

Supplementary Materials: "Epidemic inference through generative neural networks"

Indaco Biazzo^{1*}, Alfredo Braunstein^{1,2,3}, Luca Dall'Asta^{1,2,3}, and Fabio Mazza¹

¹ Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129, Torino, Italy,

² IIGM - Italian Institute for Genomic Medicine, c/o IRCSS, 10060 Candiolo, Torino, Italy

³ Collegio Carlo Alberto, P.za Arbarello 8, 10122, Torino, Italy

*To whom correspondence should be addressed; E-mail: indaco.biazzo@polito.it

1 Learning procedures and regularization

The learning procedure involves the minimization of the Kullback-Leibler divergence, eq.9 in the main text, between the posterior probability and the variational autoregressive neural networks. More precisely, the expression of the posterior can be factorized as a product over nodes (individuals) and times, $p(\underline{\mathbf{x}}|\mathcal{O}) \propto \prod_i \Psi_i((x_i, \underline{\mathbf{x}}_{\partial i})) = \prod_{i,t} \psi_i^t(x_i^t, \mathbf{x}_{\partial i}^{t-1})$ where

$$\psi_i^t(x_i^t, \mathbf{x}_{\partial i}^{t-1}) = \begin{cases} p_i^t(x_i^{(t)} | \mathbf{x}_{\partial i}^{t-1}) \prod_{(i_r, t_r)=(i,t)} p_r(O_r^{t_r} | x_i^t) & t > 0 \\ p_i(x_i^0) \prod_{(i_r, t_r)=(i,t)} p_r(O_r^{t_r} | x_i^t) & t = 0 \end{cases} \quad (1)$$

Since the factor ψ_i^t can be zero for some epidemic configuration $\underline{\mathbf{x}}$, for example when they are forbidden by dynamical constraints, the gradient of the KL divergence, Eq.12 in the main text, can diverge preventing the application of gradient descent algorithms. In these cases, the diverging term $\log \psi_i^t = -\infty$ can be replaced with a regularization term $\log \epsilon$ with $\epsilon \ll 1$. The presence of large negatives values in the derivatives (e.g. due to $\log \epsilon$ terms) reduces the ability of gradient descent algorithms to explore all configurations compatible with the constraints. To

overcome this issue, an annealing procedure is adopted, in which a fictitious inverse temperature β is introduced in the computation the gradient of the KL divergence,

$$\nabla_{\theta} D_{KL}^{\beta}(q|p) = \sum_{\underline{x}} q_{\theta} [-\beta \log p + \log q_{\theta}] \nabla_{\theta} \log q_{\theta}. \quad (2)$$

The minimization procedure starts with $\beta = 0$, where all the configurations are allowed with uniform probability, then the parameter β is slowly increased until it reaches $\beta = 1$, at which the original expression of the loss function is recovered.

2 Neural network architecture

In the proposed approach, each conditional probability function $p_i(\underline{x}_i|\underline{\mathbf{x}}_{<i})$ has to be approximated with a neural network $q_i^{\theta_i}(\underline{x}_i|\underline{\mathbf{x}}_{<i})$. For monotonic models such as the SIR epidemic model, the state-space representation of temporal trajectories as a sequence $\underline{x}_i \in \mathcal{X}^{T+1}$ of $T+1$ individual states is redundant, and it turns out to be more efficient to represent them by the time instant $t_i^I \in (0, T+1)$ at which individual i becomes infected and time instant $t_i^R \in (0, T+1)$ at which it recovers, the time $T+1$ corresponding to the case of no infection or recovery occurred for i in the interval $[0, T]$. More precisely, the case $(t_i^I = T+1, t_i^R = T+1)$ corresponds to individual i being susceptible at every time of the epidemic process, and $(t_i^I \neq T+1, t_i^R = T+1)$ corresponds to individual i being infected but not recovering in the time interval $[0, T]$.

This parametrization reduces the state space that is explored by our method, decreasing the number of incorrect time trajectories generated (for instances, time trajectory with transitions from state I to S).

In order to compute the probability of a trajectory of a single individual, we employ two neural networks, one for the time of infection, and one for the recovery time. Each of these networks is given as input a subset of the time trajectories of individuals with lower sorting index π_i (see next section), and the one of the recovery time is also given as input the infection

time of the same individual. The input of the networks are the infection and recover times of previous individuals, encoded in *one hot* scheme. Therefore, the (conditional) probability distributions represented by the networks are, for each individual i , $q_i^{\theta_{i,I}} \left(t_i^I \mid \{t_j^I, t_j^R\}_{\pi_j < \pi_i} \right)$ for the infection times, and $q_i^{\theta_{i,R}} \left(t_i^R \mid \{t_j^I, t_j^R\}_{\pi_j < \pi_i}, t_i^I \right)$ for the recovery times, where $\theta_{i,I}$ and $\theta_{i,R}$ are the weights of each network.

In our implementation, each neural network is a multi-layer perceptron (MLP) composed of three hidden layers plus one output layer; each layer is fully connected, and can be written as $\underline{L}_{k+1} = \sigma(\mathbf{W}_k \underline{L}_k + \underline{b}_k)$ where $\underline{L}_k$ is the input vector (output of layer k), $\mathbf{W}_k \in \theta$ is the matrix of the weights, $\underline{b}_k \in \theta$ is the bias vector, and σ is the activation function, which is *Relu* for the hidden internal layers and *Softmax* for the last layer. The width of each layer (the number of neurons) varies linearly from the input size of each network to the output size.

The size of the input and outputs of the neural networks used for each individual depends on the observations and the contact graph in the following way: a) the input size depends on the number of individuals with lower index that are considered (depending on the approximation made, see section 2.1) and b) the observations made on an individual restrict the phase space of the possible instants of infection (for example, if we observe individual i is in the infected state at time t_O , this means that his/her infection time $t_i^I \leq t_O$ and the recovery time $t_i^R > t_O$). This last condition affects the size of the output of each network, but also the input size of the networks for the individuals with higher index, whose network depends on it: for example, if individual i is observed in state S at the final time (T) with certainty, this implies that the infection and recovery times are fixed, $t_i^I = T + 1$ and $t_i^R = T + 1$, and the network will always give the same value for them. Thus, in this case, for all individuals j with $\pi_j > \pi_i$, the dependency on i will effectively disappear.

2.1 Graph approximations

We now discuss the approximation to the dependency of the conditional probabilities for the individuals' trajectories. The probability of a whole epidemic cascade can be written in the following autoregressive expression:

$$q^\theta(\underline{\mathbf{x}}) = \prod_i q_i^{\theta_i}(\underline{x}_i | \underline{\mathbf{x}}_{<i}), \quad (3)$$

The dependencies in the factors $q_i^{\theta_i}(\underline{x}_i | \underline{\mathbf{x}}_{<i})$ can be reduced, in the case of acyclic contact networks, to only those corresponding to the next-nearest neighbors with lower index (we relegate the derivation to section 3), but in the general case of contact networks with cycles it is just an approximation, which we employ for all contact networks analyzed in the present work.

In order to check this approximation, we have run several tests on the **patient zero** problem (see main text for the definition), applied to the proximity random contact graphs. For this purpose, we train the ANN either (a) considering the full dependencies on all nodes with lower permutation index (we call this version *full graph*), (b) consider only the dependency on the first and second neighbors (called *next nearest neighbors* approximation) with lower permutation index, which is the approximation used in all the rest of this work, (c) considering only the first neighbors (*nearest neighbors*) with lower permutation index, or (d) ignoring the dependency on the rest of the graph (called *mean field* approximation).

Figure 1 shows how the accuracy in finding the patient zero is influenced by the approximation chosen, while in figure 2 the reduction of the number of parameters in the network, with respect to the full graph case, is shown. We see that the *next nearest neighbors* approximation gives estimates which are on par with the full graph case, while employing less parameters, thus reducing the space and time needed for the training.

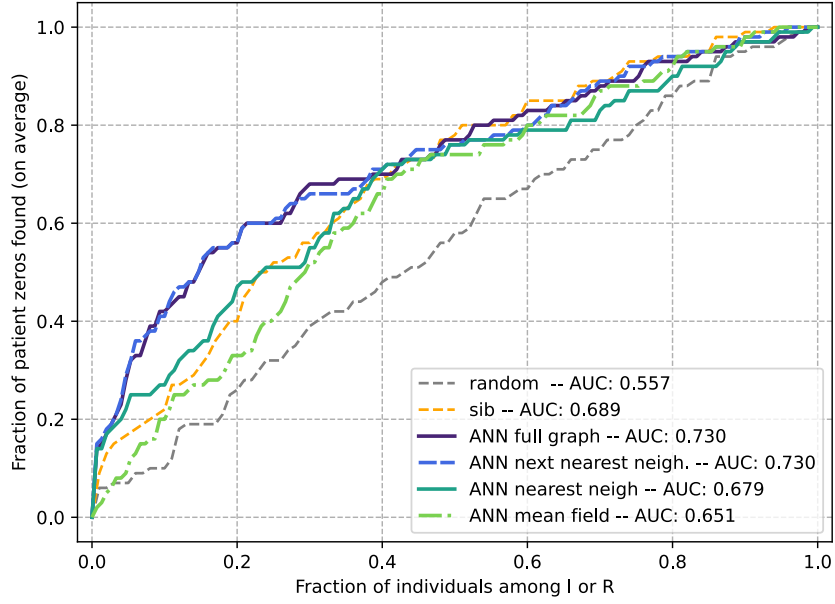


Figure 1: Accuracy in finding the patient zero with different approximations of the dependencies between individuals, compared with sib (Belief Propagation) method and random guessing of the source among the infected. The "full graph" case is when no approximation on the dependencies is made. The average is made over 100 epidemic cascade on contacts network generated from a **proximity** random graph (see main text) with $N = 100$ individuals. The epidemic parameters used for the generation are $\lambda = 0.03$ and $\mu = 0.02$, with $T = 15$ time instants. The legend also shows the Area under each curve (AUC).

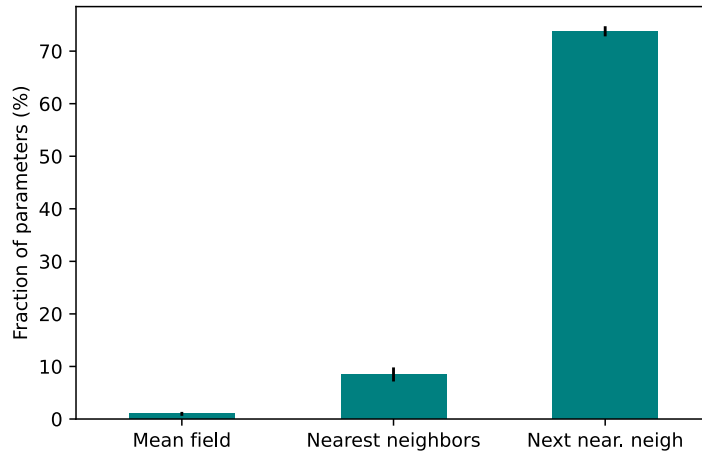


Figure 2: Percentage of parameters needed, on average, when considering the approximation of the dependencies between individuals, with respect to the case of no approximation as described in section 2.1. See figure 1 for the accuracy and generation details.

2.2 Training procedure

To train the whole network, we do 10 000 steps of the annealing procedure described in the main text (except for the **hospital** case where we do 20 000 steps), increasing β linearly from 0 to 1. At each step, we generate 10 000 epidemic cascades before updating the weights using the ADAM optimizer [1] with learning rate $lr = 0.001$.

In the case of the risk inference problem, we also include a prior probability in the model, with strength decreasing linearly with β . This prior is computed from the probabilities of each individual being sampled as S , I or R at the last time instant $t = T$, and it is designed in such a way that, when $\beta = 0$, the probabilities for the three states are equal.

The running time of the ANN algorithm on a single instance ranges from one hour to two days on a single GPU (Nvidia TITAN RTX), depending on the structure of the observations $\mathcal{O}$ and that of the underlying contact network. Our implementation using the PyTorch framework [2] is publicly available in the repository [3]. The results presented in the main text and in SI can be reproduced using the following repository [4].

3 Simplified Conditional probability on acyclic graph

In this section we show that in the case of epidemic spreading in a acyclic interaction networks, it is possible to restrict the dependence of previous nodes in the conditional probabilities Eq.7, to the second neighbors with lower index. Let us consider the configuration x of N variables, a probability distribution $p(\underline{x}) = \frac{1}{Z} \prod_a \psi_a(x_a)$ factorized over a set of factors $\{a\}$. We first demonstrate the following statements:

Lemma 1 (Markov blanket). *Let $p(x) = \frac{1}{Z} \prod_a \psi_a(x_a)$, and $I \cup J \cup K = \{1, \dots, N\}$ be disjoint and assume no factor depends on x_I and x_K simultaneously. Then $p(x_I, x_K | x_J) = p(x_I | x_J) p(x_K | x_J)$.*

Proof. Considering proportionality $\propto$ with respect to x_I only $p(x_I|x_J, x_K) \propto p(x_I, x_J, x_K) \propto \prod_a \psi_a(x_a) \propto \prod_{a \in \partial I} \psi_a(x_a)$ and $p(x_I|x_J) \propto p(x_I, x_J) \propto \sum_{x_K} \prod_a \psi_a(x_a) \propto \prod_{a \in \partial I} \psi_a(x_a)$. As both distributions are normalized wrt x_I , they must be equal. This implies that

$$\begin{aligned} p(x_I, x_K|x_J) &= p(x_I, x_J, x_K) p(x_J)^{-1} \\ &= p(x_I|x_J, x_K) p(x_J, x_K) p(x_J)^{-1} \\ &= p(x_I|x_J) p(x_K|x_J) \end{aligned}$$

where the last line follows from the derivation above.

Lemma 2 (Separated neighborhood). *Let $p(x) = \frac{1}{Z} \prod_a \psi_a(x_a)$, and $G = (V \cup A, E)$ be the associated bipartite factor graph, with $E = \{(i, a) \in V \times A : \psi_a \text{ depends on variable } x_i\}$. Let $I \cup J \cup K \subseteq \{1, \dots, N\}$ be disjoint and assume that every path from I to K in G must pass through J (equivalently, removing vertices in J leave I and K separated). Then $p(x_I, x_K|x_J) = p(x_I|x_J)p(x_K|x_J)$, and $p(x_I|x_{J \cup K}) = p(x_I|x_J)$*

Proof. Let I' be the connected component of I in $G \setminus J$ and $K' = V \setminus (I' \cup J)$. As all paths from I to K pass through J , no factors can depend on variables both in I' and in K' . By Lemma 1, $p(x_{I'}, x_{K'}|x_J) = p(x_{I'}|x_J)p(x_{K'}|x_J)$. Then $p(x_I, x_K|x_J) = \sum_{x_{I' \setminus I}} \sum_{x_{K' \setminus K}} p(x_{I'}, x_{K'}|x_J) = \sum_{x_{I' \setminus I}} p(x_{I'}|x_J) \sum_{x_{K' \setminus K}} p(x_{K'}|x_J) = p(x_I|x_J)p(x_K|x_J)$. Then $p(x_I|x_{J \cup K}) = p(x_I, x_K|x_J)p(x_K|x_J)^{-1} = p(x_I|x_J)$.

Corollary 1 (Restricted autoregression). *By calling $I = \{i\}$, $J = \{j < i : j \in \partial i\}$ and $K = \{j < i : j \notin \partial i\}$, we obtain that for an ordering of nodes such that J separates i from K we get $p(x_i|\{x_j : j \in \partial i, j < i\}) = p(x_i|\{x_j : j < i\})$.*

The last corollary is defined for a single node i . In considering the problem of approximating the posterior probability of an epidemic spreading process we distinguish two graphs: the first one, the *contact* graph, the nodes are the time trajectory of the states $\underline{x}_i$ of each individuals and

the edges are the contacts between them. The second, the *factor* graph, has as nodes, again, the time trajectory of individuals and as factors those $(\{\Psi_i\})$ in the equation 6 in the main text. If the *contact* graph is acyclic and the nodes are topological ordered¹ then, thanks to the corollary 1, the following identity holds:

$$P(\underline{\mathbf{x}}|\mathcal{O}) = \prod_i P(\underline{x}_i|\underline{\mathbf{x}}_{<i}) = \prod_i P(\underline{x}_i|\underline{\mathbf{x}}_{\partial^2<i}) \quad (4)$$

where $\underline{\mathbf{x}}_{\partial^2<i}$ define the sets of nodes up to the next nearest neighbors (in the *contacts* graph) with index lower than i . The set of nodes $\underline{\mathbf{x}}_{\partial^2<i}$ in the *contact* graph correspond to the $\underline{\mathbf{x}}_{\partial<i}$ in the *factor* graph. To better visualize the two graphs, in Fig.3 we show two cases of *contact* acyclic graphs. The first one represents a linear chain defined by the contacts among individuals. The corresponding *contact* graph is acyclic but the *factor* graph contains cycles. In this case, for instance, considering the conditional probability of node 7 we have that $P(\underline{x}_7|\underline{\mathbf{x}}_{<7}) = P(\underline{x}_7|\underline{\mathbf{x}}_{\partial^2<7}) = P(\underline{x}_7|\underline{x}_4, \underline{x}_6)$. The node 4, 6 separated the nodes 7 from the previous nodes in the *factor* graph, as the corollary 1 require. Similarly, the same concept apply to the second case, where we consider a tree graph. In both cases the nodes are ordered topologically so the equation (4) holds for each conditional probability.

4 Soft margin estimator

We also use the Soft Margin estimator from [6] for finding the patient zero. This is a Monte Carlo based estimator, which applies Bayes formula to estimate the source probability $\mathbb{P}(s = i)$: given an epidemic $\mathbf{x}(i)$ which is the result of the simulation, starting from individual i , and the set of observation $\mathbf{x}_O$ on the epidemic, the probability of node k being the source of the epi-

¹A *topological* ordered graph is such that for each pair of nodes (i, j) with $i < j$, we have that $d_i \leq d_j$, where d_i is the distance of node i from the node zero. In practice, defining a starting node (a root) we enumerate the nodes according to the distance from the root (in a acyclic graph there is a unique path connecting each nodes) [5].

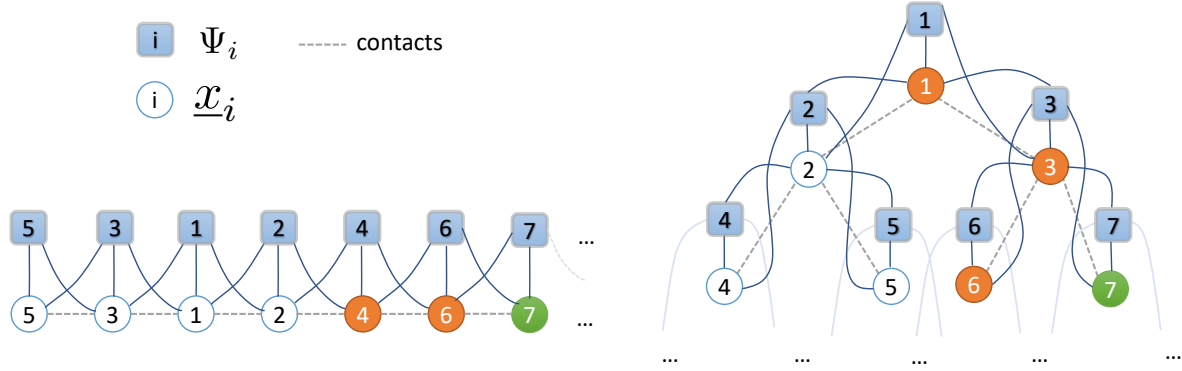


Figure 3: Two example of acyclic contact graphs ordered topologically. The two cases represent a linear chain on the left, and a tree on the right. The square represents the factor Ψ_i of the posterior probability of the dynamical process we want approximate. The circle represents the time trajectory of the states $\underline{x}_i$ of each individuals. The dashed lines are the contacts among individuals, and in both cases they generate a acyclic *contact* graph. Instead For instance the nodes 7 is separated (in the *factor* graph) from the rest of graph containing the nodes with minor index by the orange nodes.

demic is

$$\mathbb{P}(s = k \mid \mathbf{x}_O) \propto P(\mathbf{x}_O \mid s = k) P(i) \quad (5)$$

We have extended the method to the SIR model. To evaluate $P(\mathbf{x}_O \mid s = k)$, we run M Monte Carlo simulations in which the source of the epidemic is k and compare the resulting epidemic cascade $\mathbf{x}_i$ with the observations. For this we use the Jaccard similarity function $\phi_X(\mathbf{x}_i, \mathbf{x}_O)$, relating how many individuals are in state X (either infected, $X = I$ or recovered, $X = R$) in the generated cascade and the observations. Clearly, if no individuals are observed in state X , then $\phi_X = 1$ regardless of the realization. The probability of observing a certain configuration, given the source of the epidemic, can then be written as:

$$P(\mathbf{x}_O \mid s = k) = \frac{1}{M} \sum_{i=1}^M \exp \left\{ -\frac{(1 - \phi_I(\mathbf{x}_i, \mathbf{x}_O))^2 + (1 - \phi_R(\mathbf{x}_i, \mathbf{x}_O))^2}{a^2} \right\} \quad (6)$$

where the coefficient a regulates the sharpness of the peak around the case of perfect matching of observations ($\phi_I = 1$ and $\phi_R = 1$), where the corresponding value in the sum is 1. For the patient zero problems, we select the alpha values that give the better results (retrospectively) because good alpha values seem strongly dependent on the contact graphs and the number of samples considered.

References

- [1] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, January 2017. arXiv: 1412.6980.
- [2] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.
- [3] to release after acceptance. *to release*.
- [4] to release after acceptance. *to release*.
- [5] Donald Knuth. *The art of computer programming*. Addison-Wesley, Reading, Mass, 1997.
- [6] Nino Antulov-Fantulin, Alen Lančič, Tomislav Šmuc, Hrvoje Štefančić, and Mile Šikić. Identification of Patient Zero in Static and Temporal Networks: Robustness and Limita-

tions. *Physical Review Letters*, 114(24):248701, June 2015. Publisher: American Physical Society.