{1,1}\to f_{1,2}}(z_{1,1})\mu_{z_{1,2}\to f_{1,2}}(z_{1,2}). \tag{19}$$

We then write the messages as vectors, $\boldsymbol{\mu}_{z_{1,1}\to f_{1,2}}(z_{1,1})$ $\boldsymbol{\mu}_{z_{1,2}\to f_{1,2}}(z_{1,2})$, and the matrix will depend on $\mu_{z_{c,1}\to f_{1,2}}$.

Connection between BP and FN messages Table 3 compares the elements in the quadratic form between BP and FN. Both methods have a quadratic form of the shape $\boldsymbol{x}^{\boldsymbol{T}} A\boldsymbol{x}$. In BP, the vectors, x, are the messages, $\mu_{z_u\to f_s}$; in FN, the vectors are the beliefs, $b_u(z_u)$. The most striking difference in Table 3 is the $S \to E$ transition. For any user, this transition could happen by a virus transmission from another user. Sending a backward message to $z_{0,1}$, the BP calculation 'excludes' that information by omitting the incoming message, prescribed by the backslash, $\backslash$, in Equation (4). Correspondingly, FN uses the forward message, but for normalization in the denominator. These are two different calculations for decentralized inference. Other differences are the $E \to E$ and $I \to I$ transitions, where BP uses the model parameter and FN the value 1. However, that may simply be due to BP using messages and FN using beliefs.

A.2 Why not use variational inference?

Both Gibbs, BP, and FN require an expectation over neighbors' states, due to Equation 2. The calculation of this expectation could be prohibitively large. Most users might have few daily contacts, but some users could occasionally have many contacts, for example, when visiting a music concert or sports game. As such, we highlight in this subsection how that calculation in Gibbs, BP, and FN grows linearly, while in variational inference the calculation grows exponentially.

- Belief Propagation, linear:

$$\beta(z_u) \propto E_{\mu_{z_{v_0,t_c}\to f_s}, \mu_{z_{v_1,t_c}\to f_s}\cdots\mu_{z_{v_{C-1},t_c}\to f_s}}\left[1-(1-p_0)\prod_{c=0}^{C-1}(1-p_1)^{\mathbf{1}[z_{v_c,t_c}==I]}\right]$$

- Factorized Neighbors, linear:

$$b(z_u) \propto E_{b(z_{v_0,t_c})b(z_{v_1,t_c})\cdots b(z_{v_{C-1},t_c})}\left[1-(1-p_0)\prod_{c=0}^{C-1}(1-p_1)^{\mathbf{1}[z_{v_c,t_c}==I]}\right]$$

- Gibbs sampling, linear:

$$p(z_u^{(k+1)}|z_{\neg u}^{(k)}) \propto \left[1-(1-p_0)\prod_{c=0}^{C-1}(1-p_1)^{\mathbf{1}[z_{v_c,t_c}^{(k)}==I]}\right]$$

- Variational Inference, exponential:

$$q(z_u) \propto E_{q(z_{v_0})q(z_{v_1})\cdots q(z_{v_{C-1}})}\left[\log(1-(1-p_0)\prod_{c=0}^{C-1}(1-p_1)^{\mathbf{1}[z_{v_c,t_c}==I]})\right]$$

The above itemization of equations shows that the expectation required for VI cannot be simplified from an exponential to a linear computation due to the $\log(.)$-operation.

A.3 Statistical inference with few updates

Driving home the point that FN works with few messages and thus is practical, we run the simulators using down to 1 daily update round. Figure 7 shows results on the CRISP simulator; Figure 8 shows results on the Open ABM simulator. Both results have lower than random peak infection rates with as low as two daily messages per contact. Such a low amount of messages will enable all the necessary security layers for the sensitive information that is being communicated.

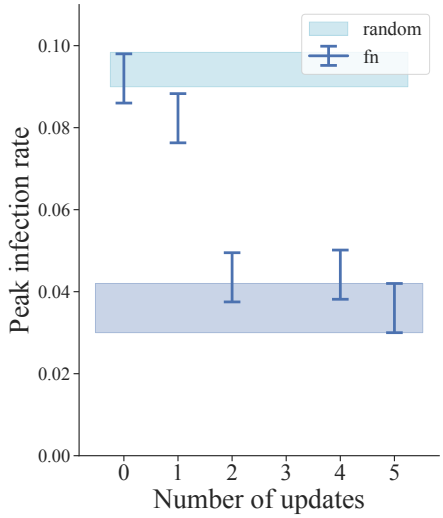

Figure 7: Simulating FN algorithm on the CRISP simulator with as low as zero or one daily update. This simulation is five bits per message.

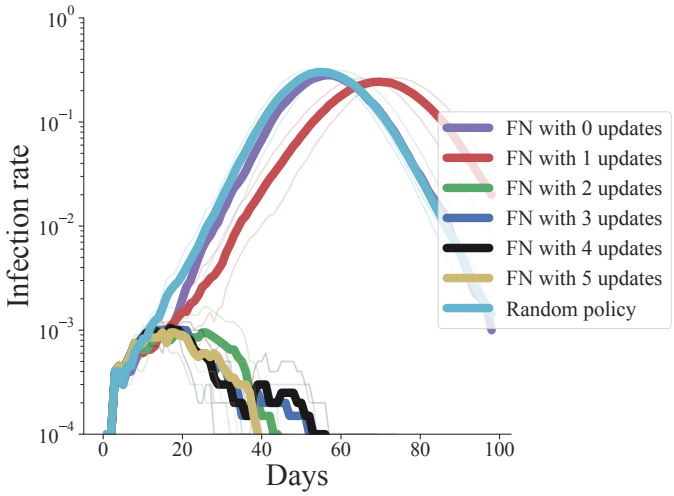

Figure 8: Simulating the FN algorithm on the OpenABM-Covid19 simulator with as low as zero or one daily update. This simulation is five bits per message.

A.4 Mixing of Gibbs chains: loopy graph

An additional analysis of the convergence of Gibbs sampling is made. The graphical model defined in Section 3 has loops, and we ask the question *How do these loops relate to the mixing of Gibbs chains and the convergence of the estimated marginals?* To this end, we calculate the Mean Absolute Error (MAE) between marginals estimated from two different Gibbs chains(c.f. Figure 4). The MAE is plotted per node against the shortest cycle length that includes the node. Figure 9 shows the resulting correlation including the 20th and 80th percentile in shaded blue. We see that nodes involved in shorter cycles (less than length 50) have a higher discrepancy and nodes involved in longer cycles have lower discrepancy. Other research exists to address the problem of loopy graphs, such as cluster-graphs (Welling, 2004), other formulations of blocked Gibbs (Herbrich et al., 2020), or hybrid approaches.

A.5 Examples of Gibbs chains

This section discusses the slow mixing of Gibbs chains. Figure 2 showed that inference with Gibbs sampling with as many as thousands of samples does poorly in mitigating a high pandemic peak. Subsequently, Figure 4 showed that Gibbs chains mix slowly in expectation. Here, we seek a more detailed answer and look at a particular user in a particular Gibbs chain. We ask the question: *how do posterior estimates differ per chain?* Figure 10 shows one such example. Note that this example is handpicked to demonstrate the behavior outlined in the following.

Figure 10 shows posterior estimates after 50 samples from five different Gibbs chains. The final row shows the FN posterior obtained with five 5-bit messages per contact. The Gibbs estimates differ wildly per chain. The difference is most pronounced in 'the point of transmission', the point where the state shifts from S to E. In the first chain of Figure 10, the transmission is estimated at day seven, evidenced by a decrease of the blue line, representing state S. However, the second chain estimates the transmission at day eight, and the third chain estimates the transmission as late as day 11. In other words, these five chains have different posterior estimates when a virus transmission happens. This variance could explain why Gibbs sampling with a low number of samples performs poorly in an experiment like Figure 2.

A.6 Analysing eigenvalues of the Gibbs transition

In this section, we seek to understand the mixing of Gibbs chains on the model given by Equation 1. The graph may be loopy (`A` contact with `B`, `B` contact with `C`, `C` contact with `A`), which makes inference difficult (Koller and Friedman, 2009). We will view Gibbs sampling through the lens of its states' transition matrix. Here, the domain of states is discrete (either in state S or not, either in state E or not, etc.). Thus (blocked) Gibbs sampling can be seen as a discrete random walk on

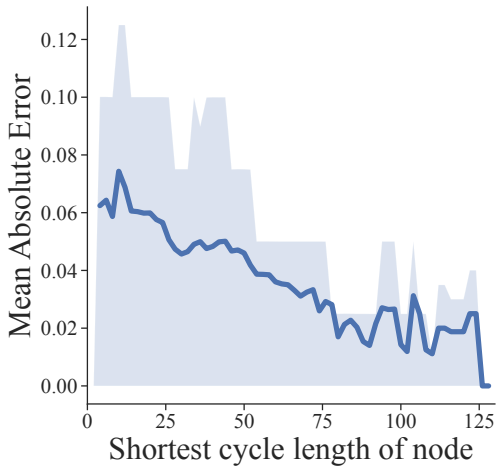

Figure 9: Mean Absolute Error between Gibbs chains, against shortest cycle length around a node. Individual nodes of the PGM (c.f. Figure 1) are compared on the discrepancy of two Gibbs chains, y-axis, and the shortest cycle around a node, x-axis. This negative correlation indicates that nodes with shorter cycles have a higher discrepancy.

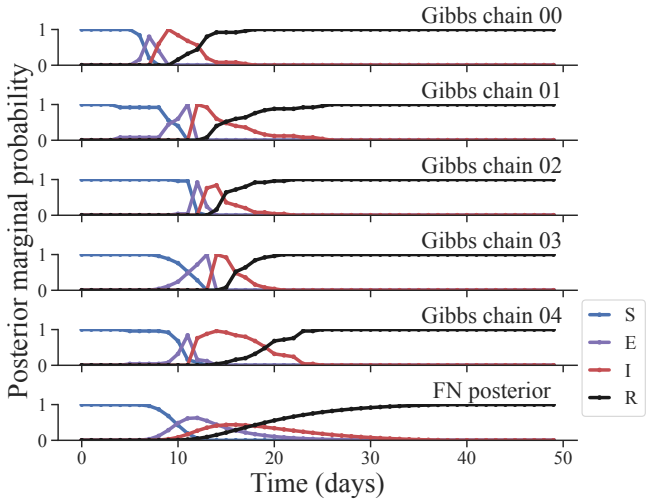

Figure 10: Different poorly-mixing Gibbs chain. The figure shows the posterior marginal estimated from five different Gibbs chains. Estimates differ wildly per Gibbs chain. For example, the first chain estimates the state change from S to E at day 7, the second chain estimates at day 8, and the third chain estimates this as late as day 11. This analysis follows from 50 Gibbs samples, which explains why Gibbs sampling with a low amount of samples performs badly in an experiment like Figure 2.

the set of all states.

With thousands of Gibbs samples, the transition matrix is multiplied as many times. The largest eigenvalue of the transition matrix has eigenvalue 1, as Gibbs sampling has a stationary distribution. Therefore, we analyze the 'spectral gap', which is 1 minus the second largest eigenvalue. A large spectral gap will indicate fast mixing (Sakai and Hukushima, 2016; Liu, 1996).

Consider a graph with two users. The transition matrix between each possible state follows:

$$T\left(\begin{bmatrix} z_0' \\ z_1' \end{bmatrix} \to \begin{bmatrix} z_0'' \\ z_1'' \end{bmatrix}\right) = p(z_0''|z_1')p(z_1''|z_0'') \tag{20}$$

$$T\left(\begin{bmatrix} z_0'' \\ z_1'' \end{bmatrix} \to \begin{bmatrix} z_0' \\ z_1' \end{bmatrix}\right) = p(z_0'|z_1'')p(z_1'|z_0'). \tag{21}$$

This matrix can be written as the eigenvalue decomposition:

$$A = Q\Lambda Q^{-1} \tag{22}$$

Therefore $A^{(t)} = Q\Lambda^{(t)}Q^{-1}$. Hence, as the largest eigenvalue is 1, the second largest eigenvalue will determine how 'slow' (in terms of t), transients decay to zero. Thus the spectral gap determines how 'fast' the Gibbs sampler reaches its stationary distribution.

Figure 11 plots the second largest eigenvalue for the two user graph when gradually adding edges. The fifth edge forms the first directed loop in the graph. Correspondingly, the spectral gap decreases at the fifth edge. This shows that directed loops cause slower mixing of the Gibbs chain.

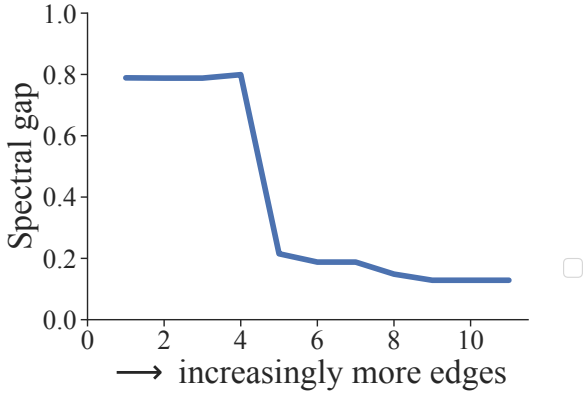

Figure 11: The spectral gap of Gibbs transition determines how 'fast' the chain mixes and how fast the expected posterior marginals converge. The spectral gap is plotted on the y-axis as more edges are added to a two-user graph, similar to Figure 1. The fifth added edge introduces the first directed loop. Correspondingly, we observe that the value of the spectral gap decreases, indicating slower mixing of Gibbs chains.

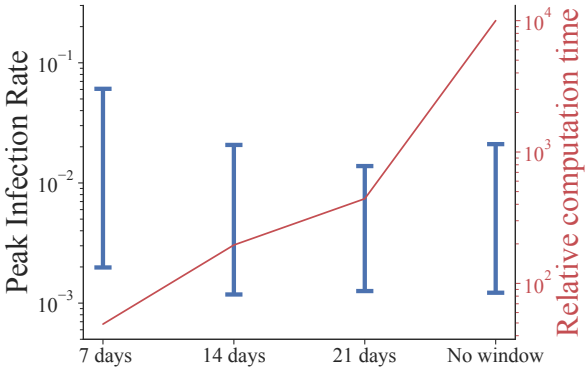

Figure 13: This figure reproduces Figure 5 for the OpenABM-Covid19 simulator. Inference with windows of length 14 or 21 yield significant savings in required compute, yet result in similar peak infection rates as inference without window. Error bars indicate the 20th and 80th percentile.

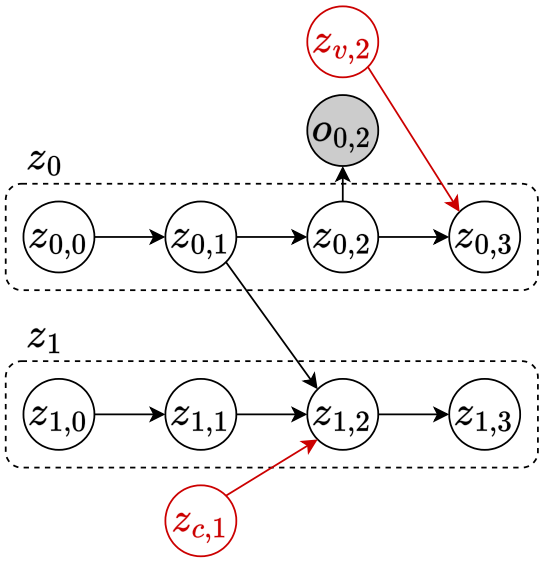

Figure 12: The PGM from 1, extended with two contacts. This graph is used to clarify the derivation in Section A.1.

B Experimental settings

This section highlights experimental settings for the two main experiments. The code to reproduce the experimental results can be found at `http://github.com/QUVA-Lab/nttw`.

This CRISP-simulator follows the simulation of Herbrich et al. (2020) and uses a stochastic block model. A population of ten thousand users consists of a hundred blocks. Users are fourty times more likely to have a contact within a block than between blocks. On a global level, the contacts have a pareto-distribution, where 20% of users make 80% of the contacts and vice versa. A spontaneous infection occurs with probability $\frac{1}{1000}$, an infected contact, in I state, transmits the infection with probability $p_1 = \frac{1}{10}$. In contrast to ABM, the experiments on CRISP do not have model specification. Both the g and h parameters in Equation 1 are set to $g = \frac{1}{5}$ and $h = \frac{1}{6}$. The simulations for Figure 2 and Table 1 run for 50 days, when most of the peaks in infection rates happen. Only Figure 3 displays 150 days to highlight the dynamics.

The OpenABM-Covid19 simulator has a parameter file comprising 150+ parameters. Our parameter file can be found in the Github. We use the same simulator parameters as Baker et al. (2021), which we verified via email correspondence. Note, however, that we follow the observation model of Herbrich et al. (2020), where individuals test positive only in the I state. Due to time limit, we have not been able to fine-tune model parameters, and use values $p_0 = \frac{1}{1000}$, $p_1 = \frac{1}{10}$, $g = \frac{1}{5}$ and $h = \frac{1}{6}$. All simulations with the OpenABM-Covid19 simulator run for 100 days. Both testing and quarantining are assumed by *highest posterior score of infection*, and, like (Baker et al., 2021), a positive tested user is not tested again in the same time window.

Unless otherwise stated, experiments are run with 4 bits quantization, i.e. 16 uniform quantization levels. The quantization scheme is highlighted in Section 4.3.