DNA: DIFFERENTIALLY PRIVATE NEURAL AUGMENTATION FOR CONTACT TRACING

Rob Romijnders[1], **Christos Louizos**[2], **Yuki M. Asano**[1], **Max Welling**[1]
[1]University of Amsterdam [2]Qualcomm AI research

ABSTRACT

The COVID19 pandemic had enormous economic and societal consequences. Contact tracing is an effective way to reduce infection rates by detecting potential virus carriers early. However, this was not generally adopted in the recent pandemic, and privacy concerns are cited as the most important reason. We substantially improve the privacy guarantees of the current state of the art in decentralized contact tracing. Whereas previous work was based on statistical inference only, we augment the inference with a learned neural network and ensure that this neural augmentation satisfies differential privacy. In a simulator for COVID19 even at $\varepsilon = 1$ per message, this can significantly improve the detection of potentially infected individuals and, as a result of targeted testing, reduce infection rates. This work marks an important first step in integrating deep learning into contact tracing while maintaining essential privacy guarantees.

1 INTRODUCTION

The COVID19 pandemic had enormous consequences (Kim et al., 2022; Kaye et al., 2021; Boden et al., 2021; Vindegaard & Benros, 2020). Contact-tracing algorithms could make early predictions of virus carriers, signaling individuals to get tested and thereby reducing the spread of the virus (Baker et al., 2021). However, population surveys show that privacy concerns are among the primary reasons for the low adoption of these algorithms (Jones et al., 2021; Gao et al., 2022; Walrave et al., 2022).

Recent work introduces a differential privacy (DP) solution against a privacy attack on contact tracing algorithms (Romijnders et al., 2024). This method, however, uses only a statistical model with a corresponding inference method to make predictions of infectiousness (Rosen-Zvi et al., 2005). We propose augmenting the inference steps with a learned neural network. Such a 'neural augmentation' can learn patterns in the data that are not captured by the statistical model. This builds on a line of work about augmenting statistical updates with learnable functions (Satorras & Welling, 2021; Lønning et al., 2019; Gregor & LeCun, 2010). We ensure differential privacy for the neural augmentation and name the method Differentially private Neural Augmentation (DNA).

In the attack scenario, an adversary tries to infer the private state of a victim. The model predicts for every user on every day a risk score, named COVIDSCORE, and it is this private score that the attacker tries to infer. We quote from previous work (Romijnders et al., 2024):

> 'An adversary wants to determine the COVIDSCORE of a victim. The adversary installs the app and only makes contact with the victim. The next day, the adversary observes a change in their COVIDSCORE. This change is due to the victim, and the adversary reconstructs the COVIDSCORE of the victim.'

Our goal is to achieve better predictions of the COVIDSCORE for an algorithm that is DP against this attack. Injecting noise is the default method for guaranteeing DP, but the noise makes the predictions worse. In order to motivate neural augmentation, we identify a hierarchy of methods to achieve DP. Whereas previous work analyzed contact tracing in terms of individual messages per user (level 1) and multiple messages per day (level 2), our analysis considers the set of all messages in a window of multiple days (level 3). The analysis reveals that the current algorithm has a bounded sensitivity, which motivates us to propose a neural augmentation module with a similar bound on the sensitivity.

In total, we make the following contributions:

- We show how statistical inference can be augmented with a neural network such that the combined prediction satisfies differential privacy. We are the first to bridge the promising field of neural augmentation with differential privacy.
- We identify a novel hierarchy of privacy in contact tracing and provide a theoretical proof of differential privacy at the most general level of the hierarchy.
- Both methods are tested on a widely used simulator. Even in the challenging situation of noisy tests, or agents not following the protocol, our method significantly reduces the number of simultaneously infected individuals, which is a key marker for pandemic mitigation.

The code for running the experiments will be open-sourced upon acceptance.

2 Related work

As this work brings together multiple fields, we overview related literature in four areas: differential privacy, neural augmentation, Lipschitz-constrained neural networks, and the application area of statistical contact tracing.

We use differential privacy (DP) as the main method to quantify the privacy of a statistical algorithm. A good overview of DP is the book by Dwork & Roth (2014). Differential privacy has been used in fields such as mean estimation (Dwork et al., 2006), deep learning (Abadi et al., 2016), and statistical query answering (Goryczka & Xiong, 2015). Moreover, DP is named as a promising area for federated learning research (Kairouz et al., 2021). Another reason for using DP is the use of building blocks like the post-processing property (Dwork & Roth, 2014) and advanced composition theorems (Dwork et al., 2010; Abadi et al., 2016; Ponomareva et al., 2023).

Neural augmentation is a method for augmenting statistical/physical algorithms with learnable neural networks. This has been studied in application areas such as sparse coding (Gregor & LeCun, 2010), MRI reconstruction (Lønning et al., 2019), and error correction codes (Satorras & Welling, 2021). We are the first to combine neural augmentation and differential privacy.

For the DP guarantee, we will make use of neural networks with constrained Lipschitz constant for the input data (defined in Section 4). The concept of Lipschitz neural networks has been studied as a method in adversarial robustness (Farnia et al., 2019), and to optimize a constrained family of models in Generative Adversarial Networks (Goodfellow et al., 2014; Miyato et al., 2018). Previous works use Lipschitz-constrained models for differentially private learning (Chaudhuri et al., 2011; Béthune et al., 2023; Minami et al., 2016) and data analysis (Jha & Raskhodnikova, 2013). In our case, we need a Lipschitz constraint with respect to the input data, and we follow previous work (Jha & Raskhodnikova, 2013) to apply this to neural augmentation.

Contact tracing has been shown to be effective in mitigating a pandemic outbreak (Jenniskens et al., 2021). Previous work shows that models based on statistical algorithms can outperform traditional methods (Baker et al., 2021; Herbrich et al., 2020). However, population studies make clear that privacy concerns are a major reason for not using contact tracing algorithms (Jones et al., 2021; Gao et al., 2022; Walrave et al., 2022). A recent work introduced a DP alternative for statistical contact tracing (Romijnders et al., 2024). That work, however, a) considers only a privacy composition per day which is only level 2 of our hierarchy and b) uses only classical statistical updates. In this work, we propose a DP method in a higher level of privacy hierarchy and show that, combined with neural augmentation, this can substantially decrease the peak impact of the infection.

3 Method

We provide a background on the statistical model and prove differential privacy for each level of the hierarchy. Appendix A.1 provides an overview of the notation used in this paper.

3.1 Statistical model

We will formulate a common statistical model for contact tracing. A Markov chain of random variables $z_{u,t}$ models the disease progression for a user u at timestep t. Each such variable takes on one of four disease states, S, E, I, R, for the Susceptible, Exposed, Infected, and Recovered state (Kermack & McKendrick, 1927; Anderson & May, 1992). The dynamics of this Markov Chain are described in Equation 7. The only non-scalar transition function is the transition between state S and state E:

$$P(z_{u,t+1} = E | z_{u,t} = S, z_{N(u,t)}) = 1 - (1 - p_0)(1 - p_1)^{|\{z_c \,\in\, z_{N(u,t)} :\, z_c = I\}|} \tag{1}$$

Parameter p_1 indicates the probability of transmitting the virus upon contact, and its value is set to a value from previous literature (Romijnders et al., 2023; Herbrich et al., 2020). A user can have contact with multiple other users on a particular day. As such, $z_{N(u,t)}$ denotes the set of random variables of all contacts of user u at time step t.

The actual observation is a test for COVID19 which can have a false positive or false negative outcome regarding the underlying state. The data set of observations is $D_{\mathcal{O}} = \{o_{u_i,t_i}\}_{i=1}^{O}$, which are O observations, each with an outcome $\{0, 1\}$ for user u_i at time step t_i. The observation model follows previous literature (Herbrich et al., 2020) and is stated in Equation 10. We denote the False Positive rate (FPR) by β and denote the False Negative rate (FNR) by α.

3.2 Statistical messages

The inference for the statistical SEIR model consists of sending decentralized messages between users. We follow the algorithm of (Romijnders et al., 2024) and run the Factorised Neighbors (FN) algorithm (Rosen-Zvi et al., 2005). This algorithm has been shown to be effective for decentralized contact tracing and its updates are amenable to the privacy analysis that we make in this paper.

We can write the FN inference algorithm as a function of the incoming messages. Each message is a number in the range $[0, 1)$ and denotes the COVIDSCORE, $\phi_{c,t}$, of user c at timestep t. On each day t, user u has C_t contacts with a user that will be denoted by the relative index $1, 2, \cdots, C_t$.

3.3 Differential privacy

The conventional definition of differential privacy is used, (ε, δ)-DP (Dwork & Roth, 2014). For every $\varepsilon > 0$, $\delta \in [0, 1]$, a function $f(\cdot)$, for any outcome Φ in the range of $f(\cdot)$, and two adjacent data sets D, D' that have at most one element different, the following constraint holds:

$$p(f(D) \in \Phi) \leq e^{\varepsilon} p(f(D') \in \Phi) + \delta \tag{2}$$

We define two data sets, D, D', as adjacent when the COVIDSCORE of one of the contacts has a different value. A dataset D is a collection of COVIDSCORE that C contacts send at particular days, $D = \{(\mu_i, t_i)\}_{i=1}^{C}$. Each $t_i \in \{1, 2, \cdots, T-1\}$ is a particular day when a contact occurs, where T is the window length. The values for t_i are public information from the point of view of the attack model, i.e. an attacker could know on what day a particular contact was established, but the COVIDSCORE of that contact should remain private. Therefore, DP is defined between two adjacent datasets, where the value of one message by a contact, μ_i, is replaced by another value μ_i'. The largest change in the output of an algorithm then defines the sensitivity:

$$\Delta = \max_{\mu_1, \mu_1'} \left\| f\big(\{(\mu_1, t_1)\} \cup D \big) - f\big(\{(\mu_1', t_1)\} \cup D \big) \right\| \qquad \forall\, D. \tag{3}$$

For a particular sensitivity, the Gaussian mechanism by (Dwork & Roth, 2014) prescribes the standard deviation of additive Gaussian noise such that the output of the algorithm satisfies (ε, δ)-DP.

DP analysis per message (level 1): If each individual message is noised, by the post-processing property of DP, the entire prediction function is DP. We add Gaussian noise according to the Gaussian mechanism to ascertain DP (Dwork & Roth, 2014; Romijnders et al., 2024).

DP analysis of messages per day (level 2): The statistical model deals with messages per day in a product, highlighted in Equation 11. Previous work suggests adding log-normal noise to each message and provides the corresponding DP analysis (Romijnders et al., 2024).

DP analysis of messages per multiple days (level 3): A novel contribution of this work is that we generalize the hierarchy of privacy on one more level. We establish a bound on the sensitivity of the prediction function in the presence of multiple contacts on multiple days. Based on the sensitivity, we add Gaussian noise according to the Gaussian mechanism to ensure DP. The sensitivity of FN is formalized in the following theorem.

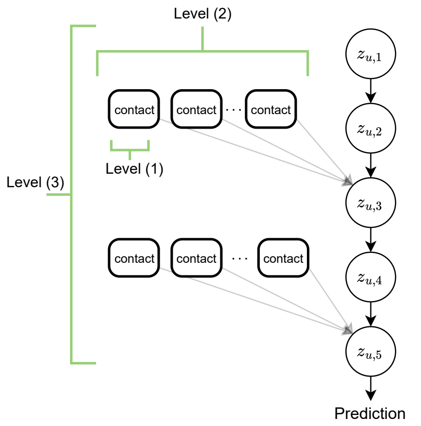

Figure 1: Three levels of DP analysis.

Theorem 1. *Following notation in Section 3 and assuming that each message μ_i is bounded in the interval $[0, \gamma_u]$, for any two adjacent datasets as defined in Equation 3, the sensitivity for the FN inference function, $F(\cdot)$, is defined by:*

$$\Delta = \max_{\mu_1, \mu_1' \in [0, \gamma_u]} \left\| F\left(\{(\mu_1, t_1)\} \cup D \right) - F\left(\{(\mu_1', t_1)\} \cup D \right) \right\| \leq p_1 \gamma_u \qquad \forall\, D. \tag{4}$$

Proof sketch: Each contact has a probability p_1 of transmitting the virus during a contact, c.f. Equation 1, and the value of the message is in the range 0 to γ_u. For every sequence of health states $S \rightarrow E \rightarrow I \rightarrow R$, the probability can only increase p_1 for a message value up to γ_u, of which the output is a convex combination. The sensitivity of $p_1 \gamma_u$ follows. See Section A.3 for the full proof. □.

This method modifies the DPFN algorithm based on bounding the sensitivity. Therefore, the experimental results will refer to this method as DPFN-sensitivity, or DPFN-S for short.

4 Neural Augmentation

We propose to improve the statistical predictions with neural augmentation. The current SEIR model has only four modifiable parameters and we hypothesize that a neural network can learn more complex patterns from data (Satorras & Welling, 2021; Lønning et al., 2019). Theorem 1 shows that the FN function has a sensitivity of p_1, typically around 0.02 (Hinch et al., 2021). However, such a sensitivity is only 2% of the output range, which is $\phi_{u,t} \in [0, 1]$. This shows that even a function with relatively low sensitivity can predict infectiousness. Therefore, we aim to learn a neural augmentation with similarly low sensitivity to augment the statistical predictions while maintaining privacy.

To obtain a bound for the sensitivity of a neural network, we use Lipschitz-constrained neural networks (Béthune et al., 2023). A function $g : \mathbb{R}^m \rightarrow \mathbb{R}^n$ has Lipschitz constant l if for every $x, y \in \mathbb{R}^m$ we have $\|g(x) - g(y)\|_2 \leq l \|x - y\|_2$. This constraint is achieved when the gradient norm is upper bounded, $\sup_x \|\nabla_x g(x)\| \leq l$ (Béthune et al., 2023).

The Lipschitz constant can be decomposed for a neural network into the Lipschitz constants of its layers. For a function $g(x) = g_1 \circ g_2 \circ \cdots \circ g_H(x)$ based on H composite functions, denote by l_h the Lipschitz constant of layer h. Then the Lipschitz constant of the composite function, $g(\cdot)$, is:

$$l = l_1 \times l_2 \times \cdots \times l_H. \tag{5}$$

For linear layers in the neural network, the Lipschitz constant equals the spectral norm, which will be restricted during training (Miyato et al., 2018; Bartlett et al., 2017). We also use an activation function with a bounded gradient, such as the Rectified Linear Unit (Nair & Hinton, 2010).

The neural network takes, as features, the messages that are sent to the agent by its contacts and predicts the infectiousness for that particular agent on that day. As there can be a variable number of messages and the prediction is invariant under permutation, we use a DeepSet model (Zaheer et al., 2017). For each contact, we denote the stacked feature vector $x_i = [\mu_i, t_i]^T$. Each feature vector is mapped to a representation by a neural network $g^{(1)}(\cdot)$, and, after averaging, a second neural network makes the prediction, $g^{(2)}(\cdot)$. The total neural network model is as follows:

$$\phi = G_\theta(\{(\mu_i, t_i)\}_{i=1}^{C_T}) = g_\theta^{(2)}\left(\quad \frac{1}{C} \sum_i g_\theta^{(1)}([\mu_i, t_i]^T) \quad \right). \tag{6}$$

Theorem 2. *The neural network in Equation 6 has Lipschitz constant $\frac{1}{C}$ w.r.t. one vector $[\mu_i, t_i]^T$.*

Proof: The mean function multiplies each vector, $g_\theta^{(1)}([\mu_i, t_i]^T)$, from a single message μ_i with $\frac{1}{C}$. The Lipschitz constant of all linear layers and activation functions in $g_\theta^{(1)}$ and $g_\theta^{(2)}$ does not exceed 1. Therefore, by Equation 5, the product of Lipschitz constants does not exceed $\frac{1}{C}$ □

Algorithm 1 denotes the full algorithm for use in contact tracing. With the sensitivity of the statistical prediction and neural augmentation together, we add noise according to the Gaussian mechanism to ensure (ε, δ)-DP. The crucial differences with the DPFN-sensitivity method are indicated in blue.

Algorithm 1 DNA: differential private neural augmentation

```
Require: Dataset $D = \{(\mu_i, t_i)\}_{i=1}^{C_T}$, constants $p_1, \gamma_u \in (0, 1)$, DP $(\varepsilon, \delta)$
Ensure: output $\phi$ is differentially private with budget $(\varepsilon, \delta)$
  $\mu_i \leftarrow \min(\mu_i, \gamma_u)$                                          ▷ Clip message value
  $\bar{\phi} \leftarrow F(\{(\mu_i, t_i)\}_{i=1}^{C_T}) + p_1 \times G_\theta(\{(\mu_i, t_i)\}_{i=1}^{C_T}) + \mathcal{N}(0, \frac{2}{\varepsilon^2}(\gamma_u p_1(1 + \frac{1}{C_T}))^2 \log(\frac{5}{4\delta}))$
  $\phi \leftarrow \min(\gamma_u, \max(0, \bar{\phi}))$                          ▷ Clip output by public knowledge
```

Lemma 1. *Algorithm 1 satisfies (ε, δ)-DP*

Proof: Function $G_\theta(\cdot)$ has Lipschitz constant with respect to an individual message $\frac{1}{C}$ per Theorem 2. When the message values are clipped and the timesteps t_i are fixed by assumption, $G_\theta(\cdot)$ has sensitivity $\frac{\gamma_u}{C}$ w.r.t. each message value μ_i. Combining this sensitivity with the sensitivity of FN at $p_1\gamma_u$, c.f. Theorem 1, the sensitivity of the composed neural augmentation is $\gamma_u p_1(1 + \frac{1}{C})$. We add noise according to the Gaussian mechanism with this sensitivity, ensuring (ε, δ)-DP. □

Learning under Lipschitz constraints To learn a Lipschitz-constrained neural network with stochastic gradient descent, we use the Power Iteration method to approximate the spectral norm of linear layers at each step during training (Miyato et al., 2018). The Power Iteration method, however, only approximates the spectral norm. Therefore, after training, we calculate the singular values exactly and project the weights accordingly. In practice, we find that the spectral norms after training are close to 1, and the projection step has negligible influence on the final performance.

4.1 Experimental details

The functions $g_{\theta_1}^{(1)}(\cdot)$ and $g_{\theta_2}^{(2)}(\cdot)$ are each a multilayer perceptron of $M = 8$ layers of width $w = 64$.

Data for training with the neural augmentation module is obtained from running the simulator three times for 100 time steps, once each time for the training, validation, and test set. Due to a large imbalance, negative samples are subsampled at random to match the number of positive samples. In the simulation, users get tested according to the estimated COVIDSCORE and self isolate upon a positive test. In this way, better predictions potentially lead to a lower overall virus spread. Further details are described in Section A.4. In the experimental results, we report the mean and 90% confidence interval of the mean after ten random restarts. The randomness arises from variability in contact patterns in the simulator, stochastic disease dynamics, and the additive noise required for DP.

5 Experimental results

We evaluate the four methods on a simulation of 10.000 agents for 100 days and report the peak infection rate (PIR), which is the largest fraction of simultaneously infected individuals. A high

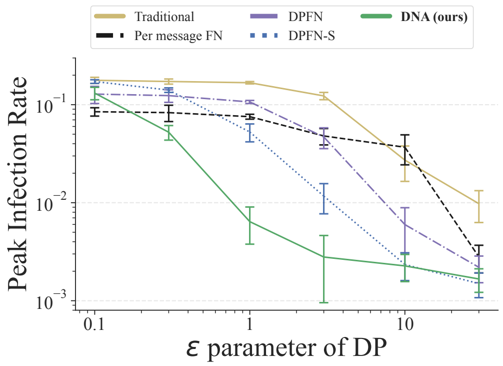

Figure 2: The trade-off between peak infection rate (y-axis) and the ε DP parameter (x-axis). At the crucial setting of $\varepsilon = 1$, our method, DNA, achieves a significantly lower peak infection rate.

	DPFN-S (‰)	**DNA** (‰)
Follow protocol		
100%	$52.7_{\pm 10.9}$	$6.4_{\pm 2.6}$
80%	$60.4_{\pm 9.6}$	$6.4_{\pm 2.2}$
50%	$100.1_{\pm 4.4}$	$27.2_{\pm 8.6}$
Noisy tests		
FPR 1%, FNR .1%	$52.7_{\pm 10.9}$	$6.4_{\pm 2.6}$
FPR 10%, FNR 1%	$81.3_{\pm 2.6}$	$19.5_{\pm 2.5}$
FPR 25%, FNR 3%	$130.4_{\pm 1.5}$	$81.3_{\pm 1.8}$

Table 1: Two ablation experiments to test the robustness. When up to 50% of the agents don't follow the protocol, or when the tests become more noisy, the DNA method still achieves lower PIR than the method without neural augmentation. Numbers are in one-per-thousand (‰).

number corresponds to a large pandemic, which strains the healthcare system and society with all its consequences. The parameters of the neural augmentation module are learned on a dataset based on the simulator, where we measure the Area Under the Receiver-Operator Curve (AUC) as a performance metric. On the test set, predictions from FN achieve an AUC of 77.0, while the combination with the neural augmentation module achieves 83.1 AUC. This improvement of more than six points in AUC already shows the benefits of neural augmentation on the prediction task.

Figure 2 shows the results of deploying the methods on the OpenABM-Covid19 simulator. We compare neural augmentation against traditional contact tracing, which counts only the number of positive contacts (Traditional, Baker et al. (2021)), against DPFN with privacy analysis per message (level 1, per-message FN), against privacy analysis per day (level 2, DPFN), and against privacy analysis by sensitivity (level 3, DPFN-S). The results show a trade-off between infection rates and privacy. For $\varepsilon = 0.1$, which is considered a very strict DP, all methods have a large PIR. On the other side, for $\varepsilon = 30$, which is not considered private, most methods have a low PIR. Expert studies identify $\varepsilon = 1$ as a target for DP (Hsu et al., 2014; Wood et al., 2018) and at this $\varepsilon = 1.0$-DP, the experimental results show that our DNA method has significantly lower PIR than other methods.

We run two ablation experiments to test the robustness of our method. First, we simulate that agents don't adhere to the protocol of voluntarily isolating after a positive test, e.g. they continue to interact with other agents. Up to 50 % non-adherence, the DNA method gets significantly lower PIR. Secondly, we run the simulation with higher false positive and false negative rates. The results in Table 1 show that even with tests as noisy as 25% FPR and 3% FNR, DNA achieves significantly lower PIR.

6 Discussion and conclusion

We propose a novel algorithm for statistical contact tracing using neural augmentation, which can learn patterns from data that are not captured by a statistical model. The algorithm is decentralized and maintains differential privacy against a recently identified attack scenario on contact tracing.

Further research is needed in two directions. Our algorithm quantifies the DP per message and repeated contacts are treated as individual messages, but a group of attackers could yield more information by repeatedly contacting the same individual. More research is needed on privacy composition in iterative decentralized inference methods. Secondly, in the classification setting, DP is known to exacerbate biases with respect to minority groups (Farrand et al., 2020), and although decentralized contact tracing is a different context, this effect should be further investigated.

The early consequences of a pandemic like COVID19 can be mitigated with contact tracing. We improve the predictions of a differentially private algorithm, and experiments show that our method significantly reduces the peak infection rate, especially at the level of $\varepsilon = 1$ per message. This is a crucial step for differentially private and decentralized contact tracing in case a new pandemic arises.

REFERENCES

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on Computer and Communications Security*, 2016.

Roy M Anderson and Robert M May. *Infectious diseases of humans: dynamics and control*. Oxford University Press, 1992.

Antoine Baker, Indaco Biazzo, Alfredo Braunstein, Giovanni Catania, Luca Dall'Asta, Alessandro Ingrosso, Florent Krzakala, Fabio Mazza, Marc Mézard, Anna Paola Muntoni, et al. Epidemic mitigation by statistical inference from contact tracing data. *Proceedings of the National Academy of Sciences*, 2021.

Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. *Advances in neural information processing systems (NeurIPS)*, 2017.

Donald Beaver, Silvio Micali, and Phillip Rogaway. The round complexity of secure protocols. In *ACM symposium on Theory of computing*, 1990.

Aner Ben-Efraim, Yehuda Lindell, and Eran Omri. Optimizing semi-honest secure multiparty computation for the internet. In *ACM SIGSAC Conference on Computer and Communications Security*, 2016.

Louis Béthune, Thomas Masséna, Thibaut Boissin, Yannick Prudent, Corentin Friedrich, Franck Mamalet, Aurelien Bellet, Mathieu Serrurier, and David Vigouroux. Dp-sgd without clipping: The lipschitz neural network way. *International Conference on Learning Representations (ICLR)*, 2023.

Alberto Blanco-Justicia, David Sánchez, Josep Domingo-Ferrer, and Krishnamurty Muralidhar. A critical review on the use (and misuse) of differential privacy in machine learning. *ACM Computing Surveys*, 2022.

Matt Boden, Lindsey Zimmerman, Kathryn J Azevedo, Josef I Ruzek, Sasha Gala, Hoda S Abdel Magid, Nichole Cohen, Robyn Walser, Naina D Mahtani, Katherine J Hoggatt, et al. Addressing the mental health impact of covid-19 through population health. *Clinical Psychology Review*, 2021.

Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 2011.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 2014.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference*. Springer, 2006.

Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*. IEEE, 2010.

ECDC. Considerations on the use of self-tests for covid-19 in the eu/eea. *ECDC technical report, 17 March 2021*, 2021.

Farzan Farnia, Jesse M. Zhang, and David Tse. Generalizable adversarial training via spectral normalization. In *International Conference on Learning Representations (ICLR)*, 2019.

Tom Farrand, Fatemehsadat Mireshghallah, Sahib Singh, and Andrew Trask. Neither private nor fair: Impact of data imbalance on utility and fairness in differential privacy. In *Proceedings of the 2020 workshop on privacy-preserving machine learning in practice*, 2020.

Golden Gao, Raynell Lang, Robert J Oxoby, Mehdi Mourali, Hasan Sheikh, Madison M Fullerton, Theresa Tang, Braden J Manns, Deborah A Marshall, Jia Hu, et al. Drivers of downloading and reasons for not downloading covid-19 contact tracing and exposure notification apps: A national cross-sectional survey. *PLOS one*, 2022.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems (NeurIPS)*, 2014.

Slawomir Goryczka and Li Xiong. A comprehensive comparison of multiparty secure additions with differential privacy. *IEEE transactions on Dependable and Secure Computing*, 2015.

Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2010.

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

Robert Hinch, William J. M. Probert, Anel Nurtay, Michelle Kendall, Chris Wymant, Matthew Hall, Katrina A. Lythgoe, Ana Bulas Cruz, Lele Zhao, Andrea Stewart, Luca Ferretti, Daniel Montero, James Warren, Nicole Mather, Matthew Abueg, Neo Wu, Olivier Legat, Katie Bentley, Thomas Mead, Kelvin Van-Vuuren, Dylan Feldner-Busztin, Tommaso Ristori, Anthony Finkelstein, David G. Bonsall, Lucie Abeler-Dörner, and Christophe Fraser. Openabm-covid19 - an agent-based model for non-pharmaceutical interventions against COVID-19 including contact tracing. *PLoS Computational Biology*, 2021.

Justin Hsu, Marco Gaboardi, Andreas Haeberlen, Sanjeev Khanna, Arjun Narayan, Benjamin C Pierce, and Aaron Roth. Differential privacy: An economic method for choosing epsilon. In *2014 IEEE 27th Computer Security Foundations Symposium*. IEEE, 2014.

Kevin Jenniskens, Martin CJ Bootsma, Johanna AAG Damen, Michiel S Oerbekke, Robin WM Vernooij, René Spijker, Karel GM Moons, Mirjam EE Kretzschmar, and Lotty Hooft. Effectiveness of contact tracing apps for sars-cov-2: a rapid systematic review. *BMJ open (British Medical Journal)*, 2021.

Madhav Jha and Sofya Raskhodnikova. Testing and reconstruction of lipschitz functions with applications to data privacy. *SIAM Journal on Computing*, 2013.

Kerina Jones, Rachel Thompson, et al. To use or not to use a covid-19 contact tracing app: Mixed methods survey in wales. *JMIR mHealth and uHealth*, 2021.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 2021.

Alan D Kaye, Chikezie N Okeagu, Alex D Pham, Rayce A Silva, Joshua J Hurley, Brett L Arron, Noeen Sarfraz, Hong N Lee, Ghali E Ghali, Jack W Gamble, et al. Economic impact of covid-19 pandemic on healthcare facilities and systems: International perspectives. *Best Practice and Research Clinical Anaesthesiology*, 2021.

William Ogilvy Kermack and Anderson McKendrick. A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society of London*, 1927.

Doyoung Kim, Hyangsuk Min, Youngeun Nam, Hwanjun Song, Susik Yoon, Minseok Kim, and Jae-Gil Lee. Covid-eenet: Predicting fine-grained impact of COVID-19 on local economies. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2022.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Kai Lønning, Patrick Putzky, Jan-Jakob Sonke, Liesbeth Reneman, Matthan WA Caan, and Max Welling. Recurrent inference machines for reconstructing heterogeneous mri data. *Medical image analysis*, 2019.

Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *International Conference on Learning Representations (ICLR)*, 2017.

Kentaro Minami, HItomi Arai, Issei Sato, and Hiroshi Nakagawa. Differential privacy without sensitivity. *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2018.

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2010.

Natalia Ponomareva, Hussein Hazimeh, Alex Kurakin, Zheng Xu, Carson Denison, H Brendan McMahan, Sergei Vassilvitskii, Steve Chien, and Abhradeep Guha Thakurta. How to dp-fy ml: A practical guide to machine learning with differential privacy. *Journal of Artificial Intelligence Research*, 2023.

Rob Romijnders, Yuki M. Asano, Christos Louizos, and Max Welling. No time to waste: practical statistical contact tracing with few low-bit messages. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2023.

Rob Romijnders, Christos Louizos, Yuki M. Asano, and Max Welling. Protect your score: Contact tracing with differential privacy guarantees. *Association for the Advancement of Artificial Intelligence (AAAI)*, 2024.

Michal Rosen-Zvi, Michael I. Jordan, and Alan L. Yuille. The DLR hierarchy of approximate inference. *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2005.

Victor Garcia Satorras and Max Welling. Neural enhanced belief propagation on factor graphs. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021.

Marten van Dijk and Phuong Ha Nguyen. Considerations on the theory of training models with differential privacy. *arXiv*, 2023.

Nina Vindegaard and Michael Eriksen Benros. Covid-19 pandemic and mental health consequences: Systematic review of the current evidence. *Brain, Behavior, and Immunity*, 2020.

Michel Walrave, Cato Waeterloos, and Koen Ponnet. Reasons for nonuse, discontinuation of use, and acceptance of additional functionalities of a covid-19 contact tracing app: cross-sectional survey study. *JMIR Public Health and Surveillance*, 2022.

Alexandra Wood, Micah Altman, Aaron Bembenek, Mark Bun, Marco Gaboardi, James Honaker, Kobbi Nissim, David R O'Brien, Thomas Steinke, and Salil Vadhan. Differential privacy: A primer for a non-technical audience. *Vanderbilt Journal of Entertainment and Technology Law*, 2018.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems (NeurIPS)*, 30, 2017.

A Appendix

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. Qualcomm AI research is an initiative of Qualcomm Technologies, Inc. and/or its subsidiaries. Correspondence may go to r.romijnders@uva.nl.

A.1 Notation

We repeat the most important notation in this paper.

- $\phi_{u,t}$ is the COVIDSCORE of user u on day t.
- μ_i is a message sent in decentralized inference. This is a function of the COVIDSCORE of the sender, $\phi_{c,t}$.
- $z_{u,t}$ is the random variable of a user u on day t and takes on the values $\{S, E, I, R\}$ for the Susceptible, Exposed, Infected, Recovered states.
- $z_{u,1:T} = \{z_{u,1}, z_{u,2}, \cdots, z_{u,T}\}$ corresponds to the set of random variables for the time range 1 up to and including T.
- C indicates the number of contacts; C_t is the number of contacts on day t.
- $p_0, p_1, g, h \in (0, 1)$ are scalar model parameters. Its values are taken from previous studies such as (Herbrich et al., 2020; Romijnders et al., 2023).
- $b(z)$: the functions b generally indicate beliefs (Rosen-Zvi et al., 2005), which could be either $b_u(z_{u,t})$ for a particular user or $b_{N(u)}(z_{N(u)})$ for a set of users.
- $N(u, t)$ are the neighbors of user u at day t. We write $N(u)$ and $N(t)$ for briefness when u or t is clear from context.
- $\varepsilon > 0,\ \delta \in [0, 1)$ are the parameters for differential privacy.
- γ_u is the clipping bound for messages between decentralized agents, e.g. $\mu_i \in [0, \gamma_u]$.

A.2 Background

Details on the statistical model: The random variables are written as $z_{u,t}$ for user u, at time step t. For a particular user, the variables $z_{u,t}, z_{u,t+1}, \cdots$ form a Markov chain. The Markov chain is described by a conditional distribution between one timestep and the next:

$$P(z_{u,t+1}|z_{u,t}, z_{N(u,t)}) = \begin{cases} \psi(u, t, z_{N(u,t)}) & S \to S \\ 1 - \psi(u, t, z_{N(u,t)}) & S \to E \\ 1 - g & E \to E \\ g & E \to I \\ 1 - h & I \to I \\ h & I \to R \\ 1 & R \to R \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

The prior for the first timestep for every user is:

$$P(z_{u,1}) = \begin{cases} 1 - p_0 & z_{u,1} = S \\ p_0 & z_{u,1} = E \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

The function $\psi(\cdot)$ is the statistical model for virus transmission over a user contact. This dynamic is described with a noisy-OR model, c.f. Equation 1 and repeated here:

$$\psi(u, t, z_{N(u,t)}) = (1 - p_0)(1 - p_1)^{|\{z_c \, \in \, z_{N(u,t)} \colon \, z_c = I\}|}. \tag{9}$$

Further introduced variables are g, h, p_0, and p_1, which are set to value from previous literature (Herbrich et al., 2020; Romijnders et al., 2023). A user can have contact with multiple users on a particular day. As such, $z_{N(u,t)}$ denotes the set of random variables of all contacts of user u at time step t.

Particular users can also take a test for COVID19. These tests can be assigned randomly, or a potential contact tracing algorithm could predict a COVIDSCORE per user and prompt users with a high score to take a test. The test may have a false positive or false negative result with respect to the underlying state. We denote the False Positive rate (FPR) by β and denote the False Negative rate (FNR) by α. The conditional observation model then follows:

$$P\left(o_{u,t} | z_{u,t}\right) = \begin{cases} \alpha & \text{if } z_{u,t} = I \wedge o_{u,t} = 0 \\ 1 - \alpha & \text{if } z_{u,t} = I \wedge o_{u,t} = 1 \\ 1 - \beta & \text{if } z_{u,t} \in \{S, E, R\} \wedge o_{u,t} = 0 \\ \beta & \text{if } z_{u,t} \in \{S, E, R\} \wedge o_{u,t} = 1 \end{cases}. \tag{10}$$

Lemma 2. *FN update of noisy-or model*

This lemma repeats Equation 8 in Romijnders et al. (2023).

The expectation of the noisy-OR model under FN for user v after C contacts happened at timestep τ. The random variables of these contacts are denoted $z_{c,\tau}$. This has the corresponding FN belief $b_c(z_{c,\tau})$, with message parameter $\mu_{c,\tau}$. When clear from context, we write the message parameter as μ_i for contact $i = c$. A central assumption in FN is that the belief over the set of neighbor nodes, $B_{N(u)}$, follows a factor distribution $B_{N(u)} = \prod_{c=1}^{C} b_c(z_{c,\tau})$.

$$\begin{aligned} &\mathbb{E}_{B_{N(u)}(z_{N(u)})}\Big[p(z_{v,\tau+1} = S | z_{v,\tau} = S, z_{N(v,\tau)} = \{z_{c,\tau}\}_{c=1}^{C})\Big] \\ &\qquad = E_{B_{N(u)}(z_{N(u)})}\left[(1 - p_0)\prod_{c=1}^{C}(1 - p_1)^{\mathbf{1}[z_{c,\tau}]}\right] \\ &\qquad = (1 - p_0)\prod_{c=1}^{C} E_{b_c(z_{c,\tau})}\left[(1 - p_1)^{\mathbf{1}[z_{c,\tau}]}\right] \\ &\qquad = (1 - p_0)\prod_{c=1}^{C}\left[1 - p_1\mu_{c,\tau}\right] \end{aligned} \tag{11}$$

Notation $\mathbf{1}[\cdot]$ indicates the Iverson bracket for having state I, which evaluates to 1 when the argument has state I and 0 otherwise.

A.3 Global sensitivity proof

This section provides the proof for Theorem 1. We aim to bound the sensitivity of the FN algorithm on two adjacent datasets, as defined in Equation 3. The main establishment of this proof is to address the sensitivity with respect to the value of one message in the context of arbitrarily many other statistical messages on that day and other days.

Each dataset has all contacts' messages in a time window of the past T days. Figure 3 provides a schematic illustration of the proof. Without loss of generality, we consider two adjacent datasets where the value of a message, $\mu_{c,\tau}$, of contact c on day τ from dataset D to dataset D' is changed. On that day, assume there are C contacts in total, $c = 1, c = 2, \cdots, c = C$. Each contact sends a message $(1 - p_1\mu_{1,\tau}), (1 - p_1\mu_{2,\tau}), \cdots, (1 - p_1\mu_{C,\tau})$. Here $p_1 \in (0, 1)$ is a model parameter. The definition of these messages follow from (Romijnders et al., 2023), and are repeated in Lemma 2.

We introduce two shorthand notations for a set of random variables. We write the set of random variables in a sequence of days $z_{1:T} = \{z_1, z_2, \cdots, z_T\}$. We also use $z_{N(u,1:T)}$ to denote the neighboring nodes, contacts, of user u in timesteps 1 up to and including T, where a neighbor could be any user v not equal to u. We write $Z_{N(u)}$ when the timesteps are evident from the context.

The general update rule for FN is:

$$\begin{aligned}
F(D) &= b(z_T = I) \\
&= \sum_{z_{1:T-1}} \mathbb{E}_{b_{N(u)}}[p(z_T = I, z_{1:T-1}|z_{N(u,1:T)})] && (12) \\
&= \sum_{z_{1:T-1}} \mathbb{E}[p(z_T = I, z_{\tau+1:T-1}|z_{1:\tau}, z_{N(u,\tau+1:T)})]\mathbb{E}[p(z_{1:\tau}|z_{N(u,1:\tau)})] && (13) \\
&= \sum_{z_{1:T-1}} \mathbb{E}[p(z_T = I, z_{\tau+1:T-1}|z_{\tau}, z_{N(u,\tau+1:T)})]\mathbb{E}[p(z_{1:\tau}|z_{N(u,1:\tau)})] && (14)
\end{aligned}$$

The equalities in Equations 13 and 14 follow from the Markov property of the model, c.f. Equation 7. The Markov property implies the conditional independence:

$$\begin{aligned}
z_{\tau+1:T} \perp z_{1:\tau-1} \mid z_\tau && (15) \\
z_{\tau+1:T} \perp z_{N(u,1:\tau)} \mid z_\tau. && (16)
\end{aligned}$$

We split summation in Equation 14 in three parts: timesteps before τ, the transition at timestep τ, and all timesteps after τ. In the following, all expectations are taken with respect to the factored FN beliefs, so we use shorthand $\mathbb{E}[\cdot]$ to indicate $\mathbb{E}_{b_{N(u)}}[\cdot]$, c.f. Lemma 2.

$$\begin{aligned}
F(D) &= \sum_{z_{\tau+1:T-1}} \sum_{z_\tau} \mathbb{E}[p(z_T = I, z_{\tau+1:T-1}|z_\tau, z_{N(u,\tau+1:T)})] \sum_{z_{1:\tau-1}} \mathbb{E}[p(z_{1:\tau}|z_{N(u,1:\tau)})] && (17) \\
&= \sum_{z_{\tau+1:T-1}} \sum_{z_\tau} \mathbb{E}[p(z_T = I, z_{\tau+2:T-1}|z_{\tau+1}, z_{N(u,\tau+2:T)})p(z_{\tau+1}|z_\tau, z_{N(u,\tau+1)})] \\
&\quad \cdot \sum_{z_{1:\tau-1}} \mathbb{E}[p(z_{1:\tau}|z_{N(u,1:\tau)})]. && (18)
\end{aligned}$$

The conditional distribution for $z_{\tau+1}$ in Equation 18 is highlighted in blue for clarity, as we will consider two explicit cases for z_τ in this conditional distribution for the following analysis. The random variable z_τ can be either in state S or not in state S, which are states $\{E, I, R\}$. From state S the only non-zero conditional probabilities, under Equation 7, are to state S and state E for $z_{\tau+1}$. For brevity of notation, we will write $z_{N(t_1:t_2)}$ for $z_{N(u,t_1:t_2)}$ when the user is clear from context.

$$\begin{aligned}
&F(D) = \\
&\Bigg(\sum_{z_{\tau+2:T-1}} (\mathbb{E}[p(z_T = I, z_{\tau+2:T-1}|z_{\tau+1} = S, z_{N(\tau+2:T)})]\mathbb{E}[p(z_{\tau+1} = S|z_\tau = S, z_{N(\tau+1)})] \\
&\qquad + \mathbb{E}[p(z_T = I, z_{\tau+2:T-1}|z_{\tau+1} = E, z_{N(\tau+2:T)})]\mathbb{E}[p(z_{\tau+1} = E|z_\tau = S, z_{N(\tau+1)})]) \\
&\qquad\qquad \cdot \sum_{z_{1:\tau-1}} \mathbb{E}[p(z_\tau = S, z_{1:\tau-1}|z_{N(1:\tau)})] \Bigg) \\
&\qquad + \Bigg(\sum_{z_{\tau+1:T-1}} \sum_{z_\tau \in \{E,I,R\}} \mathbb{E}[p(z_T = I, z_{\tau+1:T-1}|z_\tau \neq S, z_{N(\tau+2:T)})] \sum_{z_{1:\tau-1}} \mathbb{E}[p(z_\tau, z_{1:\tau-1}|z_{N(1:\tau)})] \Bigg). \\
& && (19)
\end{aligned}$$

The outer sum in Equation 19 starts at $\tau + 2$ because for timestep $\tau + 1$ in the first factor, the states S and E are explicitly written. When the contact occurs on day τ and we name this contact user v, then

$z_{v,\tau} \in z_{N(u,\tau+1)}$. In Equation 19, conditioned on $z_\tau \neq S$, the random variables $z_T = I, z_{\tau+1:T-1}$ are conditionally independent from $z_{N(u,\tau+1)}$. Therefore, we define two factors that are constant w.r.t. the random variables of the contacts on day τ:

$$K_0 = \sum_{z_{\tau+1:T-1}} \sum_{z_\tau \in \{E,I,R\}} \mathbb{E}[p(z_T = I, z_{\tau+1:T-1} | z_\tau, z_{N(\tau+2:T)})] \sum_{z_{1:\tau-1}} \mathbb{E}[p(z_\tau, z_{1:\tau-1} | z_{N(1:\tau)})] \tag{20}$$

$$K_1 = \sum_{z_{1:\tau-1}} \mathbb{E}[p(z_\tau = S, z_{1:\tau-1} | z_{N(1:\tau)})]. \tag{21}$$

Rewrite Equation 19 and fill in the conditional distribution as defined in Equation 11. For ease of notation, all messages on day τ will be written as μ_i instead of $\mu_{i,\tau}$.

$$\begin{aligned}
F(D) = K_0 + K_1 \sum_{z_{\tau+2:T-1}} \Big(&\mathbb{E}[p(z_T = I, z_{\tau+2:T-1} | z_{\tau+1} = S, z_{N(\tau+2:T)})] \mathbb{E}[p(z_{\tau+1} = S | z_\tau = S, z_{N(\tau+1)})] \\
&+ \mathbb{E}[p(z_T = I, z_{\tau+2:T-1} | z_{\tau+1} = E, z_{N(\tau+2:T)})] \mathbb{E}[p(z_{\tau+1} = E | z_\tau = S, z_{N(\tau+1)})] \Big) \\
= K_0 + K_1 \Big(&\mathbb{E}[p(z_{\tau+1} = S | z_\tau = S, z_{N(\tau+1)})] \sum_{z_{\tau+2:T-1}} \mathbb{E}[p(z_T = I, z_{\tau+2:T-1} | z_{\tau+1} = S, z_{N(\tau+2:T)})] \\
&+ \mathbb{E}[p(z_{\tau+1} = E | z_\tau = S, z_{N(\tau+1)})] \sum_{z_{\tau+2:T-1}} \mathbb{E}[p(z_T = I, z_{\tau+2:T-1} | z_{\tau+1} = E, z_{N(\tau+2:T)})] \Big) \\
= K_0 + K_1 \Big(&(1 - p_0)(1 - p_1\mu_1)(1 - p_1\mu_2) \cdots (1 - p_1\mu_C) \\
&\cdot \sum_{z_{\tau+2:T-1}} \mathbb{E}[p(z_T = I, z_{\tau+2:T-1} | z_{\tau+1} = S, z_{N(\tau+2:T)})] \\
&+ (1 - (1 - p_0)(1 - p_1\mu_1)(1 - p_1\mu_2) \cdots (1 - p_1\mu_C)) \\
&\cdot \sum_{z_{\tau+2:T-1}} \mathbb{E}[p(z_T = I, z_{\tau+2:T-1} | z_{\tau+1} = E, z_{N(\tau+2:T)})] \Big)
\end{aligned} \tag{22}$$

We further introduce two constants w.r.t. the message value μ_i:

$$K_S = \sum_{z_{\tau+2:T-1}} \mathbb{E}[p(z_T = I, z_{\tau+2:T-1} | z_{\tau+1} = S, z_{N(\tau+2:T)})] \tag{23}$$

$$K_E = \sum_{z_{\tau+2:T-1}} \mathbb{E}[p(z_T = I, z_{\tau+2:T-1} | z_{\tau+1} = E, z_{N(\tau+2:T)})] \tag{24}$$

Lemma 3. *Constants K_0, K_1, K_S, and K_E are in $[0, 1)$.*

The conditional probability distributions in Equations 7 and 1 take value in $[0, 1)$ for any variable realization because the model parameters are chosen such that $0 < p_0, p_1, g, h < 1$ and the joint probability distributions are on a discrete domain. This holds for all conditional distributions in the definitions of K_0, K_1, K_S, and K_E.

The expectation operator, $\mathbb{E}[\cdot]$, is a convex combination, and a convex combination of numbers in $[0, 1)$, is itself in $[0, 1)$. The constants K_0, K_1, K_S, and K_E in Equations 20, 21, 23, 24 can all be rewritten to an expectation of a joint probability distribution and are, therefore, in $[0, 1)$.

Now, we can rewrite Equation 22, replacing all factors that are constant with respect to μ_1.

$$\begin{aligned}
F(D) = \big(&K_S(1 - p_0)(1 - p_1\mu_1)(1 - p_1\mu_2) \cdots (1 - p_1\mu_C) \\
&+ K_E(1 - (1 - p_0)(1 - p_1\mu_1)(1 - p_1\mu_2) \cdots (1 - p_1\mu_C)) \big) K_1 + K_0.
\end{aligned} \tag{25}$$

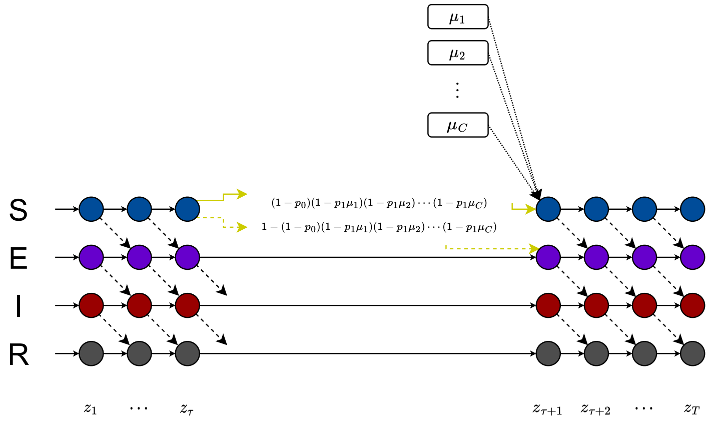
μ_1
μ_2
$\vdots$
μ_C
S
E
I
R
$(1-p_0)(1-p_1\mu_1)(1-p_1\mu_2)\cdots(1-p_1\mu_C)$
$1-(1-p_0)(1-p_1\mu_1)(1-p_1\mu_2)\cdots(1-p_1\mu_C)$
z_1
$\cdots$
z_τ
$z_{\tau+1}$
$z_{\tau+2}$
$\cdots$
z_T

Figure 3: This diagram illustrates the proof setup in Section A.3. The value of μ_1 on day τ occurs in the conditional probability distribution $p(z_{\tau+1}|z_\tau = S, z_{N(u)})$. Filled lines indicate a transition to an equal state, and dashed lines indicate a transition to the next state in the $S \rightarrow E \rightarrow I \rightarrow R$ order.

Finally, we obtain the sensitivity from taking the difference between the two adjacent datasets D, D'. The FN inference update equation is invariant to a permutation of the messages in the same day. Therefore, as mentioned, without loss of generality, we assume that the change from D to D' happens in contact 1, which changes its message value from μ_1 to μ_1'.

$$\begin{aligned}
\|F(D) - F(D')\| = \|\Big(\big(&K_S(1-p_0)(1-p_1\mu_1)(1-p_1\mu_2)\cdots(1-p_1\mu_C) \\
&+ K_E(1-(1-p_0)(1-p_1\mu_1)(1-p_1\mu_2)\cdots(1-p_1\mu_C))\big)K_1 + K_0\Big) - \\
&\Big(\big(K_S(1-p_0)(1-p_1\mu_1')(1-p_1\mu_2)\cdots(1-p_1\mu_C) \\
&+ K_E(1-(1-p_0)(1-p_1\mu_1')(1-p_1\mu_2)\cdots(1-p_1\mu_C))\big)K_1 + K_0\Big)\| \qquad (26)
\end{aligned}$$

$$\begin{aligned}
\leq \|\Big(&K_S(1-p_0)(1-p_1\mu_1)(1-p_1\mu_2)\cdots(1-p_1\mu_C) \\
&+ K_E(1-(1-p_0)(1-p_1\mu_1)(1-p_1\mu_2)\cdots(1-p_1\mu_C))\Big) - \\
&\Big(K_S(1-p_0)(1-p_1\mu_1')(1-p_1\mu_2)\cdots(1-p_1\mu_C) \\
&+ K_E(1-(1-p_0)(1-p_1\mu_1')(1-p_1\mu_2)\cdots(1-p_1\mu_C))\Big)\| \qquad (27)
\end{aligned}$$

$$\leq \|(1-p_1\mu_1) - (1-p_1\mu_1')\| \qquad (28)$$

$$= p_1\|\mu_1 - \mu_1'\| \qquad (29)$$

Arriving at Equation 27, we use Lemma 3 that $0 \leq K_0 < 1$ and $0 \leq K_1 < 1$. Likewise, arriving at Equation 28, we use that $0 \leq K_S < 1$ and $0 \leq K_E < 1$.

If we assume that every message μ_i is clipped to the range $[0, \gamma_u]$, then we prove the sensitivity.

$$\Delta = \max_{\mu_1, \mu_1' \in [0, \gamma_u]} \left\| F\big(\ \{(\mu_1, t_1)\} \cup D\ \big) - F\big(\ \{(\mu_1', t_1)\} \cup D\ \big) \right\| \leq p_1\gamma_u \qquad \forall\ D \qquad (30)$$

A.4 Experimental details

The different methods are compared on a simulator for COVID19. We need a simulator to evaluate the interaction between predictions from each method, how that affects which individuals get tested, and the resulting change in infection rates if infectious individuals go in isolation. All methods in this paper are decentralized to ensure the locality of data. Each day, the method assigns a COVIDSCORE for the local user. The top 8% of agents with the highest score receive a signal to get tested. An agent that tests positively isolates themselves for ten days, and no contacts occur during this period. We assume that a private algorithm exists to determine the agents with the highest score (Beaver et al., 1990; Ben-Efraim et al., 2016). The tests have a False Positive Rate of 1% and False Negative Rate of 0.1%. In Table 1, we report on two ablation experiments where users do not adhere to the isolation protocol and when the false positive and false negative rates increase.

The settings for the OpenABM-Covid19 simulator (Hinch et al., 2021) follow the parameter settings in previous work (Baker et al., 2021; Romijnders et al., 2024). The simulator has over 150 modifiable parameters and uses different network properties to model agents. The parameters are calibrated against population data from the UK, in terms of age distributions and household and occupation patterns. Due to computational limits, the simulator's experimental results are each with 10.000 agents. The noise rates in Table 1 also follow previous work (Romijnders et al., 2024). The largest values of 25% False Positive Rate and 3% False Negative Rate correspond to the maximum allowance for an approved COVID19 test by the European Centre for Disease Control (ECDC, 2021).

For training the neural augmentation module, we use a dataset that is extracted from the OpenABM-Covid19 simulator – on different random seeds than later testing. We run the simulator on 100.000 agents for 100 time steps. As the model will be used in a decentralized setting, we store, per agent per timestep, the local messages, observations, and the underlying simulator state. The loss function for training is a Mean Squared Error, and the target label is one if the agent has the infected state, I, and zero if the agent has any other state, e.g., S, E, R. We measure the Area Under Receiver-Operator Curve (AUC) as a performance metric to make model decisions. The test set is obtained with the same protocol but a different random seed. On this test set, FN by itself achieves an AUC of 77.0, while the neural augmentation module achieves 83.1. This improvement of almost six points in AUC already shows the benefits of neural augmentation on the static dataset. Note that without neural augmentation, we found that the best Lipschitz-constrained network achieves only 80.3 AUC.

The parameter vector $\theta = [\theta_1; \theta_2]$ is learned with stochastic gradient descent, ADAM (Kingma & Ba, 2015). The model trains for 40 epochs, with weight decay 10^{-9} and learning rate 0.002, which decays to 0.0002 by a cosine learning rate decay schedule (Loshchilov & Hutter, 2017). The functions $g^{(1)}_{\theta_1}(\cdot)$ and $g^{(2)}_{\theta_2}(\cdot)$ are each a multilayer perceptron of $M = 8$ layers of width $w = 64$, as discussed in Section 4.1. The activation function is the Rectified Linear Unit (Nair & Hinton, 2010). During training, we find that using two or more power iteration steps per gradient descent step results in spectral norms close to 1. After training, we calculate the spectral norms exactly and project the weights accordingly.

All inference algorithms in this study use a finite time window of $T = 14$ days. This means that after 14 days, a message from a contact is deleted and has no influence on future predictions anymore. The function for FN, $F(\cdot)$, is also stateless. So, any user can withdraw their data at any time, and it will not be used the next day or thereafter.

Experiments in this paper use (ε, δ) differential privacy. We follow the convention and set δ smaller than one divided by the dataset size (Blanco-Justicia et al., 2022; Hsu et al., 2014; van Dijk & Nguyen, 2023). The simulators have an average of fifteen contacts daily, and the time window is fourteen days. Therefore, we set $\delta = \frac{1}{1000}$, which is well below the recommended $\frac{1}{14 \times 15}$.