A Proofs

A.1 Proof of invariance with respect to $\lambda_{\max}$

To prove that Algorithm 1 is invariant with respect to the choice of $\lambda_{\max}$ as long as $\lambda_{\max} \geq \lambda_m(t)$ for all $m \in \mathcal{M}$ and $t \in \mathbb{R}^+$, we only need to prove that

$$\text{(i)} \quad \lambda_m(t) P^{\mathcal{C} \mid X=1, \Lambda=\lambda_m(t)\,;\,\mathrm{do}(\Lambda=\lambda_{m'}(t))}(X) \quad \text{whenever} \quad \lambda_m(t) < \lambda_{m'}(t)$$

and

$$\text{(ii)} \quad \lambda_m(t) + (\lambda_{max} - \lambda_m(t))\, P^{\mathcal{C} \mid X=0, \Lambda=\lambda_m(t)\,;\,\mathrm{do}(\Lambda=\lambda_{m'}(t))}(X) \quad \text{whenever} \quad \lambda_m(t) \geq \lambda_{m'}(t)$$

are invariant to the specific choice of λ_{max}.

To prove that (i) is invariant, we first rewrite the counterfactual probability $P^{\mathcal{C} \mid X=1, \Lambda=\lambda_m(t)\,;\,\mathrm{do}(\Lambda=\lambda_{m'}(t))}$ in terms of uniform random variables, following Section 4.2 in Huijben et al. [55]:

$$P^{\mathcal{C} \mid X=1, \Lambda=\lambda_m(t)\,;\,\mathrm{do}(\Lambda=\lambda_{m'}(t))}(X=1) =$$
$$\mathbb{P}_{U_0,U_1}\left[-\log\left(-\log(U_1)\right) + \log\left(\frac{\lambda_{m'}(t)}{\lambda_{\max}}\right) - \log\left(\frac{\lambda_m(t)}{\lambda_{\max}}\right) > \right.$$
$$\left. -\log\left(-\log(U_1) - \frac{\log(U_0)}{1-\frac{\lambda_m(t)}{\lambda_{\max}}}\right) + \log\left(1-\frac{\lambda_{m'}(t)}{\lambda_{\max}}\right) - \log\left(1-\frac{\lambda_m(t)}{\lambda_{\max}}\right)\right]$$

where $U_0, U_1 \sim U[0,1]$. Now, for a fixed U_0 and U_1, the above inequality can be rewritten as follows:

$$-\log\left[-\log(U_1)\frac{\lambda_m(t)}{\lambda_{\max}}\frac{\lambda_{\max}}{\lambda_{m'}(t)}\right] > -\log\left[\left(-\log(U_1) - \frac{\log(U_0)}{1-\frac{\lambda_m(t)}{\lambda_{\max}}}\right)\frac{1-\frac{\lambda_m(t)}{\lambda_{\max}}}{1-\frac{\lambda_{m'}(t)}{\lambda_{\max}}}\right]$$
$$-\log(U_1)\frac{\lambda_m(t)}{\lambda_{m'}(t)}\left(1-\frac{\lambda_{m'}(t)}{\lambda_{\max}}\right) < -\log(U_1)\left(1-\frac{\lambda_m(t)}{\lambda_{\max}}\right) - \log(U_0)$$
$$-\log(U_1)\left(\frac{\lambda_m(t)}{\lambda_{m'}(t)} - \frac{\lambda_m(t)}{\lambda_{\max}}\right) < -\log(U_1)\left(1-\frac{\lambda_m(t)}{\lambda_{\max}}\right) - \log(U_0)$$

In the last inequality, the terms containing $\lambda_{\max}$ on the left and right hand side are identical and can be canceled. This proves that (i) is invariant to the specific choice of λ_{max}.

To prove that (ii) is also invariant, we proceed similarly as in (i) and first rewrite the counterfactual probability $P^{\mathcal{C} \mid X=0, \Lambda=\lambda_m(t)\,;\,\mathrm{do}(\Lambda=\lambda_{m'}(t))}$ in terms of uniform random variables:

$$P^{\mathcal{C} \mid X=0, \Lambda=\lambda_m(t)\,;\,\mathrm{do}(\Lambda=\lambda_{m'}(t))}(X=1) =$$
$$\mathbb{P}_{U_0,U_1}\left[-\log\left(-\log(U_0) - \frac{\log(U_1)}{\frac{\lambda_m(t)}{\lambda_{\max}}}\right) + \log\left(\frac{\lambda_{m'}(t)}{\lambda_{\max}}\right) - \log\left(\frac{\lambda_m(t)}{\lambda_{\max}}\right) > \right.$$
$$\left. -\log\left(-\log(U_0)\right) + \log\left(1-\frac{\lambda_{m'}(t)}{\lambda_{\max}}\right) - \log\left(1-\frac{\lambda_m(t)}{\lambda_{\max}}\right)\right]$$

where $U_0, U_1 \sim U[0,1]$. Now, for a fixed U_0 and U_1, the above inequality can be rewritten as follows:

$$\log(U_0)\left(1-\frac{\lambda_m(t)}{\lambda_{m'}(t)}\right) \leq \log(U_1)\left(\frac{\lambda_{\max}}{\lambda_{m'}(t)} - 1\right)$$

Now, using the fact that $-\log(U_0)$ and $-\log(U_1)$ are distributed as exponential random variables with rate $\lambda = 1$ and the CDF of the ratio $X = \log(U_0)/\log(U_1)$ of two exponential random variables with rate λ_1 is given by $\mathbb{P}[X \leq x] = 1/(1/x+1)$, we have that:

$$P^{\mathcal{C} \mid X=0, \Lambda=\lambda_m(t)\,;\,\mathrm{do}(\Lambda=\lambda_{m'}(t))}(X=1) = \mathbb{P}_{U_0,U_1}\left[\frac{\log(U_0)}{\log(U_1)} > \frac{\frac{\lambda_{\max}}{\lambda_{m'}(t)} - 1}{1-\frac{\lambda_m(t)}{\lambda_{m'}(t)}}\right] = \frac{\lambda_{m'}(t) - \lambda_m(t)}{\lambda_{\max} - \lambda_m(t)}.$$

Then, it readily follows that

$$(\lambda_{\max} - \lambda_m(t))\,\frac{\lambda_{m'}(t) - \lambda_m(t)}{\lambda_{\max} - \lambda_m(t)} = \lambda_{m'}(t) - \lambda_m(t).$$

This proves that (ii) is invariant to the specific choice of λ_{max}.

A.2 Proof of Proposition 1

If $\lambda_m(t_i) \geq \lambda_{m'}(t_i)$, then we have:

$$\lambda_m(t_i) \geq \lambda_{m'}(t_i) \implies \lambda_m(t_i)\left(1 - \frac{\lambda_{m'}(t_i)}{\lambda_{\text{max}}}\right) \geq \lambda_{m'}(t_i)\left(1 - \frac{\lambda_m(t_i)}{\lambda_{\text{max}}}\right) \implies \frac{1 - \frac{\lambda_{m'}(t_i)}{\lambda_{\text{max}}}}{1 - \frac{\lambda_m(t_i)}{\lambda_{\text{max}}}} \geq \frac{\frac{\lambda_{m'}(t_i)}{\lambda_{\text{max}}}}{\frac{\lambda_m(t_i)}{\lambda_{\text{max}}}},$$

Now, by Eq. 5, the last inequality is equivalent to:

$$\frac{P^{\mathcal{C}\,;\,\text{do}(\Lambda=\lambda_{m'}(t_i))}(X=0)}{P^{\mathcal{C}\,;\,\text{do}(\Lambda=\lambda_m(t_i))}(X=0)} \geq \frac{P^{\mathcal{C}\,;\,\text{do}(\Lambda=\lambda_{m'}(t_i))}(X=1)}{P^{\mathcal{C}\,;\,\text{do}(\Lambda=\lambda_m(t_i))}(X=1)}, \tag{7}$$

which is exactly the counterfactual stability property. Finally, by Theorem 2 in [21], we know that the Gumbel-Max SCM satisfies the counterfactual stability property. As a result, Eq. 7 implies that

$$P^{\mathcal{C}\,|\,X=0,\Lambda=\lambda_m(t_i)\,;\,\text{do}(\Lambda=\lambda_{m'}(t_i))}(X=1) = 0,$$

as desired. The proof for the second case is exactly the same.

B Algorithms

B.1 Lewis' thinning algorithm

Within Algorithm 4, lines 4–6 sample an event from a Poisson process with constant intensity $\lambda_{\max}$ using inversion sampling and lines 11–14 accept/reject the event according to the ratio between the intensity of interest and $\lambda_{\max}$ at the time of the event.

Algorithm 4 Lewis' thinning algorithm

1: **Input**: $\lambda(t)$, $\lambda_{\max}$, T.
2: **Initialize**: $s = 0, \mathcal{H} = \emptyset$.
3: **function** LEWIS($\lambda(t), \lambda_{\max}, T$)
4: **while** true **do**
5: $u_1 \sim \text{Uniform}(0, 1)$
6: $w \leftarrow -\ln u / \lambda_{\max}$
7: $s \leftarrow s + w$
8: **if** $s > T$ **then**
9: break
10: **end if**
11: $\mathcal{H}_{\max} \leftarrow \mathcal{H}_{\max} \cup \{s\}$
12: $p \leftarrow \lambda(s)/\lambda_{\max}$
13: $u_2 \sim \text{Uniform}(0, 1)$
14: **if** $u_2 \leq p$ **then**
15: $\mathcal{H} \leftarrow \mathcal{H} \cup \{s\}$
16: **end if**
17: **end while**
18: **return** $\mathcal{H}, \mathcal{H}_{\max} \backslash \mathcal{H}$
19: **end function**

B.2 Thinning algorithm for Hawkes processes

Algorithm 5 samples a sequence of events from a linear Hawkes process using its branching process interpretation [49], where LEWIS$(\cdot)$ samples a sequence of events using Algorithm 4, $\gamma_0(t) = \mu$ and $\gamma_i(t) = \alpha g(t - t_i)$.

Algorithm 5 It samples a sequence of events from a linear Hawkes process using its branching process interpretation.

1: **Input**: μ, α, $g(t)$, $\lambda_{\max}$, T.
2: **Initialize**: $\mathcal{H} = \emptyset$.
3: **function** SAMPLEHAWKES($\lambda(t), \lambda_{\max}, T$)
4: $\mathcal{H} \leftarrow$ LEWIS$(\gamma_0(t), \lambda_{\max}, T)$
5: $\mathcal{H}' \leftarrow \mathcal{H}$
6: **while** $|\mathcal{H}'| > 0$ **do**
7: $t_i \leftarrow \min_{t \in \mathcal{H}'} t$
8: $\mathcal{H}_i, \_ \leftarrow$ LEWIS$(\gamma_i(t), \lambda_{\max}, T)$
9: $\mathcal{H}' \leftarrow \mathcal{H}_i \cup \mathcal{H}' \backslash \{t_i\}$
10: $\mathcal{H} \leftarrow \mathcal{H} \cup \mathcal{H}_i$
11: **end while**
12: **return** $\mathcal{H}$
13: **end function**

B.3 Sampling from the Gumbel noise posterior

Algorithm 6 draws a sample from the noise posterior $P^{\mathcal{C} \mid X_i = x_i, \Lambda_i = \lambda_m(t_i) \,;\, \text{do}(\Lambda_i = \lambda_{m'}(t_i))}(U_{i,x})$ using the idea of A* sampling [48]. In the noise posterior, the maximum value and the argmax of the shifted Gumbel variables are independent, and the maximum value has a standard Gumbel distribution.

Therefore, one can sample the maximum value and then sample the remaining values from the shifted Gumbel distributions truncated at this maximum value, which we denote as $\textsc{TruncGumbel}(\cdot)$.

Algorithm 6 It draws a sample from the posterior of a Gumbel noise variable using A* sampling [48].

Input: $\mathcal{H}_m$, $\mathcal{H}_{\max} \backslash \mathcal{H}_m$, $\lambda_m(t)$, $\lambda_{m'}(t)$, $\lambda_{\max}$.
Initialize: $G \sim \textsc{Gumbel}(0, 1)$, $\alpha_1 = \lambda_m(t)/\lambda_{\max}$, $\alpha_0 = 1 - \alpha_1$.

if $t_i \in \mathcal{H}_m$ **then**
  $P^{\mathcal{C} \mid X=1,\, \Lambda=\lambda_m(t)\,;\, \mathrm{do}(\Lambda_i=\lambda_{m'}(t))}(U_0) = \textsc{TruncGumbel}(\log(\alpha_0), G) - \log(\alpha_0)$
  $P^{\mathcal{C} \mid X=1,\, \Lambda=\lambda_m(t)\,;\, \mathrm{do}(\Lambda_i=\lambda_{m'}(t))}(U_1) = G - \log(\alpha_1)$
else
  $P^{\mathcal{C} \mid X=0,\, \Lambda=\lambda_m(t)\,;\, \mathrm{do}(\Lambda_i=\lambda_{m'}(t))}(U_0) = G - \log(\alpha_0)$
  $P^{\mathcal{C} \mid X=0,\, \Lambda=\lambda_m(t)\,;\, \mathrm{do}(\Lambda_i=\lambda_{m'}(t))}(U_1) = \textsc{TruncGumbel}(\log(\alpha_1), G) - \log(\alpha_1)$
end if

B.4 Sampling Counterfactual Events in a SIR Epidemic Model

As discussed in Section 6, the SIR model defined by Eq. 6 can be viewed as a (networked) multidimensional Hawkes process with stochastic triggering kernels defined by step functions. More specifically, let t_i and $\tau_i = t_i + \Delta_i$ be the infection and recovery times of each node $i \in \mathcal{G}$. Then, the intensity $\mathbb{E}[dY_i(t) \,|\, \mathcal{H}(t)]$ can be (re-)written as:

$$\mathbb{E}[dY_i(t) \,|\, \mathcal{H}(t)] = \beta \sum_{j \in \mathcal{G}(i)} g_i(t - t_j)$$

where $\mathcal{G}(i)$ denotes the set of neighborhood of node i, $g_i(t) = [1 - Y_i(t)][u(t) - u(t - \Delta)]$ with $\Delta \sim \mathrm{Exp}(\delta)$ can be viewed as a stochastic triggering kernel, and $u(\cdot)$ is the step function. As a result, we can adapt Algorithm 3, originally developed for unidimensional linear Hawkes processes, to sample counterfactual realizations of the SIR model. Algorithm 7 summarizes the resulting algorithm.

Algorithm 7 It samples a counterfactual sequence of infections given a sequence of observed infections from the SIR process defined by Eq. 6.

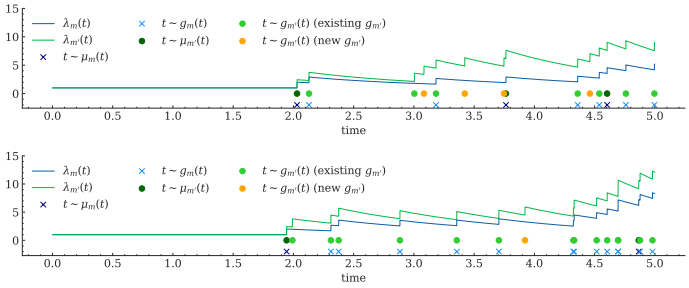
Input: β_m, δ, $\beta_{m'}$, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$, T.
Initialize: queue = PriorityQueue($\{i \mid i \in \mathcal{V}$ and is_seed(i)$\}$), $\beta_{max} = max(\beta_m, \beta_{m'})$, processed = $\{\}$, infector = $\{\}$.

for $i \in \mathcal{G}$ **do**
  processed$[i]$ = False
  if is_seed(i) **then**
    $t_i = 0$
    $\tau_i = t_i + \exp(\delta)$
  else
    $t_i = \infty$
  end if
end for

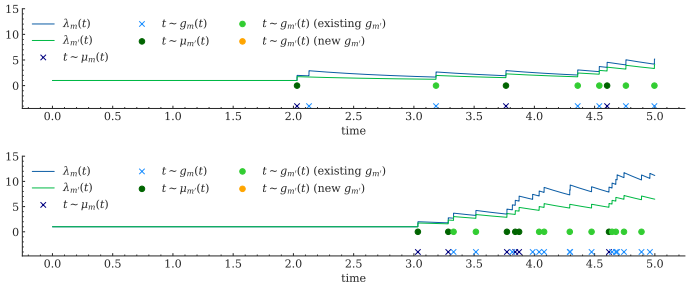
while $\neg$ queue.isEmpty() **do**
  i = queue.*pop*()
  if $\neg$ processed$[i]$ **then**
    processed$[i]$ = True
    for $j \in \mathcal{G}'$(i) **do**
      $\gamma_{m'}(t) = \beta_{m'} u(t - t_i) - \beta_{m'} u(t - \tau_i)$
      if infector$[j]$ == i **then**
        $t = t_j$
        $\gamma_m(t) = \beta_m u(t - t_i) - \beta_m u(t - min(\tau_i, t))$
        $\mathcal{H}_{m'} = \textsc{Cf}(\gamma_m(t), \gamma_{m'}(t), \{t\}, \beta_{max}, T)$
        if $\mathcal{H}_{m'} \neq \emptyset$ **then**
          $t = \min_{t' \in \mathcal{H}_{m'}} t'$
        else
          $t = \infty$
        end if
      else
        $\mathcal{H}, \_ = \textsc{Lewis}(\gamma_{m'}(t), \beta_{max}, T)$
        $t = \min_{t' \in \mathcal{H}} t'$
      end if
      if $t < t_j$ **then**
        $t_j = t$
        $\tau_j = t_j + \exp(\delta)$
        infector$[j] = i$
        queue.*add*$(j, \text{priority} = t_j)$
      end if
    end for
  end if
end while

C Additional Results on Synthetic Data

Figure 7 shows the Hawkes intensities and event times corresponding to specific realizations of the original process and the counterfactual processes.



(a) Intervention increases self-excitation, $\alpha_{m'} > \alpha_m$



(b) Intervention decreases self-excitation, $\alpha_{m'} < \alpha_m$

Figure 7: Effect of interventions in Hawkes processes (continued). Panels show the Hawkes intensities and event times corresponding to specific realizations of the original process and the counterfactual processes. Crosses (dots) denote events of the original (counterfactual) processes and the color pattern indicates which type of process generated each event (be it $\mu_{m'}$ or $g_{m'}$ due to a counterfactual event that also existed (did not exist) in the original realization). The parameters of the original and intervened process are the same as in Figure 2.

Figure 8 shows the estimated counterfactual thinning probability for five different events of the same inhomogeneous Poisson process considered in the main paper against the number of samples from the Gumbel noise distribution. The results show that, as the number of noise sample increases, the variance of the counterfactual thinning probability quickly decreases.

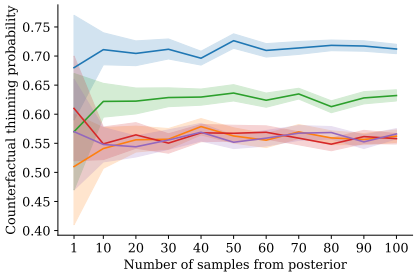


Figure 8: Sensitivity analysis of the Monte-Carlo approximation of the counterfactual thinning probability used in Algorithm 1.

D Additional Details about the Experiments on Real Data

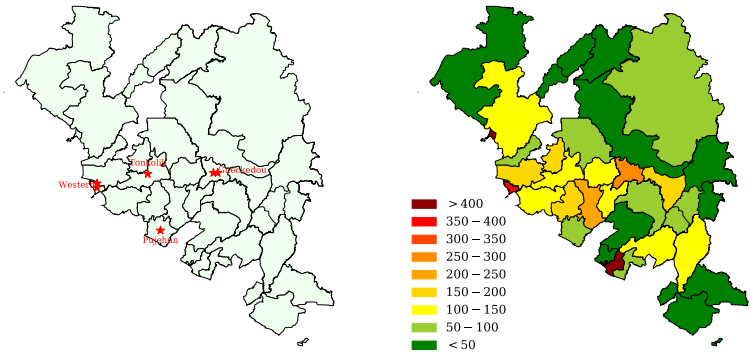
To set the β and δ, we resort to well-known epidemiological quantities whose values have been estimated by the World Health Organization (WHO) for the Ebola outbreak we study [56]. More specifically, the average generation time, *i.e.*, the time between the infection of a primary case and one of its secondary cases [57], is $1/\beta \approx 15.3$ days and the mean time from the onset of symptoms to death or discharge from the hospital is $1/\delta \approx 11.4$ days.

To generate the contact network $\mathcal{G}$ which underpins our model, we use a stochastic block model network whose parameters are informed by estimates of the reproduction numbers for each of the countries (Guinea, Liberia, and Sierra Leone) affected by the Ebola outbreak. For computational reasons, for each of the 55 districts where infection cases were identified, we create a set of nodes $\mathcal{V}_i$ proportional to the population in the district, as estimated by the WorldPop project [58], and assume $\cup_i \mathcal{V}_i = \mathcal{V}$ with $|\mathcal{V}| = \sum_i |\mathcal{V}_i| = 8{,}000$ individuals. Moreover, we add an edge between each pair of nodes (u, v) nodes independently at random with probability $p(u, v) = 10^{-2}$ if (u, v) are in the same district, $p(u, v) = 2.15 \cdot 10^{-3}$ if they are in two contiguous districts in Guinea, $p(u, v) = 3 \cdot 10^{-3}$ if they are in two continuous districts in Liberia, $p(u, v) = 3.15 \cdot 10^{-3}$ if they are in two continuous districts in Sierra Leona, and $p(u, v) = 1.9 \cdot 10^{-3}$ if they are in two continuous districts in different countries.

In the above, we found contiguous districts using publicly available district-level shapefiles[11] and set the values of the between-district probabilities using grid-search so that, at a country level, the basic reproduction number of the simulated outbreaks matches that estimated from real data (refer to Table 1).

Country	R_0 (WHO)	R_0 (simulated)
Guinea (GN)	1.71 (1.44 – 2.01)	1.71 (1.66 – 1.76)
Liberia (LB)	1.83 (1.72 – 1.94)	1.83 (1.74 – 1.91)
Sierra Leone (SL)	2.02 (1.79 – 2.26)	2.02 (1.95 – 2.10)

Table 1: Reproduction numbers (R_0) for each of the three countries affected by the Ebola outbreak estimated from real cases by WHO [56] and estimated from simulated cases. In each cell, the first number is the average and the numbers in parentheses are the 95% confidence interval.



(a) Seed infections (b) Overall cumulative infections

Figure 9: Geographical distribution of infections in a *realistic* Ebola outbreak in West Africa. Panel (a) shows the seed infections in each district, where each red star represents an infection. Panels (b) shows the overall cumulative number of infections per district. Darker red (green) corresponds to high (low) number of cumulative infections.

[11] https://data.humdata.org/