Politecnico di Torino

Politecnico di Torino

Master's Degree in Physics of Complex Systems

Academic year 2024/2025

Graduation Session July 2025

Epidemic Inference in Metapopulation Models: optimization algorithm through forward-backward propagation

Supervisors:

Prof. Luca DALL'ASTA

Mr. Mattia TARABOLO

Dr. Eugenio VALDANO

Candidate:

Alessandro CALCIANO

Abstract

The epidemic inference allows us to make predictions on the evolution of an epidemic to develop containment measures or infer the infection channels and the origin of the epidemic.
In this thesis work, we discuss a metapopulation structure approach to stochastic inference, where the total population is partitioned in many subpopulations which interact with each other.
We assume to receive daily information about the aggregate number of infected individuals at the single population level. The information from such time-scattered observations together with some prior information can be used to develop a Bayesian framework to address different epidemic inference problems such as to predict the future evolution of the outbreak (epidemic forecast), to infer the current state of the epidemics in unobserved populations (risk assessment), or to infer the past state of the epidemics (causal paths, patient zero).
To address these problems, we provide a common Bayesian framework to compute the joint probability of the overall history $\mathcal{H}$ of the metapopulation system given a set of observations $\mathcal{O}$. We derive the prior distribution $P[\mathcal{H}]$ associated with the stochastic metapopulation system, as well as the likelihood of the data $P[\mathcal{O}|\mathcal{H}]$, which is derived from a simple Gaussian sampling process for the observations.
The prior distribution is formulated using a Path Integral approach to the underlying stochastic process. By applying a saddle-point method, we derive a set of deterministic equations for the epidemic state variables and their associated conjugate fields. These equations are integrated using a forward-backward algorithm, which efficiently identifies the most probable epidemic trajectory consistent with the observed data.
The numerical results demonstrate a strong agreement between the inferred trajectories and ground truth simulations, validating the effectiveness of the proposed framework. Additionally, due to numerical challenges and the complexity of the underlying optimization landscape, we implement gradient-based optimization methods, which enhance convergence and allow for more stable and accurate inference.

Acknowledgements

First of all, I would like to express my profound gratitude to Professor Luca Dall'Asta, who gave me the opportunity to challenge myself with such a fascinating topic as epidemic inference. He has been an inspiration through his way of approaching problems, greatly helping me to understand crucial aspects, and always being present and available. I could not have asked for a better professor to work with.

A special thanks goes to Dr. Mattia Tarabolo, who always found time to support me throughout the coding phase, from the very beginning to the final stages of this work, guiding me step by step in the development of the project.

I would also like to thank Dr. Eugenio Valdano, who provided us with the data used in the thesis project, and gave me valuable advice on how to use them effectively.

All of this would not have been possible without their contribution, making me proud of the choice I made seven months ago.

I also want to thank the Politecnico di Torino which, despite the challenges faced over the years, has allowed me to acquire new knowledge and critical thinking skills, and has made me develop a true passion for learning.

I wish to express my deepest thanks to my family, who have always been there for me, physically and, above all, with their hearts. Through many sacrifices, they have made this journey possible, and I will never be able to thank them enough. I love you all.

Last but not least, I want to thank Marilena, my other half, the person who has always cheered for me and supported me in every choice I have made. I love you.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 State of the art

In recent decades, epidemic modeling has been a widely studied topic in statistical physics and has played a central role in public healthcare, especially with the emergence of global epidemics such as SARS, Ebola, and the latest COVID-19.
The ability to perform epidemic inference is crucial: it encompasses a set of methodologies aimed at estimating the key parameters that regulate the dynamics of an epidemic, starting from a partial knowledge of the data.
The starting point for this problem can be recovered in compartmental models like SIR, SEIR, and their variants, which aim to simplify the overall vision of the epidemic by compartmentalizing the population (e.g. susceptibles, infected, removed, and so on), and formalizing the dynamics through differential equations that depict the several transitions from a group to another.
Many studies, starting from the seminal work of Kermack and McKendrick, [1], have used compartmental models to understand the main properties of epidemic dynamics; however, the application to real data remains challenging.
In such context, several approaches have been proposed to make inference, for example, the Maximum Likelihood Estimate(MLE) [2, 3], but they are of limited effectiveness when working with systems affected by strong stochasticity or characterized by a small amount of data.
In recent years, approaches based on Bayesian Inference [4][5, section 4] have become widespread since it allows to quantify the uncertainty of the estimated parameters working with probability distributions. In particular, the integration of Monte Carlo Markov Chain(MCMC) techniques [6] within Bayesian approaches makes it possible to exploit the theory of large numbers to perform accurate estimates of the epidemic dynamics through a sampling process. This approach can be used for different tasks, such as the estimate of the parameters(β, γ, μ that

lead the evolution of the epidemic [7], or the estimate of the trajectories for the compartments of the epidemic model, as one can see in the work of Dureau et al. [8], where the authors developed a stochastic SEIR model with time-variable parameters and used an adaptive Particle MCMC to estimate both parameters and trajectories in time.
Moreover, in several studies it has been shown that a Bayesian approach can be effective in adapting to the current dynamical trends and produce good real-time estimates [9, 10].
With the increasing amount of available data, machine learning approaches have become more and more widespread. Among them, neural networks, in particular Graph Neural Networks(GNN), have gained prominence. In GNN, each node is considered as part of a structure of connections with other nodes, enabling this model to to capture the dependencies between neighboring nodes and to learn about the graph structure in details. GNN are typically composed of multiple layers where each layer plays a specific role in the entire learning process. Several applications of this approach have been proposed, see for instance the work by Fritz et al. (2022) [11], in which the Authors combine GNNs with spatio-temporal disease description to make better predictions of the weekly COVID-19 cases in Germany, or the work of Song et al. [12], in which they combine GNNs with epidemic data and mobility to improve inference of the dynamics, in particular, the use of GNN allows to exploit spatio-temporal relations between nodes, improving the outbreak forecasting and risk assessment process.
Similar applications can be found in the work of Shah, Dehmamy, Yu et al.(2020)[13].
Despite the huge amount of studies and progress made on this topic, criticalities about the quality of data, modeling of complex dynamics(e.g. heterogeneity of contacts, mobility) are still present.
In this perspective, the present thesis work aims to apply a metapopulation-based approach to an epidemic compartmental model for the purpose of epidemic inference in realistic settings. To this end, real-world colocation data is used to capture the spatial interaction structure among subpopulations.
The observations used to validate the inference technique are generated through Monte Carlo simulations, allowing the exploration of plausible epidemic trajectories under stochastic dynamics and uncertainty.
This framework enables a more accurate representation of the dynamics of contagion by incorporating both the stochastic nature of disease spread and the spatial heterogeneity of interactions. The results contribute to a better understanding of inference methods based on stochastic models in metapopulation contexts and provide a practical tool for analyzing realistic epidemic scenarios.

1.2 Thesis structure

The thesis is structured as follows:

- **Chapter 2**: This chapter aims to introduce the reader to fundamental concepts of graph theory and statistical inference, along with epidemic models and the approach applied to the problem.
  Namely, it provides an overview of graph theory, with a focus on its applications in modeling interactions in epidemic settings; then we classify different types of epidemic compartmental models,especially SIR and SEIR, along with the associated state equations, and exploit the metapopulation approach as a way to incorporate spatial heterogeneity.
  Finally we outline the fundamental of Bayesian Inference, which represents the statistical framework for the inference methods used in the thesis.

- **Chapter 3**: It focuses on the path-integral formulation of stochastic epidemic models. The chapter begins with a review of the formulation and its main physico-mathematical concepts, laying the groundwork for the derivation of the system's partition function $\mathcal{Z}$ and the corresponding action $\mathcal{S}$.
  Finally, the mean-field approximation is introduced along with its fundamentals and the application to the present case to obtain tractable estimates of the system's behavior, working in the small rates-small fluctuations regime.

- **Chapter 4**: This chapter is dedicated to the Saddle-Point method, a key analytical tool used to approximate the behavior of the system and, in the present work, to obtain a Maximum a Posteriori prediction of the behavior of the system. The saddle-point method is an elegant and powerful way to pass from the stochastic description of the epidemic dynamics (random trajectories) to a deterministic one (mean-field like trajectories).
  The chapter begins with a general overview of the method and its relevance in statistical physics and stochastic processes. It then focuses on its application to the current study, with the derivation of the saddle-point (or Euler-Lagrange) equations associated with the most probable epidemic trajectory.
  These equations are then used to construct the core of the inference algorithm that will be presented in the following chapters, as they govern the system's deterministic dynamics under the inference framework.

- **Chapter 5**: here the colocation dataset used in the inference algorithm is introduced. The data were provided by one of the thesis supervisors affiliated with INSERM (Institut National de la Santé et de la Recherche Médical) in Paris, and originate from geolocation information provided by META (Facebook, Instagram and so on).
  This dataset captures spatial proximity patterns between individuals from

different administrative regions, such as provinces in Italy, enabling the construction of the dynamic interaction networks used in the metapopulation model.
The chapter clarifies the data structure, preprocessing steps, and its integration into the model to represent spatial transmission dynamics.

- **Chapter 6**: In this chapter the implementation of forward-backward algorithm for the epidemic inference is presented. It outlines the structure of the algorithm and explains its role within the inference framework.
  Particular attention is dedicated to the explanation of the main components of the code, including the forward propagation of the epidemic states (S,E,I,R for the SEIR model), the backward update for the adjoint variables (which are represented as θ variables), and the update rules derived from the S-P equations from the chapter 4.
  This chapter connects the theoretical formulation with its computational realization, in such a way to retrieve a reproducible and interpretable inference.

- **Chapter 7**: It discusses the broader range of epidemiological problems that can be addressed using the inference framework developed in this thesis work. Building upon the theoretical and computational tools presented in the previous chapters, this chapter shows how the forward-backward algorithm can be used to extract meaningful information about the disease dynamics.
  In particular, it focuses on the task of assessing the risk of infection across different regions knowing some other information about other regions, and exploring retrospective scenario analysis based on observed data.
  Through these examples, the flexibility and generality of the developed method are highlighted, illustrating its potential for aiding public health decision-making, surveillance efforts, and policy planning.

- **Chapter 8**: The last chapter provides the concluding remarks of the thesis and outlines potential directions for future research.
  It begins by resuming the key contributions of the work, reflecting on the methodological developments and their application to real-world epidemiological data.
  This section continues with the discussion of the limitations associated with the current approach, including assumptions made in the model, computational constraints, and data availability. It eventually identifies several promising avenues for future investigation, such as extending the inference framework to other disease models, using heterogeneous behavioral data, introducing different kind of priors, or integrating real-time data streams for online epidemic tracking.
  In the end, this chapter highlights the role of the developed tools and concepts

as a ground for more advanced and flexible epidemic modeling frameworks, filling the gap between theoretical epidemiology and practical public health applications.

Chapter 2

Mathematical and Statistical Foundations of Epidemic Inference

2.1 Basic concepts of graph theory

A graph is a mathematical structure that describes the relationships between objects: these are called vertices or nodes, and they constitute the set denoted by $\mathbf{V}$. Each vertex/node will be indexed by a number such that $\mathbf{V} = [1,2,3,...,N]$, where N is the total number of nodes.
The relationships between nodes are described as connections, known as edges, which in turn form the set $\mathbf{E} \subset \mathbf{V} \times \mathbf{V}$. Each edge is represented as a couple of nodes, such as $e = (i, j)$, where i and j are two different nodes.
A graph can be classified depending on the type of interactions between nodes, namely we can have:

- **Directed** or **undirected**, depending on whether the connections have directionality or they work both ways.
- **Weighted** if each edge (i, j) is associated with a real number w_{ij} representing the strength of the contact.
- **Time-varying** (dynamic graphs), as you can well imagine, the structure and/or the weights of the network change in time, a common situation in epidemic spreading scenarios.

An **adjacency matrix** $A \in \mathbb{R}^{N \times N}$ can be defined to describe a graph, where each element A_{ij} indicates the presence and/or the weight of an edge between nodes i

and j. This can be formally represented as:

$$A_{ij} = \begin{cases} 1 & (i,j) \in E \\ 0 & else \end{cases}$$

In this thesis we will work with a colocation matrix W, where each element w_{ij} represents the weight of each connection, in particular it represents the rate at which an individual from region i and another one from region j are colocated at a given time in any other place. In fact, $w_{ii} \neq 0$, since it will represent the rate of colocation between individuals from the same region i. These data will be well described in **Chapter 5**.
The degree of a node is defined as the number of nodes to which it is connected; in fact, one can define the **neighborhood** of a node i as the list of nodes connected to it. This will be addressed as ∂i with the formal definition:

$$\partial i = \{j \in V : (i,j) \in E\} \tag{2.1}$$

and the degree of node i as $k_i = |\partial i|$.
From now on, we will work with a **undirected, weighted, time-varying** graph represented by the aforementioned colocation matrix W. Initially, we worked with fictitious colocation data, randomly generated by using some specific Julia libraries which generate different types of graphs, such as random regular graphs or Erdös-Renýi graphs.

2.1.1 Random regular graphs

A *random regular graph* is a graph in which each node has the same degree d, and edges are randomly assigned under this constraint. These graphs are useful in epidemic modeling because they provide a homogeneous interaction structure since each node is equally connected, allowing for controlled experiments on how the interaction network affects the spreading of the disease.
Efficient algorithms for generating such graphs have been developed, such as the one proposed by Steger and Wormald (1999) [14], which ensures a uniform distribution over all d-regular graphs.
An example of a random regular graph with degree $d = 3$ and 8 nodes is shown in Figure 2.1, where the uniform degree distribution is visually apparent.

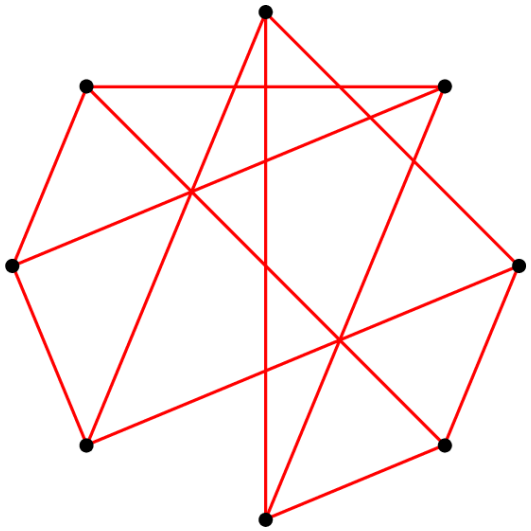

Figure 2.1: Random Regular Graph with fixed degree $d = 3$.

2.1.2 Erdös-Renýi graphs

The *Erdös-Renýi model* [15] represents one of the earliest and most studied formulations of random graphs. In the $G(n, p)$ variant, a graph is constructed on n nodes where each possible edge between pairs of nodes is included independently with a fixed probability p. This ensures that the number of edges is a random variable, and different realizations of the graph can lead to very different outcomes both in density and connectivity.
On the other hand, in the $G(n, m)$ variant, a graph is built by selecting uniformly at random precisely m edges from the $\binom{n}{2}$ possible edges between 2 nodes. In this way, the graph will always contain the same number of edges, which permits better control of the sparsity of the graph.
Asymptotically, the properties of $G(n, p)$ with $p = \frac{2m}{n(n-1)}$ converge to those of $G(n, m)$. Despite their simplicity and analytical tractability, these models serve as idealized baselines in many applications, such as epidemic modeling.
An example of both variants of a Erdös-Renýi random graph is shown in Figure 2.2, demonstrating variability in node degrees and the absence of global regularity.

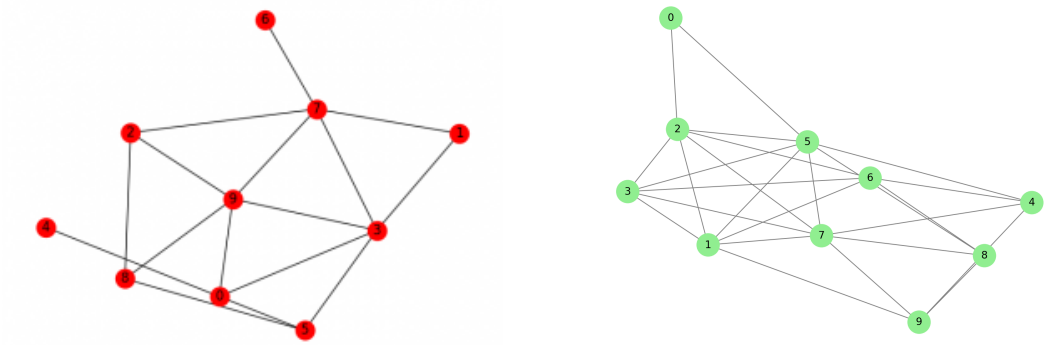

Figure 2.2: Comparison of two Erdős–Rényi random graphs with $n = 10$ nodes. On the left: a $G(n, p)$ graph with $p = 0.5$, source: GeeksforGeeks (2023) [16] . On the right: a $G(n, m)$ graph with $m = 25$ edges.

2.2 Epidemic models and Metapopulation approach

Understanding the dynamics of infectious diseases is essential for predicting outbreaks and informing public health interventions. In this sense, mathematical modeling is crucial since it provides a systematic framework for describing how diseases spread within and between populations.
Among them, compartmental models play a central role as the foundation of epidemic theory, dating back to the ground work by Kermack and McKendrick (1927), who introduced the classical SIR model [1]. These models divide the population into compartments, such as susceptible, infected, and recovered, and define transitions between compartments through differential equations.
In this section the main types of compartmental models will be presented, focusing on SIR and SEIR frameworks, along with their mathematical formulations. In order to describe properly the structure of such models, we also describe the metapopulation approach. This modeling strategy integrates network structures and spatial distribution, as seen in the work of Colizza et al. (2007) [17], allowing the analysis of disease dynamics between interconnected subpopulations.
These models constitute the ground for many modern epidemiological studies and are useful when integrated with inference techniques, allowing forecasting, risk assessment and patient zero analysis, as explored in later chapters of this thesis.

2.2.1 The homogeneous SIR model

The SIR model, as mentioned above, was introduced for the first time by Kermack and McKendrick (1927) [1],and represents the cornerstone of mathematical epidemiology for describing the spread of infectious diseases in a population or, as in the present work, in a network of subpopulations.
The homogeneous framework assumes that each individual has the same probability of contacting any other, so that we can ignore spatial or network structure. Assuming the total population considered is given by N, we will have three compartments:

- $S(t)$: susceptible individuals, namely individuals who can be infected if they contact with infectious individuals;
- $I(t)$: infectious individuals, who can infect susceptible individuals with which they contact, or can recover from the disease;
- $R(t)$: recovered (or removed) individuals, which are individuals that recovered from the disease with immunity, or died.

The dynamics can be described through the following set of ordinary differential equations [18]:

$$\frac{dS}{dt} = -\beta \frac{SI}{N}, \tag{2.2a}$$

$$\frac{dI}{dt} = \beta \frac{SI}{N} - \mu I, \tag{2.2b}$$

$$\frac{dR}{dt} = \mu I, \tag{2.2c}$$

where β is the infection rate and μ the recovery rate.
In Figure 2.3 one can observe the graphical solution for the aforementioned set of equations.
We can define a central parameter that is used to ensure if the disease will spread or die out; this is the **reproduction number** $R_0 = \frac{\beta}{\mu}$. When $R_0 > 1$ the disease will spread, otherwise it will die out.
Some assumptions lead this model:

- closed, constant-size population, that ensures that at each time the sum of all compartments is the same $\rightarrow S(t) + I(t) + R(t) = N$,
- homogeneous mixing,
- constant parameters over time,
- permanent immunity after recovery.

Despite the simplicity of this mathematical model, it has been widely used since it is analytically tractable and useful in analyzing threshold phenomena like **herd immunity** [19], providing the foundation for more complex approaches, obtained, for example, by augmenting the number of compartments, introducing stochasticity and using a metapopulation-based approach in place of the individual-based one.

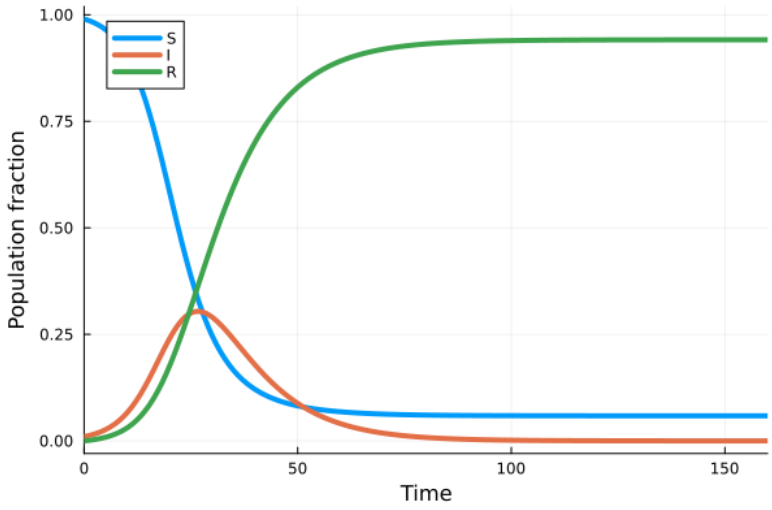

Figure 2.3: Example of a homogeneous normalized SIR model: solution of the deterministic system of equations (2.2a)–(2.2c). The parameter used are $\beta = 0.3, \mu = 0.1$, with initial conditions $S(0) = 0.99, I(0) = 0.01, R(0) = 0.0$. Blue line represents the fraction of susceptible individuals, red line the fraction of infected individuals, and green line the recovered ones.

2.2.2 Extension to the SEIR model

While the SIR model describes the essential dynamics of many infectious diseases, it is limited by the fact that susceptible individuals become infected right after contact with an infectious individual, which is biologically quite unrealistic for most pathogens that have a **latent period**.
To overcome this limitation, the **SEIR model** add a compartment for **exposed but not yet infectious individuals**, $E(t)$.
This model assumes relevance especially when treating diseases like measles, **Ebola**, and most recent **COVID-19**, where the incubation time plays a crucial role in transmission dynamics [20, 21].
The SEIR model is divided in the following compartments:

- $S(t)$: susceptibles,

- $E(t)$: exposed (infected but not yet infectious),
- $I(t)$: infectious,
- $R(t)$: recovered or removed.

The differential equations governing the model are:

$$\frac{dS}{dt} = -\beta \frac{SI}{N}, \tag{2.3a}$$
$$\frac{dE}{dt} = \beta \frac{SI}{N} - \eta E, \tag{2.3b}$$
$$\frac{dI}{dt} = \eta E - \mu I, \tag{2.3c}$$
$$\frac{dR}{dt} = \mu I, \tag{2.3d}$$

where β is the exposure rate for the transition $S \to E$, η is the infection rate for the transition $E \to I$ ($\frac{1}{\eta}$ is the average incubation time), and μ is the recovery time for the transition $I \to R$. The solution of this system of equations is provided in Figure 2.4.
The homogeneous mixing assumption still holds, along with the definition of the reproduction number $R_0 = \frac{\beta}{\mu}$ under standard assumptions [18]. The SEIR framework is more accurate than the SIR model and is widely used both for deterministic and stochastic modeling studies [22], although it is more complicated to implement. This will be the central model implemented in this thesis.

2.2.3 The stochastic SEIR metapopulation model

While deterministic compartmental models like SEIR offer a relatively simple framework to understand average epidemic behavior, they are not able to take into account the randomness and spatial heterogeneity typical of real-world disease spread.
In this sense, a better approach would be that of **stochastic SEIR models**. A particularly powerful framework is the **metapopulation approach**, which is characterized by the division of the global population in several subpopulations that in the framework of this thesis are provinces or departments. This approach is quite different from **individual-based models**, since each subpopulation has its own disease spread, and in addition we have **mobility** between different regions, which contributes to the dynamics of involved subpopulations.
This is especially useful when modeling infectious disease spread across cities, regions, or countries, as shown, for instance, in the work of Colizza, Pastor-Satorras,

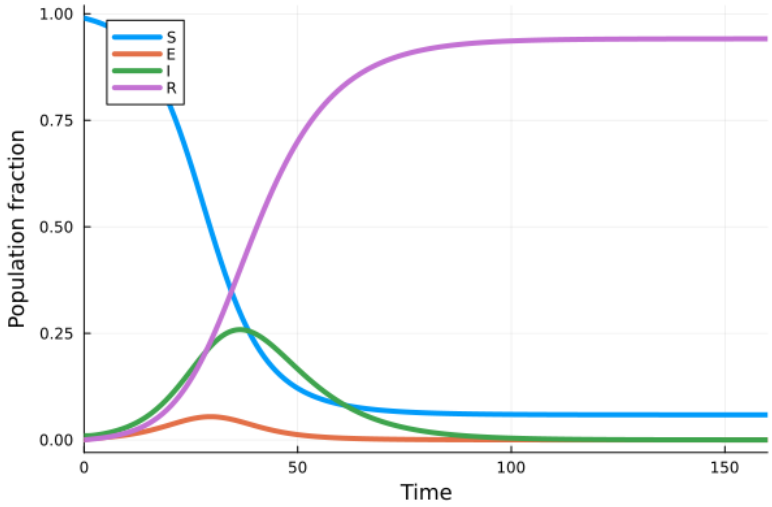

Figure 2.4: Example of a homogeneous normalized SEIR model: solution of the deterministic system of equations (2.3a)–(2.3d). The parameter used are $\beta = 0.3, \eta = 0.6, \mu = 0.1$, with initial conditions $S(0) = 0.99, E(0) = 0.0, I(0) = 0.01, R(0) = 0.0$. Blue line represents the fraction of susceptible individuals, red line the fraction of exposed individuals, green line the infected individuals, and pink line the recovered ones.

and Vespignani [17], and that of Belik, Geisel, and Brockmann [23], in which the reaction-diffusion approach for mobility is used.
In this thesis, the model does not account for actual movements of individuals between different geographic regions (there are no transport terms), but interregion contagions can occur because of colocation effects, taken into account by the colocation matrix W mentioned in Section 2.1 that represents the rate at which people from the geographic regions i and j are colocated (in any other place over a given interval of time).
We adopt a time discretization at daily level, i.e. $\Delta t = 1day$. The epidemic dynamics in region i is described by the number S_i of susceptible individuals, E_i of exposed individuals, I_i of infected individuals and R_i of recovered individuals, with the constraint $S_i + E_i + I_i + R_i = N_i$ where N_i is the total population of subpopulation i.
The total number of different states in the subpopulations evolves as follows:

$$S_i(t+1) = S_i(t) - \Delta E_i, \tag{2.4}$$
$$E_i(t+1) = E_i(t) + \Delta E_i - \Delta I_i \tag{2.5}$$
$$I_i(t+1) = I_i(t) + \Delta I_i - \Delta R_i, \tag{2.6}$$
$$R_i(t+1) = R_i(t) + \Delta R_i, \tag{2.7}$$

with daily variations between day t and day $t+1$ given by

$$\Delta E_i \sim \text{Binomial}(S_i(t), p_{\beta,i}(t)) = \begin{pmatrix} S_i(t) \\ \Delta E_i \end{pmatrix} (p_{\beta,i}(t))^{\Delta E_i} (1 - p_{\beta,i}(t))^{S_i(t)-\Delta E_i}, \tag{2.8}$$

$$\Delta I_i \sim \text{Binomial}(E_i(t), p_{\eta,i}(t)) = \begin{pmatrix} E_i(t) \\ \Delta I_i \end{pmatrix} (p_{\eta}(t))^{\Delta I_i} (1 - p_{\eta}(t))^{E_i(t)-\Delta I_i}, \tag{2.9}$$

$$\Delta R_i \sim \text{Binomial}(I_i(t), p_{\mu}(t)) = \begin{pmatrix} I_i(t) \\ \Delta R_i \end{pmatrix} (p_{\mu}(t))^{\Delta R_i} (1 - p_{\mu}(t))^{I_i(t)-\Delta R_i}, \tag{2.10}$$

where

$$p_{\beta,i}(t) = 1 - e^{-\beta \Delta t \sum_j w_{ij} I_j(t)}, \tag{2.11}$$

$$p_{\eta} = 1 - e^{-\eta \Delta t} \tag{2.12}$$

$$p_{\mu} = 1 - e^{-\mu \Delta t}, \tag{2.13}$$

which represent respectively the probability of exposure, infection and recovery at each time-step, derived as 1− **(the probability that no exposure, infection or recovery event occur in that time-step)**.
In Figure 2.5 one can see a single Monte Carlo simulation of the model described above, representing the curves of infected individuals for each subpopulation.

2.3 Bayesian Inference

Bayesian inference is a statistical framework that allows to integrate prior knowledge with observed data in order to learn information about unknown parameters of a model.
In the context of epidemic modeling, this approach can be used to infer parameters (e.g., transmission rate, recovery rate) or latent variables (e.g., unobserved states of populations) from incomplete data.

2.3.1 Bayes' theorem

Bayesian inference is based on the Bayes' theorem, which relates the posterior distribution $P(\theta|x)$ of parameters θ (also called **stochastic "machine"**) given data x to the prior distribution $P(\theta)$ and the likelihood (or stochastic rule) $P(x|\theta)$. With these ingredients, we will have a double stochastic process:

- θ is sampled from $P(\theta)$

- x is sampled from $P(x|\theta)$.

This relation states:

$$P(\theta|x) = \frac{P(x|\theta)P(\theta)}{P(x)}. \tag{2.14}$$

In this context the data x are the time-scattered observations which we refer to as $\mathcal{O}$, namely we assume to receive daily information about the aggregate number of infected individuals at the single population level, while parameters θ are represented by the overall history of the metapopulation system $\mathcal{H}$:

$$\begin{aligned} P(\mathcal{H}|\mathcal{O}) &= \frac{P(\mathcal{O}|\mathcal{H})\,P(\mathcal{H})}{P(\mathcal{O})} \\ &\propto P(\mathcal{O}|\mathcal{H})\,P(\mathcal{H}). \end{aligned} \tag{2.15}$$

2.3.2 Bayesian inference in epidemics

Epidemic systems are intrinsically stochastic and often subject to problems related to the correct counting of cases, the time of observation, and incomplete information. In this sense, Bayesian inference plays a central role since it allows one to:

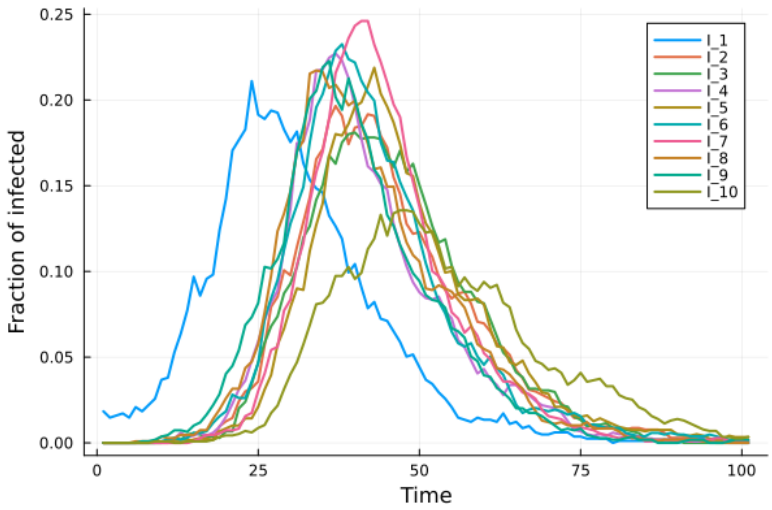

Figure 2.5: Monte Carlo simulation of a normalized stochastic SEIR metapopulation model: each curve represents the fraction of infected individuals at each time for each subpopulation. The network graph is obtained using an Erdös-Renýi random graph $G(n,p)$ with $n = 10, p = \frac{K}{n}$, and $K = 2$ is the average degree of the graph. The parameters used are $\beta = 0.3, \eta = 0.1, \mu = 0.2$, with epidemic starting in node 1.

- Incorporate external information through priors,
- Quantify uncertainty in parameters and predictions,
- Update predictions in real time using newly available data,
- Infer hidden variables, such as unreported cases.

This method has been widely used in history, for example, Bettencourt and Ribeiro (2008) [9] used this framework to estimate the reproduction number of an emerging outbreak in real time for containment measures purposes.
Similarly, Flaxman et al. (2020) [10] employed hierarchical Bayesian models to assess the effect of non-pharmaceuticals interventions across multiple countries during the COVID-19 pandemic.

2.3.3 The prior distribution and the likelihood of the data

In the following we derive (approximate) expressions for the two ingredients of the Bayes' formula: the prior $P(\mathcal{H})$, which is the probabilistic model associated with the stochastic SEIR metapopulation system, and the likelihood of the data $P(\mathcal{O}|\mathcal{H})$, based on a simple gaussian sampling process for the observations.
From Section 2.2.3 we have seen that the sum of individuals in each subpopulation is always equal to N_i; this allows to consider only three equations. The overall probability of a variation $(\Delta E_i, \Delta I_i, \Delta R_i)$ in subpopulation i is

$$\begin{aligned} P_i^{t,t+1}\left(\Delta E_i, \Delta I_i, \Delta R_i | \left\{S_i(t), E_i(t), I_i(t)\right\}_{i\in V}\right) &= \\ = \begin{pmatrix} S_i(t) \\ \Delta E_i \end{pmatrix} (p_{\beta,i}(t))^{\Delta E_i} (1 - p_{\beta,i}(t))^{S_i(t)-\Delta E_i} & \\ \times \begin{pmatrix} E_i(t) \\ \Delta I_i \end{pmatrix} (p_\eta(t))^{\Delta I_i} (1 - p_\eta(t))^{E_i(t)-\Delta I_i} & \\ \times \begin{pmatrix} I_i(t) \\ \Delta R_i \end{pmatrix} (p_\mu(t))^{\Delta R_i} (1 - p_\mu(t))^{I_i(t)-\Delta R_i} & \end{aligned} \tag{2.16}$$

The corresponding characteristic function is given by the Fourier transform,

$$
\begin{aligned}
\hat{P}_i^{t,t+1}\left(\vec{X}_i \mid \left\{\vec{S}_i\left(t\right)\right\}_{i\in V}\right) &= \sum_{\vec{\Delta S}_i=0}^{\vec{S}_i(t)} P_i^{t,t+1}\left(\vec{\Delta S}_i \mid \left\{\vec{S}_i\left(t\right)\right\}_{i\in V}\right) e^{-i\vec{X}_i\cdot\vec{\Delta S}_i} \\
&= \sum_{\Delta E_i=0}^{S_i(t)}\sum_{\Delta I_i=0}^{E_i(t)}\sum_{\Delta R_i=0}^{I_i(t)} \binom{S_i(t)}{\Delta E_i} p_{\beta,i}(t)^{\Delta E_i}(1-p_{\beta,i}(t))^{S_i(t)-\Delta E_i} \\
&\quad \times \binom{E_i(t)}{\Delta I_i} p_\eta^{\Delta I_i}(1-p_\eta)^{E_i(t)-\Delta I_i} \binom{I_i(t)}{\Delta R_i} p_\mu^{\Delta R_i}(1-p_\mu)^{I_i(t)-\Delta R_i} \\
&\quad \times e^{-iX_i\Delta E_i - iY_i\Delta I_i - iZ_i\Delta R_i} \\
&= \left(1-p_{\beta,i}(t)(1-e^{-iX_i})\right)^{S_i(t)} \left(1-p_\eta(1-e^{-iY_i})\right)^{E_i(t)} \\
&\quad \times \left(1-p_\mu(1-e^{-iZ_i})\right)^{I_i(t)} \\
&= \exp\left[S_i(t)\log\left(1-p_{\beta,i}(t)(1-e^{-iX_i})\right)\right] \\
&\quad \times \exp\left[E_i(t)\log\left(1-p_\eta(1-e^{-iY_i})\right)\right] \\
&\quad \times \exp\left[I_i(t)\log\left(1-p_\mu(1-e^{-iZ_i})\right)\right] \qquad (2.17)
\end{aligned}
$$

where $\vec{X}_i = [X_i, Y_i, Z_i]$, $\vec{S}_i(t) = [S_i(t), E_i(t), I_i(t)]$ and $\vec{\Delta S}_i = [\Delta E_i, \Delta I_i, \Delta R_i]$.
It is convenient to consider the anti-transform of the same quantity,

$$
\begin{aligned}
P_i^{t,t+1}\left(\vec{\Delta S}_i \middle| \left\{\vec{S}_i\left(t\right)\right\}_{i\in V}\right) &= \int_{-\infty}^{+\infty} \frac{d\vec{X}_i}{(2\pi)^3} e^{\mathrm{i}\vec{X}_i\cdot\Delta\vec{S}_i} \hat{P}_i^{t,t+1}\left(\vec{X}_i \middle| \left\{\vec{S}_i\left(t\right)\right\}_{i\in V}\right) \\
&= \int_{-\infty}^{+\infty}\frac{dX_i}{2\pi}\int_{-\infty}^{+\infty}\frac{dY_i}{2\pi}\int_{-\infty}^{+\infty}\frac{dZ_i}{2\pi} e^{\mathrm{i}X_i\Delta E_i + \mathrm{i}Y_i\Delta I_i + \mathrm{i}Z_i\Delta R_i} \\
&\quad \times \exp\left[S_i\left(t\right)\log\left(1-p_{\beta,i}\left(t\right)\left(1-e^{-\mathrm{i}X_i}\right)\right)\right] \\
&\quad \times \exp\left[E_i\left(t\right)\log\left(1-p_\eta\left(1-e^{-\mathrm{i}Y_i}\right)\right)\right] \\
&\quad \times \exp\left[I_i\left(t\right)\log\left(1-p_\mu\left(1-e^{-\mathrm{i}Z_i}\right)\right)\right] \qquad (2.18)
\end{aligned}
$$

and express the increments in terms of the state variables,

$$
\begin{aligned}
\Delta R_i &= R_i\left(t+1\right) - R_i\left(t\right) && (2.19)\\
\Delta I_i &= I_i\left(t+1\right) + R_i\left(t+1\right) - I_i\left(t\right) - R_i\left(t\right) && (2.20)\\
\Delta E_i &= E_i\left(t+1\right) + I_i\left(t+1\right) + R_i\left(t+1\right) - E_i\left(t\right) - I_i\left(t\right) - R_i\left(t\right) && (2.21)
\end{aligned}
$$

in order to eliminate the increments in favor of the state variables. We obtain the

transition probability

$$
\begin{aligned}
& P_i^{t,t+1}\left(E_i\left(t+1\right), I_i\left(t+1\right), R_i\left(t+1\right) \mid \left\{E_i\left(t\right), I_i\left(t\right), R_i\left(t\right)\right\}_{i\in V}\right) = \\
= & \int_{-\infty}^{+\infty}\frac{dX_i}{2\pi}\int_{-\infty}^{+\infty}\frac{dY_i}{2\pi}\int_{-\infty}^{+\infty}\frac{dZ_i}{2\pi}e^{\mathrm{i}X_i(E_i(t+1)+I_i(t+1)+R_i(t+1))} \\
& \times e^{\mathrm{i}X_i(-E_i(t)-I_i(t)-R_i(t))+\mathrm{i}Y_i(I_i(t+1)+R_i(t+1)-I_i(t)-R_i(t))} \\
& \times e^{\mathrm{i}Z_i(R_i(t+1)-R_i(t))+S_i(t)\log\left(1-p_{\beta,i}(t)\left(1-e^{-\mathrm{i}X_i}\right)\right)} \\
& \times e^{E_i(t)\log\left(1-p_\eta\left(1-e^{-\mathrm{i}Y_i}\right)\right)+I_i(t)\log\left(1-p_\mu\left(1-e^{-\mathrm{i}Z_i}\right)\right)} \\
= & \int_{-\infty}^{+\infty}\frac{dX_i}{2\pi}\int_{-\infty}^{+\infty}\frac{dY_i}{2\pi}\int_{-\infty}^{+\infty}\frac{dZ_i}{2\pi}e^{\mathrm{i}X_i(E_i(t+1)-E_i(t))} \\
& \times e^{\mathrm{i}(X_i+Y_i)(I_i(t+1)-I_i(t))+\mathrm{i}(X_i+Y_i+Z_i)(R_i(t+1)-R_i(t))} \\
& \times e^{(N_i-E_i(t)-I_i(t)-R_i(t))\log\left(1-p_{\beta,i}(t)\left(1-e^{-\mathrm{i}X_i}\right)\right)} \\
& \times e^{E_i(t)\log\left(1-p_\eta\left(1-e^{-\mathrm{i}Y_i}\right)\right)+I_i(t)\log\left(1-p_\mu\left(1-e^{-\mathrm{i}Z_i}\right)\right)}. \qquad (2.22)
\end{aligned}
$$

This expression will be further analyzed through **Path-Integral formulation** in the next chapter.
For the likelihood of the data we assume that observations are obtained by means of an independent sampling process based on gaussian sampling with standard deviation σ,

$$
P\left(\mathcal{O}|\left\{\vec{E}_i, \vec{I}_i, \vec{R}_i\right\}_{i\in V}\right) = \prod_{o=1}^{N_{\mathrm{obs}}} P\left(I_o|I_{i_o}\left(t_o\right)\right) \qquad (2.23)
$$

with

$$
P\left(I_o|I_{i_o}\left(t_o\right)\right) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{\left(I_{i_o}(t_o)-I_o\right)^2}{2\sigma^2}}. \qquad (2.24)
$$

Chapter 3

Path Integral Formulation

3.1 Brief review on path-integral formulation

In this section we introduce the foundation of the path integral formulation for statistical physics and stochastic processes. Also, we briefly explain how it is used to represent the probability of a trajectory, reconnecting to previous chapters where we described the SEIR dynamics.
It is frequently essential to record the entire course of the system's evolution when studying stochastic dynamical systems, particularly those that simulate the spread of infectious diseases. The path integral approach, first developed by Richard Feynman [24] for single-particle quantum mechanics, was then adopted as a central and powerful formalism in statistical mechanics, field theory, and stochastic processes. In particular, in the latter case, methods based on functional integration, such as the *Response Functional Formalism* introduced by Martin, Siggia and Rose (MSR) [25], were shown to provide a compact and theoretical insightful description for stochastic processes involving many degrees of freedom.
This method leads to the construction of a dynamic action starting from a phenomenological Langevin equation for the field. The reason for the denomination as "response functional" formalism stays in the introduction of a so-called response field when using the integral representation of the Dirac delta function [26].
By integrating over all potential paths the system could take, each weighted by an exponential factor involving an action functional, the path integral formulation essentially represents the probability of a trajectory without the need to specifically realize the stochastic process. This method is perfect for analyzing epidemic models in which transmission variability and demographic noise are not negligible because it naturally accounts for randomness and fluctuations.
In the context of epidemiology, this technique allows us to recover a statistical field theory of the stochastic SEIR model. Every possible path an epidemic might

follow, representing the progression over time of susceptible, exposed, infected, and recovered individuals, affects the system's behavior according to how likely it is. This likelihood is quantified using an exponential function involving a mathematical construction known as the **action**, which captures both the underlying deterministic behavior and the inherent randomness present in the system.
From the mathematical point of view, this formulation leads to express the (dynamical) **partition function** of the system as follows:

$$\mathcal{Z} = \int \mathcal{D}\left[\phi\right] e^{-\mathcal{S}[\phi]} \tag{3.1}$$

where ϕ will represent the dependencies of the action $\mathcal{S}$.
As well as the application of the saddle point to the equilibrium partition function results in the realization of the ground state of the system, here the saddle-point condition gives the most probable trajectory, that under natural conditions would correspond to the deterministic description provided by a mean-field approximation, neglecting stochasticity and correlations between variables. This formalism provides a compact but complete representation of the behavior of the system across all time steps and the whole set of degrees of freedom (e.g. subpopulations), in this respect offering a good compromise between classical probabilistic models that work with transition matrices and oversimplified deterministic models described by ordinary differential equations. For this reason, the path integral approach seems very suitable for tackling challenging inference problems, where the aim is to reconstruct the trajectories of variables given some information or observations.
In the following, we derive the partition function of the system starting from Equation (2.22) and then derive the corresponding action $\mathcal{S}$. We will then show that the mean-field approximation corresponds to a set of equations that are linear in the conjugate variables.

3.2 Partition function and Action of the system

We start from the expression for the transition probability in Equation (2.22):

$$\begin{aligned}
& P_i^{t,t+1}\left(E_i\left(t+1\right), I_i\left(t+1\right), R_i\left(t+1\right) \mid \left\{E_i\left(t\right), I_i\left(t\right), R_i\left(t\right)\right\}_{i\in V}\right) = \\
= \; & \int_{-\infty}^{+\infty} \frac{dX_i}{2\pi} \int_{-\infty}^{+\infty} \frac{dY_i}{2\pi} \int_{-\infty}^{+\infty} \frac{dZ_i}{2\pi} e^{\mathrm{i}X_i\left(E_i(t+1)-E_i(t)\right)} \\
& \times e^{\mathrm{i}(X_i+Y_i)\left(I_i(t+1)-I_i(t)\right)+\mathrm{i}(X_i+Y_i+Z_i)\left(R_i(t+1)-R_i(t)\right)} \\
& \times e^{\left(N_i-E_i(t)-I_i(t)-R_i(t)\right)\log\left(1-p_{\beta,i}(t)\left(1-e^{-\mathrm{i}X_i}\right)\right)} \\
& \times e^{E_i(t)\log\left(1-p_\eta\left(1-e^{-\mathrm{i}Y_i}\right)\right)+I_i(t)\log\left(1-p_\mu\left(1-e^{-\mathrm{i}Z_i}\right)\right)}.
\end{aligned}$$

The overall process can be described by a path integral obtained by multiplying all single-region single-time transition probabilities over the whole set of nodes and

over all time steps and integrating over all possible realizations of the numbers of exposed, infected and recovered individuals in the regions. Such dynamical partition function is given by:

$$\begin{aligned}\mathcal{Z} &= \sum_{\left\{\vec{E}_i,\vec{I}_i,\vec{R}_i\right\}_{i\in V}} \prod_{i\in V} \Big\{ P_i^0\left(E_i\left(0\right),I_i\left(0\right),R_i\left(0\right)\right) \\ &\quad \times \prod_{t=0}^{T-1} P_i^{t,t+1}\left(E_i\left(t+1\right),I_i\left(t+1\right),R_i\left(t+1\right)\mid\left\{E_i\left(t\right),I_i\left(t\right),R_i\left(t\right)\right\}_{i\in V}\right)\Big\} \\ &= \sum_{\left\{\vec{E}_i,\vec{I}_i,\vec{R}_i\right\}_{i\in V}} \int_{-\infty}^{+\infty} \mathcal{D}\vec{X}\mathcal{D}\vec{Y}\mathcal{D}\vec{Z}e^{-\mathcal{S}\left[\vec{X},\vec{Y},\vec{Z},\vec{E},\vec{I},\vec{R}\right]}, \end{aligned} \tag{3.2}$$

where $\vec{E}_i, \vec{I}_i, \vec{R}_i$ represent the trajectories over all time steps respectively for exposed, infected, and removed individuals, and $P_i^0\left(E_i\left(0\right),I_i\left(0\right),R_i\left(0\right)\right)$ is the probability distribution of the system at time $t=0$.
Also, one can see the introduction, in the expression of the partition function $\mathcal{Z}$, of the action $\mathcal{S}$, which can be expressed as follows:

$$\begin{aligned}\mathcal{S}\left[\vec{X},\vec{Y},\vec{Z},\vec{E},\vec{I},\vec{R}\right] &= \sum_{i=1}^{|V|}\sum_{t=0}^{T-1}\{-\mathrm{i}X_i\left(t+1\right)\left(E_i\left(t+1\right)-E_i\left(t\right)\right) \\ &\quad -\left(\mathrm{i}X_i\left(t+1\right)+\mathrm{i}Y_i\left(t+1\right)\right)\left(I_i\left(t+1\right)-I_i\left(t\right)\right) \\ &\quad -\left(\mathrm{i}X_i\left(t+1\right)+\mathrm{i}Y_i\left(t+1\right)+\mathrm{i}Z_i\left(t+1\right)\right)\left(R_i\left(t+1\right)-R_i\left(t\right)\right) \\ &\quad -\left(N_i-E_i\left(t\right)-I_i\left(t\right)-R_i\left(t\right)\right) \\ &\quad \times\log\left(1-\left(1-e^{-\tilde{\beta}\sum_j w_{ij}I_j(t)}\right)\left(1-e^{-\mathrm{i}X_i(t+1)}\right)\right) \\ &\quad -E_i\left(t\right)\log\left(1-\left(1-e^{-\tilde{\eta}}\right)\left(1-e^{-\mathrm{i}Y_i(t+1)}\right)\right) \\ &\quad -I_i\left(t\right)\log\left(\left(1-e^{-\tilde{\mu}}\right)\left(1-e^{-\mathrm{i}Z_i(t+1)}\right)\right)\Big\} \\ &\quad -\sum_{i=1}^{|V|}\log P_i^0\left(E_i\left(0\right),I_i\left(0\right),R_i\left(0\right)\right). \end{aligned} \tag{3.3}$$

The initial conditions are usually assumed to be concentrated in one or few subpopulations and to consists of a small (but not negligible) number of infected individuals. A possible parametrization is that of using a common initial number, say $I_0 \sim 10 \div 10^2$ with a parameter q_i as follows

$$
\begin{aligned}
& P_i^0\left(E_i\left(0\right),I_i\left(0\right),R_i\left(0\right)\right)=\delta_{E_i(0),0}\delta_{I_i(0),q_iI_0}\delta_{R_i(0),0}= \\
= & \int_{-\infty}^{+\infty}dX_i\left(0\right)dY_i\left(0\right)dZ_i\left(0\right)e^{\mathrm{i}X_i(0)E_i(0)+\mathrm{i}Y_i(0)(I_i(0)-q_iI_0)+\mathrm{i}Z_i(0)R_i(0)} \\
= & \int_{-\infty}^{+\infty}dX_i\left(0\right)dY_i\left(0\right)dZ_i\left(0\right)e^{\mathrm{i}X_i(0)(E_i(0))+(\mathrm{i}X_i(0)+\mathrm{i}Y_i(0))(I_i(0)-q_iI_0)} \\
& \times e^{(\mathrm{i}X_i(0)+\mathrm{i}Y_i(0)+\mathrm{i}Z_i(0))R_i(0)}
\end{aligned}
\tag{3.4}
$$

where in the last expression a change of variable was done to provide a similar form to what already found in the main part of the action.
The parameter q_i represents the probability that node i is the infection seed. Since it is strictly in the interval [0,1], instead of introducing a prior on $\{q_i\}_{i\in V}$, we could introduce a regularization term which takes the form of an neg-entropy function, i.e.

$$
S\left[\{q_i\}_{i\in V}\right]=\epsilon\sum_{i=1}^{N}\left(q_i\log q_i+\left(1-q_i\right)\log\left(1-q_i\right)\right). \tag{3.5}
$$

This term favors polarization of the parameters q_i. In addition, we can insert a penalty $\delta\sum_{i=1}^{N}q_i$ to favor a small number of non-zero initial conditions.
According to this derivation, we derive the general form of the prior model with $\mathcal{H}=\left\{\vec{E}_i,\vec{I}_i,\vec{R}_i\right\}_{i\in V}$ as:

$$
P\left(\left\{\vec{E}_i,\vec{I}_i,\vec{R}_i\right\}_{i\in V}\right)\ \propto\ \int\mathcal{D}\vec{X}\mathcal{D}\vec{Y}\mathcal{D}\vec{Z}e^{-\mathcal{S}\left[\vec{X},\vec{Y},\vec{Z},\vec{E},\vec{I},\vec{R}\right]}, \tag{3.6}
$$

and by consequence we find:

$$
\begin{aligned}
\mathcal{S}\left[\vec{X},\vec{Y},\vec{Z},\vec{E},\vec{I},\vec{R}\right] = & \sum_{i=1}^{|V|}\sum_{t=0}^{T-1}\{-\mathrm{i}X_i\left(t+1\right)\left(E_i\left(t+1\right)-E_i\left(t\right)\right) \\
& -\left(\mathrm{i}X_i\left(t+1\right)+\mathrm{i}Y_i\left(t+1\right)\right)\left(I_i\left(t+1\right)-I_i\left(t\right)\right) \\
& -\left(\mathrm{i}X_i\left(t+1\right)+\mathrm{i}Y_i\left(t+1\right)+\mathrm{i}Z_i\left(t+1\right)\right)\left(R_i\left(t+1\right)-R_i\left(t\right)\right) \\
& -\left(N_i-E_i\left(t\right)-I_i\left(t\right)-R_i\left(t\right)\right) \\
& \times\log\left(1-\left(1-e^{-\tilde{\beta}\sum_j w_{ij}I_j(t)}\right)\left(1-e^{-\mathrm{i}X_i(t+1)}\right)\right) \\
& -E_i\left(t\right)\log\left(1-\left(1-e^{-\tilde{\eta}}\right)\left(1-e^{-\mathrm{i}Y_i(t+1)}\right)\right) \\
& -I_i\left(t\right)\log\left(1-\left(1-e^{-\tilde{\mu}}\right)\left(1-e^{-\mathrm{i}Z_i(t+1)}\right)\right)\} \\
& -\sum_{i=1}^{|V|}\{\mathrm{i}X_i\left(0\right)E_i\left(0\right)+\left(\mathrm{i}X_i\left(0\right)+\mathrm{i}Y_i\left(0\right)\right)\left(I_i\left(0\right)-q_iI_0\right) \\
& +\left(\mathrm{i}X_i\left(0\right)+\mathrm{i}Y_i\left(0\right)+\mathrm{i}Z_i\left(0\right)\right)R_i\left(0\right)\} \\
& -\epsilon\sum_{i=1}^{|V|}\left(q_i\log q_i+\left(1-q_i\right)\log\left(1-q_i\right)\right)-\delta\sum_{i=1}^{|V|}q_i.
\end{aligned}
\tag{3.7}
$$

At this point, we recall the likelihood of the data that we mentioned in Section 2.3.3, and refer to Equation (2.24) in such a way to obtain the posterior probability:

$$
\begin{aligned}
& P\left(\left\{\vec{E}_i, \vec{I}_i, \vec{R}_i\right\}_{i\in V} | \mathcal{O}\right) \propto \int \mathcal{D}\vec{X}\mathcal{D}\vec{Y}\mathcal{D}\vec{Z} e^{-\mathcal{S}\left[\vec{X},\vec{Y},\vec{Z},\vec{E},\vec{I},\vec{R}\right]+\sum_{o\in\mathcal{O}}\log P(I_o|I_{i_o}(t_o))} = \\
= & \sum_{i=1}^{|V|}\sum_{t=0}^{T-1}\{-\mathrm{i}X_i\left(t+1\right)\left(E_i\left(t+1\right)-E_i\left(t\right)\right) \\
& -\left(\mathrm{i}X_i\left(t+1\right)+\mathrm{i}Y_i\left(t+1\right)\right)\left(I_i\left(t+1\right)-I_i\left(t\right)\right) \\
& -\left(\mathrm{i}X_i\left(t+1\right)+\mathrm{i}Y_i\left(t+1\right)+\mathrm{i}Z_i\left(t+1\right)\right)\left(R_i\left(t+1\right)-R_i\left(t\right)\right) \\
& -\left(N_i-E_i\left(t\right)-I_i\left(t\right)-R_i\left(t\right)\right)\log\left(1-\left(1-e^{-\tilde{\beta}\sum_j w_{ij}I_j(t)}\right)\left(1-e^{-\mathrm{i}X_i(t+1)}\right)\right) \\
& -E_i\left(t\right)\log\left(1-\left(1-e^{-\tilde{\eta}}\right)\left(1-e^{-\mathrm{i}Y_i(t+1)}\right)\right) \\
& -I_i\left(t\right)\log\left(1-\left(1-e^{-\tilde{\mu}}\right)\left(1-e^{-\mathrm{i}Z_i(t+1)}\right)\right)\} \\
& -\sum_{i=1}^{|V|}\{\mathrm{i}X_i\left(0\right)E_i\left(0\right)+\left(\mathrm{i}X_i\left(0\right)+\mathrm{i}Y_i\left(0\right)\right)\left(I_i\left(0\right)-q_iI_0\right) \\
& +\left(\mathrm{i}X_i\left(0\right)+\mathrm{i}Y_i\left(0\right)+\mathrm{i}Z_i\left(0\right)\right)R_i\left(0\right)\} - \epsilon\sum_{i=1}^{|V|}\left(q_i\log q_i+\left(1-q_i\right)\log\left(1-q_i\right)\right) \\
& -\delta\sum_{i=1}^{|V|}q_i - \sum_{o\in\mathcal{O}}\log\left[\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{\left(I_{i_o}(t_o)-I_o\right)^2}{2\sigma^2}}\right]. \qquad (3.8)
\end{aligned}
$$

The overall action becomes

$$\begin{aligned}
\mathcal{S}^{\text{tot}} &= \sum_{i=1}^{|V|}\sum_{t=0}^{T-1}\{-\mathrm{i}X_i(t+1)\left(E_i(t+1)-E_i(t)\right) \\
&\quad -\left(\mathrm{i}X_i(t+1)+\mathrm{i}Y_i(t+1)\right)\left(I_i(t+1)-I_i(t)\right) \\
&\quad -\left(\mathrm{i}X_i(t+1)+\mathrm{i}Y_i(t+1)+\mathrm{i}Z_i(t+1)\right)\left(R_i(t+1)-R_i(t)\right) \\
&\quad -\left(N_i-E_i(t)-I_i(t)-R_i(t)\right)\log\left(1-\left(1-e^{-\tilde{\beta}\sum_j w_{ij}I_j(t)}\right)\left(1-e^{-\mathrm{i}X_i(t+1)}\right)\right) \\
&\quad -E_i(t)\log\left(1-\left(1-e^{-\tilde{\eta}}\right)\left(1-e^{-\mathrm{i}Y_i(t+1)}\right)\right) \\
&\quad -I_i(t)\log\left(1-\left(1-e^{-\tilde{\mu}}\right)\left(1-e^{-\mathrm{i}Z_i(t+1)}\right)\right)\} \\
&\quad -\sum_{i=1}^{|V|}\{\mathrm{i}X_i(0)E_i(0)+\left(\mathrm{i}X_i(0)+\mathrm{i}Y_i(0)\right)\left(I_i(0)-q_iI_0\right) \\
&\quad +\left(\mathrm{i}X_i(0)+\mathrm{i}Y_i(0)+\mathrm{i}Z_i(0)\right)R_i(0)\}-\epsilon\sum_{i=1}^{|V|}\left(q_i\log q_i+(1-q_i)\log(1-q_i)\right) \\
&\quad -\delta\sum_{i=1}^{|V|}q_i-\sum_{o\in\mathcal{O}}\left[\log\left(\frac{1}{\sigma\sqrt{2\pi}}\right)-\frac{\left(I_{i_o}(t_o)-I_o\right)^2}{2\sigma^2}\right].
\end{aligned} \tag{3.9}$$

It is now convenient to define new variables which will be, along with the state variables E_i, I_i, R_i, the central point of the inference algorithm; we define $\theta_i^E(t) = \mathrm{i}X_i(t)$, $\theta_i^I(t) = \mathrm{i}X_i(t)+\mathrm{i}Y_i(t)$ and $\theta_i^R(t) = \mathrm{i}X_i(t)+\mathrm{i}Y_i(t)+\mathrm{i}Z_i(t)$, so that the action becomes:

$$\begin{aligned}
\mathcal{S}^{\text{tot}} &= \sum_{i=1}^{|V|}\sum_{t=0}^{T-1}\left\{-\theta_i^E(t+1)\left(E_i(t+1)-E_i(t)\right)-\theta_i^I(t+1)\left(I_i(t+1)-I_i(t)\right)\right. \\
&\quad -\theta_i^R(t+1)\left(R_i(t+1)-R_i(t)\right) \\
&\quad -\left(N_i-E_i(t)-I_i(t)-R_i(t)\right)\log\left(1-\left(1-e^{-\tilde{\beta}\sum_j w_{ij}I_j(t)}\right)\left(1-e^{-\theta_i^E(t+1)}\right)\right) \\
&\quad -E_i(t)\log\left(1-\left(1-e^{-\tilde{\eta}}\right)\left(1-e^{-\left(\theta_i^I(t+1)-\theta_i^E(t+1)\right)}\right)\right) \\
&\quad \left.-I_i(t)\log\left(1-\left(1-e^{-\tilde{\mu}}\right)\left(1-e^{-\left(\theta_i^R(t+1)-\theta_i^I(t+1)\right)}\right)\right)\right\} \\
&\quad -\sum_{i=1}^{|V|}\left\{\theta_i^E(0)E_i(0)+\theta_i^I(0)\left(I_i(0)-q_iI_0\right)+\theta_i^R(0)R_i(0)\right\} \\
&\quad -\epsilon\sum_{i=1}^{|V|}\left(q_i\log q_i+(1-q_i)\log(1-q_i)\right) \\
&\quad -\delta\sum_{i=1}^{|V|}q_i-\sum_{o\in\mathcal{O}}\left[\log\left(\frac{1}{\sigma\sqrt{2\pi}}\right)-\frac{\left(I_{i_o}(t_o)-I_o\right)^2}{2\sigma^2}\right].
\end{aligned} \tag{3.10}$$

3.3 Rescaled model

For the sake of simplicity and numerical stability in the algorithm, we can consider rescaling the state variables with respect to the total population of each node N_i so that we will deal with densities. This brings to:

$$
\begin{aligned}
\mathcal{S}^{\text{tot}} \;=\; & \sum_{i=1}^{|V|}\sum_{t=0}^{T-1} N_i \left\{ -\mathrm{i}X_i\left(t+1\right)\left(e_i\left(t+1\right)-e_i\left(t\right)\right)\right. \\
& -\left(\mathrm{i}X_i\left(t+1\right)+\mathrm{i}Y_i\left(t+1\right)\right)\left(i_i\left(t+1\right)-i_i\left(t\right)\right) \\
& -\left(\mathrm{i}X_i\left(t+1\right)+\mathrm{i}Y_i\left(t+1\right)+\mathrm{i}Z_i\left(t+1\right)\right)\left(r_i\left(t+1\right)-r_i\left(t\right)\right) \\
& -\left(1-e_i\left(t\right)-i_i\left(t\right)-r_i\left(t\right)\right)\log\left(1-\left(1-e^{-\tilde{\beta}\sum_j w_{ij}N_j i_j(t)}\right)\left(1-e^{-\mathrm{i}X_i(t+1)}\right)\right) \\
& -e_i\left(t\right)\log\left(1-\left(1-e^{-\tilde{\eta}}\right)\left(1-e^{-\mathrm{i}Y_i(t+1)}\right)\right) \\
& \left.-i_i\left(t\right)\log\left(1-\left(1-e^{-\tilde{\mu}}\right)\left(1-e^{-\mathrm{i}Z_i(t+1)}\right)\right)\right\} \\
& -\sum_{i=1}^{|V|} N_i\left\{\mathrm{i}X_i\left(0\right)e_i\left(0\right)+\left(\mathrm{i}X_i\left(0\right)+\mathrm{i}Y_i\left(0\right)\right)\left(i_i\left(0\right)-q_i i_0\right)\right. \\
& \left.+\left(\mathrm{i}X_i\left(0\right)+\mathrm{i}Y_i\left(0\right)+\mathrm{i}Z_i\left(0\right)\right)r_i\left(0\right)\right\} - \tilde{\epsilon}\sum_{i=1}^{|V|} N_i\left(q_i\log q_i+\left(1-q_i\right)\log\left(1-q_i\right)\right) \\
& -\tilde{\delta}\sum_{i=1}^{|V|} N_i q_i - \sum_{o\in\mathcal{O}}\left[\log\left(\frac{1}{\sigma\sqrt{2\pi}}\right)-\frac{N_i^2\left(i_{i_o}\left(t_o\right)-i_o\right)^2}{2\sigma^2}\right]
\end{aligned}
\tag{3.11}
$$

with rescaled variables $e_i\left(t\right)=\frac{E_i(t)}{N_i}$, $i_i\left(t\right)=\frac{I_i(t)}{N_i}$, $i_0=\frac{I_0}{N_i}$, $r_i\left(t\right)=\frac{R_i(t)}{N_i}$, $\tilde{\epsilon}=\frac{\epsilon}{N_i}$, $\tilde{\delta}=\frac{\delta}{N_i}$ and $\tilde{w}_{ij}=w_{ij}N_i$.

Again we introduce θs and the action for the rescaled model becomes:

$$\begin{aligned}
\mathcal{S}^{\text{tot}} \;=\; & \sum_{i=1}^{|V|}\sum_{t=0}^{T-1} N_i \Big\{ -\theta_i^E(t+1)\left(e_i(t+1)-e_i(t)\right) - \theta_i^I(t+1)\left(i_i(t+1)-i_i(t)\right) \\
& -\theta_i^R(t+1)\left(r_i(t+1)-r_i(t)\right) \\
& -\left(1-e_i(t)-i_i(t)-r_i(t)\right)\log\left(1-\left(1-e^{-\tilde{\beta}\sum_j \tilde{w}_{ij} i_j(t)}\right)\left(1-e^{-\theta_i^E(t+1)}\right)\right) \\
& -e_i(t)\log\left(1-\left(1-e^{-\tilde{\eta}}\right)\left(1-e^{-\left(\theta_i^I(t+1)-\theta_i^E(t+1)\right)}\right)\right) \\
& -i_i(t)\log\left(1-\left(1-e^{-\tilde{\mu}}\right)\left(1-e^{-\left(\theta_i^R(t+1)-\theta_i^I(t+1)\right)}\right)\right)\Big\} \\
& -\sum_{i=1}^{|V|} N_i\left\{\theta_i^E(0)\,e_i(0)+\theta_i^I(0)\left(i_i(0)-q_i i_0\right)+\theta_i^R(0)\,r_i(0)\right\} \\
& -\tilde{\epsilon}\sum_{i=1}^{|V|} N_i\left(q_i\log q_i+(1-q_i)\log(1-q_i)\right) \\
& -\tilde{\delta}\sum_{i=1}^{|V|} N_i q_i - \sum_{o\in\mathcal{O}}\left[\log\left(\frac{1}{\sigma\sqrt{2\pi}}\right)-\frac{N_i^2\left(i_{i_o}(t_o)-i_o\right)^2}{2\sigma^2}\right].
\end{aligned} \tag{3.12}$$

3.4 Mean-Field Approximation

Sometimes, it is useful to apply further approximations to obtain a simplified and more tractable model. The **Mean-Field approximation** is a classical tool in statistical physics that simplifies the analysis of complex interacting systems by replacing the influence of all other components on a given element with an average or **mean** effect.
This method was first developed in the context of magnetism and phase transitions, in particular in analyzing the Ising model (see e.g. [27]), then it has been widely adopted in many contexts, such as neuroscience, network theory, and epidemiology. In the context of the Ising model, the mean-field approximation can be derived in three equivalent ways which highlight different aspects of this method:

- the **effective field** approach, developed by Pierre Weiss (1907) [28], which substitutes the effect of all the neighbor spins on a given spin with an effective constant field,

- the **variational free energy** approach, which has its conceptual origins in Gibbs and Jaynes [29], but it was lately formalized in modern contexts. This method is founded on the introduction of a functional of the entropy and the internal energy $\mathcal{F}[q] = \langle \mathcal{H} \rangle_q - TS[q]$ in such a way to obtain a variational free

energy [30, 31] to be minimized with respect to the probability distribution of the system q,

- the **trial Hamiltonian** approach, in which one chooses a simplified Hamiltonian and computes its free energy, then the Bogoliubov's inequality is used: $F \leq F_0 + \langle \mathcal{H} - \mathcal{H}_0 \rangle_0$ to obtain an upper limit for the free energy F [32].

In our stochastic model, instead, the mean-field approximation can be obtained employing two assumptions:

- rates are so small that daily probabilities can be considered small as well,

$$\begin{aligned} p_{\beta,i}(t) &= 1 - e^{-\tilde{\beta}\sum_j \tilde{w}_{ij} i_j(t)}, \\ &\approx 1 - \prod_j \left[1 - \tilde{\beta}\tilde{w}_{ij} i_j(t)\right] \\ &= 1 - \prod_j \left[1 - \tilde{\lambda}_{ij} i_j(t)\right] \end{aligned} \tag{3.13}$$

  $p_\eta \approx \tilde{\eta}$ and $p_\mu \approx \tilde{\mu}$.

- only small fluctuations are considered, then we can approximate

$$1 - e^{-\mathrm{i}X_i(t+1)} \approx \mathrm{i}X_i(t+1), \tag{3.14}$$
$$1 - e^{-\mathrm{i}Y_i(t+1)} \approx \mathrm{i}Y_i(t+1), \tag{3.15}$$
$$1 - e^{-\mathrm{i}Z_i(t+1)} \approx \mathrm{i}Z_i(t+1). \tag{3.16}$$

With the usual introduction of θ-variables, we obtain a mean-field action:

$$\begin{aligned}
\mathcal{S}_{MF}^{\text{tot}} &= \sum_{i=1}^{|V|}\sum_{t=0}^{T-1} N_i \Bigg\{ -\theta_i^E(t+1)\Bigg(e_i(t+1) - (1-\tilde{\eta})\, e_i(t) \\
&\quad - (1 - e_i(t) - i_i(t) - r_i(t))\left(1 - \prod_j \left[1 - \tilde{\lambda}_{ij} i_j(t)\right]\right)\Bigg) \\
&\quad -\theta_i^I(t+1)\Big(i_i(t+1) - (1-\tilde{\mu})\, i_i(t) - \tilde{\eta} e_i(t)\Big) \\
&\quad -\theta_i^R(t+1)\Big(r_i(t+1) - r_i(t) - \tilde{\mu} i_i(t)\Big)\Bigg\} \\
&\quad - \sum_{i=1}^{|V|} N_i \Big(\theta_i^E(0) e_i(0) + \theta_i^I(0)\,(i_i(0) - q_i i_0) + \theta_i^R(0) r_i(0)\Big) \\
&\quad - \tilde{\epsilon} \sum_{i=1}^{|V|} N_i \left(q_i \log q_i + (1-q_i)\log(1-q_i)\right) \\
&\quad - \tilde{\delta} \sum_{i=1}^{|V|} N_i q_i \\
&\quad - \sum_{o\in\mathcal{O}} \left[\log\left(\frac{1}{\sigma\sqrt{2\pi}}\right) - \frac{N_i^2\left(i_{i_o}(t_o) - i_o\right)^2}{2\sigma^2}\right]. \qquad (3.17)
\end{aligned}$$

The associated saddle-point equations, which correspond to the solution with $\theta_i^E = 0, \quad \theta_i^I = 0, \quad \theta_i^R = 0$, give a deterministic mean-field description of the original stochastic model and represent the usual set of ordinary differential equations for the different epidemic compartments for the subpopulations.

3.5 Discussion and outlook

In this chapter, we provided the theoretical core of the epidemic inference problem using the powerful **path integral** formalism. This framework enables the description of the complete dynamics of our model in terms of a global action functional which allows to compute the probability of each epidemic trajectory.
We first derived the **partition function Z** and then the **action $\mathcal{S}$** associated with the model. Then we incorporated the prior information on initial conditions by also introducing the regularization and penalty terms for initial distribution q_i, and the likelihood of the data, obtaining the total action $\mathcal{S}^{\text{tot}}$.
To handle the complexity of the full dynamics, we developed a mean-field approximation. The tools developed in this chapter, along with the saddle-point equations derived in the next chapter, form the analytical core of the inference

saddle-point algorithm described later in the thesis. In particular, they provide the foundation for constructing a forward-backward algorithm that allows efficient reconstruction of epidemic trajectories and inference of unobserved variables from scattered observations.

In the next chapter, we will fill the gap between theory and practical inference by introducing the **saddle-point method**, an essential tool to convert the stochastic path integral formulation into a set of coupled equations which enables to define the most probable epidemic trajectory.

Chapter 4

The Saddle-Point Method

4.1 Introduction to the Saddle-Point method

In the previous chapter, we introduced the path integral formalism to describe the overall evolution of a stochastic epidemic system in terms of a global action, $\mathcal{S}^{\text{tot}}$, which includes all the information about the prior, namely the intrinsic dynamics of the model, and the available observations through the likelihood of the data. Although this formalism is very precise and allows one to fully describe the dynamics of the system over all the possible trajectories, it is computationally intractable.
To overcome this problem, we can use a widely adopted technique in statistical physics and field theory, the **saddle-point method**, also known as **steepest descent method** or **Laplace approximation**. This method is based on the fact that when we have a large scale parameter, such as large populations N_i in our case, the dominant contribution to the path integral stems from the configuration (or, in this case, trajectory) that minimizes the action.
This trajectory is the most probable of the system, that is the one that maximizes the posterior probability of the model. From the mathematical point of view, starting from Equation (3.1), we can approximate the integral by evaluating the action in its minimum ϕ^*:

$$\mathcal{Z} \approx e^{-\mathcal{S}[\phi^*]}. \tag{4.1}$$

This approximation is valid under the assumption that $\mathcal{S}[\phi]$ is convex and the fluctuations around the minimum are negligible, which is typical when we have large populations. This method has been widely adopted in the physics of complex systems, in both statistical classical and quantum mechanics, and in network theory. In the context of epidemic inference, Altarelli, Braunstein, Dall'Asta and Zecchina (2014) [33] showed that it is possible to derive an inference algorithm on networks using belief propagation, derived from a variational action. In this work, the saddle point of the Bethe free energy represents the analogue of the most probable

trajectory.
Similarly, Aurell and Sneppen (2002) [34] analyzed the stochastic evolution of biological systems as escape problems from stable states, by determining the most likely path of the system under stochastic fluctuations. This is represented by the solution of a Hamilton-Jacobi equation, which is conceptually equivalent to the saddle-point method for an action.
In recent years, Gratton (2020) [35] explicitly applied the **path integral formalism** to SIR models, showing how the deterministic classical dynamics can be obtained as a saddle-point of the action.
From the mathematical point of view, the saddle-point method is obtained by imposing that the functional derivatives of the action with respect to each dynamical variable and the conjugate fields are zero:

$$\frac{\delta \mathcal{S}}{\delta V_i\left(t\right)} = 0, \quad \frac{\delta \mathcal{S}}{\delta \theta_i^V\left(t+1\right)} = 0, \quad \text{where } V \in \{E, I, R\}\,. \tag{4.2}$$

This provides a system of coupled equations that describes both the forward evolution of the epidemic variables and the backward propagation of conjugate fields. In this context, the epidemic inference is performed as research of the trajectory that satisfies these stationarity conditions, subject to initial constraints and observations.
In the following sections, the saddle-point equations for our model will be derived, and we will discuss the dynamical interpretation, along with the boundary conditions and initial conditions.

4.2 Saddle-Point conditions

In this section we provide the saddle-point equations for the initial version of the model, that is the one with the variables E_i, I_i, R_i, then we will recover the equations for the rescaled model, that is the one with density variables e_i, i_i, r_i and the one implemented in the forward-backward algorithm, and eventually the equations for the mean-field model presented in Section 3.4.

4.2.1 Saddle-Point equations for the standard model

We first recall the action of the standard model from Equation (3.10):

$$
\begin{aligned}
\mathcal{S}^{\text{tot}} \;=\; & \sum_{i=1}^{|V|}\sum_{t=0}^{T-1}\Big\{-\theta_i^E(t+1)\left(E_i(t+1)-E_i(t)\right)-\theta_i^I(t+1)\left(I_i(t+1)-I_i(t)\right) \\
& -\theta_i^R(t+1)\left(R_i(t+1)-R_i(t)\right) \\
& -\left(N_i-E_i(t)-I_i(t)-R_i(t)\right)\log\left(1-\left(1-e^{-\tilde{\beta}\sum_j w_{ij}I_j(t)}\right)\left(1-e^{-\theta_i^E(t+1)}\right)\right) \\
& -E_i(t)\log\left(1-\left(1-e^{-\tilde{\eta}}\right)\left(1-e^{-\left(\theta_i^I(t+1)-\theta_i^E(t+1)\right)}\right)\right) \\
& -I_i(t)\log\left(1-\left(1-e^{-\tilde{\mu}}\right)\left(1-e^{-\left(\theta_i^R(t+1)-\theta_i^I(t+1)\right)}\right)\right)\Big\} \\
& -\sum_{i=1}^{|V|}\Big\{\theta_i^E(0)\,E_i(0)+\theta_i^I(0)\left(I_i(0)-q_iI_0\right)+\theta_i^R(0)\,R_i(0)\Big\} \\
& -\epsilon\sum_{i=1}^{|V|}\left(q_i\log q_i+(1-q_i)\log(1-q_i)\right) \\
& -\delta\sum_{i=1}^{|V|}q_i-\sum_{o\in\mathcal{O}}\left[\log\left(\frac{1}{\sigma\sqrt{2\pi}}\right)-\frac{\left(I_{i_o}(t_o)-I_o\right)^2}{2\sigma^2}\right]
\end{aligned}
$$

At this point we differentiate this action with respect to θ variables in order to obtain the deterministic equations for the forward dynamics.

$$
\frac{\delta\mathcal{S}^{\text{tot}}}{\delta\theta_i^E(t+1)}=0,\quad \frac{\delta\mathcal{S}^{\text{tot}}}{\delta\theta_i^I(t+1)}=0,\quad \frac{\delta\mathcal{S}^{\text{tot}}}{\delta\theta_i^R(t+1)}=0 \tag{4.3}
$$

Through this procedure, one obtains:

$$\begin{aligned}
\frac{\delta \mathcal{S}^{tot}}{\delta \theta_i^E\left(t+1\right)}=0 \quad \Longrightarrow \quad & E_i\left(t+1\right)=E_i\left(t\right)+\left(N_i-E_i\left(t\right)-I_i\left(t\right)-R_i\left(t\right)\right) \\
& \times \frac{\left(1-e^{-\tilde{\beta}\sum_j w_{ij}I_j(t)}\right)e^{-\theta_i^E(t+1)}}{\left(1-\left(1-e^{-\tilde{\beta}\sum_j w_{ij}I_j(t)}\right)\left(1-e^{-\theta_i^E(t+1)}\right)\right)} \\
& -E_i\left(t\right)\frac{\left(1-e^{-\tilde{\eta}}\right)e^{-\left(\theta_i^I(t+1)-\theta_i^E(t+1)\right)}}{\left(1-\left(1-e^{-\tilde{\eta}}\right)\left(1-e^{-\left(\theta_i^I(t+1)-\theta_i^E(t+1)\right)}\right)\right)} \qquad (4.4) \\
\frac{\delta \mathcal{S}^{tot}}{\delta \theta_i^I\left(t+1\right)}=0 \quad \Longrightarrow \quad & I_i\left(t+1\right)=I_i\left(t\right) \\
& +E_i\left(t\right)\frac{\left(1-e^{-\tilde{\eta}}\right)e^{-\left(\theta_i^I(t+1)-\theta_i^E(t+1)\right)}}{\left(1-\left(1-e^{-\tilde{\eta}}\right)\left(1-e^{-\left(\theta_i^I(t+1)-\theta_i^E(t+1)\right)}\right)\right)} \\
& -I_i\left(t\right)\frac{\left(1-e^{-\tilde{\mu}}\right)e^{-\left(\theta_i^R(t+1)-\theta_i^I(t+1)\right)}}{\left(1-\left(1-e^{-\tilde{\mu}}\right)\left(1-e^{-\left(\theta_i^R(t+1)-\theta_i^I(t+1)\right)}\right)\right)} \qquad (4.5) \\
\frac{\delta \mathcal{S}^{tot}}{\delta \theta_i^R\left(t+1\right)}=0 \quad \Longrightarrow \quad & R_i\left(t+1\right)=R_i\left(t\right) \\
& +I_i\left(t\right)\frac{\left(1-e^{-\tilde{\mu}}\right)e^{-\left(\theta_i^R(t+1)-\theta_i^I(t+1)\right)}}{\left(1-\left(1-e^{-\tilde{\mu}}\right)\left(1-e^{-\left(\theta_i^R(t+1)-\theta_i^I(t+1)\right)}\right)\right)} \qquad (4.6)
\end{aligned}$$

As one can see, these are the equations that lead the update of the states E, I, R forward in time.
Now we derive the equations for the backward propagation by differentiating the action with respect to the state variables:

$$\frac{\delta \mathcal{S}^{\mathrm{tot}}}{\delta E_i\left(t\right)}=0, \quad \frac{\delta \mathcal{S}^{\mathrm{tot}}}{\delta I_i\left(t\right)}=0, \quad \frac{\delta \mathcal{S}^{\mathrm{tot}}}{\delta R_i\left(t\right)}=0 \qquad (4.7)$$

This brings to:

$$\begin{aligned}
\frac{\delta \mathcal{S}^{tot}}{\delta E_i\left(t\right)} = 0 \quad \Longrightarrow \quad & \theta_i^E\left(t\right) = \theta_i^E\left(t+1\right) \\
& + \log\left(1 - \left(1 - e^{-\tilde{\beta}\sum_j w_{ij} I_j(t)}\right)\left(1 - e^{-\theta_i^E(t+1)}\right)\right) \\
& - \log\left(1 - \left(1 - e^{-\tilde{\eta}}\right)\left(1 - e^{-\left(\theta_i^I(t+1) - \theta_i^E(t+1)\right)}\right)\right) \qquad (4.8) \\
\frac{\delta \mathcal{S}^{tot}}{\delta I_i\left(t\right)} = 0 \quad \Longrightarrow \quad & \theta_i^I\left(t\right) = \theta_i^I\left(t+1\right) \\
& + \log\left(1 - \left(1 - e^{-\tilde{\beta}\sum_j w_{ij} I_j(t)}\right)\left(1 - e^{-\theta_i^E(t+1)}\right)\right) \\
& + \sum_{k=1}^{|V|}\left(N_k - E_k\left(t\right) - I_k\left(t\right) - R_k\left(t\right)\right) \\
& \times \frac{\tilde{\beta} w_{ki} e^{-\tilde{\beta}\sum_j w_{kj} I_j(t)}\left(1 - e^{-\theta_k^E(t+1)}\right)}{\left(1 - \left(1 - e^{-\tilde{\beta}\sum_j w_{kj} I_j(t)}\right)\left(1 - e^{-\theta_k^E(t+1)}\right)\right)} \\
& - \log\left(1 - \left(1 - e^{-\tilde{\mu}}\right)\left(1 - e^{-\left(\theta_i^R(t+1) - \theta_i^I(t+1)\right)}\right)\right) \\
& + \delta_{i,i_o}\delta_{t,t_o}\left(\frac{I_i\left(t\right) - I_o}{\sigma^2}\right) \qquad (4.9) \\
\frac{\delta \mathcal{S}^{tot}}{\delta R_i\left(t\right)} = 0 \quad \Longrightarrow \quad & \theta_i^R\left(t\right) = \theta_i^R\left(t+1\right) \\
& + \log\left(1 - \left(1 - e^{-\tilde{\beta}\sum_j w_{ij} I_j(t)}\right)\left(1 - e^{-\theta_i^E(t+1)}\right)\right) \qquad (4.10)
\end{aligned}$$

4.2.2 Saddle-Point equations for the rescaled model

Let's recall the expression for the total action of the rescaled model from Equation (3.12):

$$
\begin{aligned}
\mathcal{S}^{\text{tot}} \;=\; & \sum_{i=1}^{|V|}\sum_{t=0}^{T-1} N_i \Big\{ -\theta_i^E(t+1)\left(e_i(t+1)-e_i(t)\right) - \theta_i^I(t+1)\left(i_i(t+1)-i_i(t)\right) \\
& -\theta_i^R(t+1)\left(r_i(t+1)-r_i(t)\right) \\
& -\left(1-e_i(t)-i_i(t)-r_i(t)\right)\log\left(1-\left(1-e^{-\tilde{\beta}\sum_j \tilde{w}_{ij} i_j(t)}\right)\left(1-e^{-\theta_i^E(t+1)}\right)\right) \\
& -e_i(t)\log\left(1-\left(1-e^{-\tilde{\eta}}\right)\left(1-e^{-\left(\theta_i^I(t+1)-\theta_i^E(t+1)\right)}\right)\right) \\
& -i_i(t)\log\left(1-\left(1-e^{-\tilde{\mu}}\right)\left(1-e^{-\left(\theta_i^R(t+1)-\theta_i^I(t+1)\right)}\right)\right)\Big\} \\
& -\sum_{i=1}^{|V|} N_i\left\{\theta_i^E(0)\,e_i(0)+\theta_i^I(0)\left(i_i(0)-q_i i_0\right)+\theta_i^R(0)\,r_i(0)\right\} \\
& -\tilde{\epsilon}\sum_{i=1}^{|V|} N_i\left(q_i\log q_i+(1-q_i)\log(1-q_i)\right) \\
& -\tilde{\delta}\sum_{i=1}^{|V|} N_i q_i - \sum_{o\in\mathcal{O}}\left[\log\left(\frac{1}{\sigma\sqrt{2\pi}}\right)-\frac{N_i^2\left(i_{i_o}(t_o)-i_o\right)^2}{2\sigma^2}\right]
\end{aligned}
$$

We first compute the forward evolution for the state variables in the same way of Equations (4.4)–(4.6) . What one obtains is the following:

$$
\begin{aligned}
\frac{\delta\mathcal{S}^{tot}}{\delta\theta_i^E(t+1)}=0 \quad\Longrightarrow\quad & e_i(t+1)=e_i(t)+\left(1-e_i(t)-i_i(t)-r_i(t)\right) \\
& \times\frac{\left(1-e^{-\tilde{\beta}\sum_j \tilde{w}_{ij} i_j(t)}\right)e^{-\theta_i^E(t+1)}}{\left(1-\left(1-e^{-\tilde{\beta}\sum_j \tilde{w}_{ij} i_j(t)}\right)\left(1-e^{-\theta_i^E(t+1)}\right)\right)} \\
& -e_i(t)\,\frac{\left(1-e^{-\tilde{\eta}}\right)e^{-\left(\theta_i^I(t+1)-\theta_i^E(t+1)\right)}}{\left(1-\left(1-e^{-\tilde{\eta}}\right)\left(1-e^{-\left(\theta_i^I(t+1)-\theta_i^E(t+1)\right)}\right)\right)}
\end{aligned}
\tag{4.11}
$$

$$\frac{\delta \mathcal{S}^{tot}}{\delta \theta_i^I\left(t+1\right)}=0 \quad \Longrightarrow \quad i_i\left(t+1\right)=i_i\left(t\right)$$
$$+e_i\left(t\right)\frac{\left(1-e^{-\tilde{\eta}}\right)e^{-\left(\theta_i^I(t+1)-\theta_i^E(t+1)\right)}}{\left(1-\left(1-e^{-\tilde{\eta}}\right)\left(1-e^{-\left(\theta_i^I(t+1)-\theta_i^E(t+1)\right)}\right)\right)}$$
$$-i_i\left(t\right)\frac{\left(1-e^{-\tilde{\mu}}\right)e^{-\left(\theta_i^R(t+1)-\theta_i^I(t+1)\right)}}{\left(1-\left(1-e^{-\tilde{\mu}}\right)\left(1-e^{-\left(\theta_i^R(t+1)-\theta_i^I(t+1)\right)}\right)\right)} \tag{4.12}$$

$$\frac{\delta \mathcal{S}^{tot}}{\delta \theta_i^R\left(t+1\right)}=0 \quad \Longrightarrow \quad r_i\left(t+1\right)=r_i\left(t\right)$$
$$+i_i\left(t\right)\frac{\left(1-e^{-\tilde{\mu}}\right)e^{-\left(\theta_i^R(t+1)-\theta_i^I(t+1)\right)}}{\left(1-\left(1-e^{-\tilde{\mu}}\right)\left(1-e^{-\left(\theta_i^R(t+1)-\theta_i^I(t+1)\right)}\right)\right)} \tag{4.13}$$

We now calculate the backward equations:

$$\frac{\delta \mathcal{S}^{tot}}{\delta E_i\left(t\right)}=0 \quad \Longrightarrow \quad \theta_i^E\left(t\right)=\theta_i^E\left(t+1\right)$$
$$+\log\left(1-\left(1-e^{-\tilde{\beta}\sum_j \tilde{w}_{ij}i_j(t)}\right)\left(1-e^{-\theta_i^E(t+1)}\right)\right)$$
$$-\log\left(1-\left(1-e^{-\tilde{\eta}}\right)\left(1-e^{-\left(\theta_i^I(t+1)-\theta_i^E(t+1)\right)}\right)\right) \tag{4.14}$$

$$\frac{\delta \mathcal{S}^{tot}}{\delta I_i\left(t\right)}=0 \quad \Longrightarrow \quad \theta_i^I\left(t\right)=\theta_i^I\left(t+1\right)$$
$$+\log\left(1-\left(1-e^{-\tilde{\beta}\sum_j \tilde{w}_{ij}i_j(t)}\right)\left(1-e^{-\theta_i^E(t+1)}\right)\right)$$
$$+\sum_{k=1}^{|V|}\left(1-e_k\left(t\right)-i_k\left(t\right)-r_k\left(t\right)\right)$$
$$\times\frac{\tilde{\beta}\tilde{w}_{ki}e^{-\tilde{\beta}\sum_j \tilde{w}_{kj}i_j(t)}\left(1-e^{-\theta_k^E(t+1)}\right)}{\left(1-\left(1-e^{-\tilde{\beta}\sum_j \tilde{w}_{kj}i_j(t)}\right)\left(1-e^{-\theta_k^E(t+1)}\right)\right)}$$
$$-\log\left(1-\left(1-e^{-\tilde{\mu}}\right)\left(1-e^{-\left(\theta_i^R(t+1)-\theta_i^I(t+1)\right)}\right)\right)$$
$$+\delta_{i,i_o}\delta_{t,t_o}N_i\left(\frac{i_i\left(t\right)-i_o}{\sigma^2}\right) \tag{4.15}$$

$$\frac{\delta \mathcal{S}^{tot}}{\delta R_i(t)} = 0 \quad \Longrightarrow \quad \theta_i^R(t) = \theta_i^R(t+1) + \log\left(1 - \left(1 - e^{-\tilde{\beta}\sum_j \tilde{w}_{ij} i_j(t)}\right)\left(1 - e^{-\theta_i^E(t+1)}\right)\right) \tag{4.16}$$

4.2.3 Saddle-Point equations for the mean-field model

In the same way of equations above, we derive the saddle-point conditions for the mean-field approximation of our model, differentiating the action of Equation (3.17) with respect to θ variables and state variables.
For the forward dynamics one obtaines:

$$\frac{\delta \mathcal{S}_{MF}^{tot}}{\delta \theta_i^E(t+1)} = 0 \quad \Longrightarrow \quad e_i(t+1) = (1-\tilde{\eta})\, e_i(t) + (1 - e_i(t) - i_i(t) - r_i(t)) \times \left[1 - \prod_j \left(1 - \tilde{\lambda_{ij}} i_j(t)\right)\right] \tag{4.17}$$

$$\frac{\delta \mathcal{S}_{MF}^{tot}}{\delta \theta_i^I(t+1)} = 0 \quad \Longrightarrow \quad i_i(t+1) = (1-\tilde{\mu})\, i_i(t) + \tilde{\eta} e_i(t) \tag{4.18}$$

$$\frac{\delta \mathcal{S}_{MF}^{tot}}{\delta \theta_i^R(t+1)} = 0 \quad \Longrightarrow \quad r_i(t+1) = r_i(t) + \tilde{\mu} i_i(t) \tag{4.19}$$

For the backward propagation we have:

$$\frac{\delta \mathcal{S}_{MF}^{tot}}{\delta e_i(t)} = 0 \quad \Longrightarrow \quad \theta_i^E(t) = (1-\tilde{\eta})\,\theta_i^E(t+1) - \theta_i^E(t+1)\left[1 - \prod_j \left(1 - \tilde{\lambda}_{ij} i_j(t)\right)\right] + \tilde{\eta}\theta_i^I(t+1) \tag{4.20}$$

$$\frac{\delta \mathcal{S}^{tot}_{MF}}{\delta i_i(t)} = 0 \quad \Longrightarrow \quad \theta^I_i(t) = (1-\tilde{\mu})\,\theta^I_i(t+1)$$
$$-\theta^E_i(t+1)\left(1-\prod_j\left[1-\tilde{\lambda}_{ij}i_j(t)\right]\right)+\tilde{\mu}\theta^R_i(t+1)$$
$$+\delta_{i,i_o}\delta_{t,t_o}N_i\left(\frac{i_{i_o}(t_o)-i_o}{\sigma^2}\right)$$
$$+\sum_{k=1}^{|V|}\theta^E_k(t+1)\left(1-e_k(t)-i_k(t)-r_k(t)\right)$$
$$\times\left(\tilde{\lambda}_{ki}\prod_{j\neq i}\left[1-\tilde{\lambda}_{kj}i_j(t)\right]\right) \tag{4.21}$$

$$\frac{\delta \mathcal{S}^{tot}_{MF}}{\delta r_i(t)} = 0 \quad \Longrightarrow \quad \theta^R_i(t) = \theta^R_i(t+1)$$
$$-\theta^E_i(t+1)\left(1-\prod_j\left[1-\tilde{\lambda}_{ij}i_j(t)\right]\right) \tag{4.22}$$

4.3 Boundary conditions

The initial conditions can be imposed by considering the derivative of the action with respect to conjugate fields at time $t = 0$:

$$\frac{\delta \mathcal{S}^{tot}}{\delta \theta^E_i(0)} = 0 \quad \Longrightarrow \quad E_i(0) = 0 \tag{4.23}$$
$$\frac{\delta \mathcal{S}^{tot}}{\delta \theta^I_i(0)} = 0 \quad \Longrightarrow \quad i_i(0) = q_i i_0 \tag{4.24}$$
$$\frac{\delta \mathcal{S}^{tot}}{\delta \theta^R_i(0)} = 0 \quad \Longrightarrow \quad R_i(0) = 0 \tag{4.25}$$

These are the same for all versions of our model. Similarly, one can derive the final conditions; in particular, we set the conjugate fields to 0 except for $\theta^I_i(T)$, which will be given by the likelihood of the observations. We have:

$$\frac{\delta \mathcal{S}^{tot}}{\delta I_i(T)} = 0 \quad \Longrightarrow \quad \theta^I_i(T) = \delta_{i,i_o}\delta_{T,t_o}N_i\left(\frac{i_i(T)-i_o}{\sigma^2}\right) \tag{4.26}$$

The parameters q_i indicate the probability that node i is an initial infection seed; we can infer this parameter by deriving an update function to insert into the forward-backward algorithm. This is done by differentiating the action with respect

to q_i:

$$\frac{\delta \mathcal{S}^{tot}}{\delta q_i} = 0 \quad \Longrightarrow \quad \theta_i^I(0)\, i_0 - \tilde{\epsilon}\left(\log q_i - \log\left(1 - q_i\right)\right) - \tilde{\delta} = 0 \tag{4.27}$$

and one obtains the update function

$$q_i = \frac{1}{1 + e^{-\tilde{\epsilon}^{-1}\left(\theta_i^I(0) i_0 - \tilde{\delta}\right)}}. \tag{4.28}$$

4.4 Physical and Mathematical Interpretation of the Saddle-Point Equations

The equations derived in the previous section represent the stationary conditions of the total action $\mathcal{S}^{\text{tot}}$. From a mathematical and physical point of view, these equations describe the most probable dynamical trajectory of the epidemic process. We have seen in Chapter 3 that in the path integral formalism, the probability of a full trajectory is expressed as a functional integral over all possible realizations of the variables of the system, weighted by an exponential of the negative action. When we consider the limit of large populations (i.e.,$N_i \gg 1$) and small fluctuations, the integral is dominated by the path that minimizes the action. This is equivalent to the classical limit in quantum mechanics, where the path of least action corresponds to the classical equations of motion.

So the saddle-point equations can be seen as the Euler-Lagrange equations for a stochastic field theory discrete in time, and describe the typical evolution of the system.

As we have seen in previous section, we can distinguish two coupled components: the forward dynamics for the state variables $E_i(t+1), I_i(t+1), R_i(t+1)$, and the backward propagation for the conjugate fields $\theta_i^E(t), \theta_i^I(t), \theta_i^R(t)$. This dual structure is typical of Bayesian inference in dynamic systems, and is very similar to the Hamiltonian formalism in mechanics, where we have position q and the conjugate momentum p, which evolve in complementary directions.

The fields $\theta(t)$ can be seen as the response fields of the Martin-Siggia-Rose (MSR) formalism [25]. In the saddle-point approximation, they acquire a physical meaning: they act like adjoint variables that optimize the fit between the model trajectory and the observed data.

The saddle-point equations emerge from a global optimization principle, since the action integrates dynamics, prior information, and observations into a single cost functional. So, solving these equations allows to find the epidemic evolution most consistent with both the model and the data.

In the next chapter, we will present the colocation data, which describe the contact process in the network. Specifically, we will describe how colocation data provided by Meta and made available through collaboration with INSERM can be integrated into the model to reflect realistic spatial connectivity between regions. These data serve as the foundation for constructing the dynamic contact network, which plays a central role in shaping epidemic spread and enables data-driven inference within our metapopulation framework.

Chapter 5

Colocation Data and Mobility-driven Interaction Networks

5.1 Data Provenance and Coverage

In this thesis work, we utilize anonymized human mobility data derived from colocation information made available by **Meta's Data for Good** initiative [36], in partnership with public health institutions such as **INSERM** (Institut National de la Santé et de la Recherche Médicale) in France. This dataset contains information on physical proximity events between users of Meta platforms, like Facebook or Instagram, who have location services enabled on their devices.
Through these data, one is able to estimate interaction potentials across different geographical regions and, moreover, to construct a dynamical interaction network for epidemic modeling. The dataset is structured at the level of administrative regions, such as departments in France and provinces in Italy.

The data are collected weekly and represent the probability that an individual from region i has been colocated in the same area of $\sim (0.6km)^2$ for a certain amount of time (of order $O(1min)$) with another individual from region j. The time window for the collection of data can cause some issues when simulating epidemics at daily level, since we would have the same colocation rates for 7 days, causing the epidemics to assume very mean-field like behavior.
The time span we considered in this work ranges from April to June 2023. Another important clarification about these data is that *face-to-face* contacts are not considered, since they are particularly difficult and invasive to obtain, so we work

on a larger spatial scale. Access to the data is granted via partnership with Meta, typically in collaboration with academic or public health institutions. The data are not publicly downloadable, but can be shared under controlled conditions for public health research.

5.1.1 Administrative regions

As already mentioned, epidemiological models can incorporate different levels of interaction between individuals, ranging from a whole-population model (where every individual is supposed to interact with others at the same rate [37]) to agent-based models [38, 39]. In this sense, colocation data are meant to be applied to metapopulation models. The subpopulations are usually geographical regions, which in turn are conveniently identified in political administrative units, since public health decisions are usually taken within countries. We can provide, with Figure 5.1, an example of a colocation map constructed using the colocation data for Italy, with weighted links.

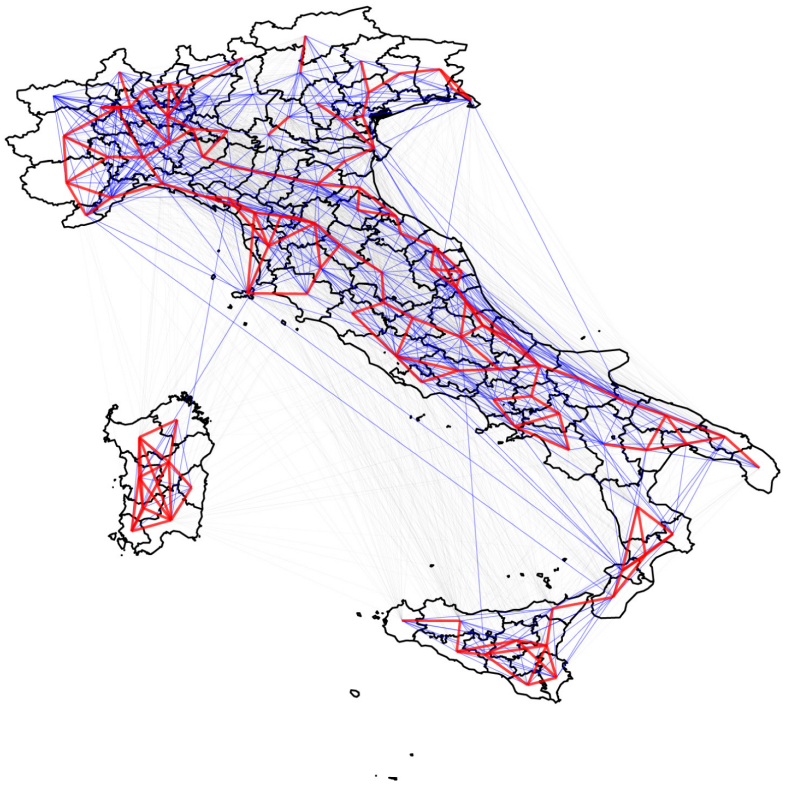

Figure 5.1: Colocation map for Italy for the week of 2020-02-26 to 2020-03-03. Red links represent strong colocation rate, blue links intermediate rate, and black are the weakest. Source: Iyer et al. (2023) [40]

5.2 Applications of Colocation Data

Colocation data have been widely used in epidemiological and human-mobility research, especially during the COVID-19 pandemic. Thanks to their spatial and time resolution, these data have enabled the modeling of interregional interactions, the assessment of containment measures, and the forecasting of epidemic dynamics. They have been employed to build contact networks between geographical units (as in this thesis) and to simulate the spread of infectious diseases. For example, in France, researchers used colocation probabilities to construct a time-varying contact matrix between departments to be implemented into ARIMA (AutoRegressive Integrated Moving Average, a technique for the analysis of time series and foresight of future values of the series) models for forecasting local hospital admissions due to COVID-19 [41].
Multiple studies adopted colocation data to evaluate the effectiveness of mobility restrictions and social distancing policies, for example, in the work of Chang et al. (2021) [42], where they examined changes in colocation probabilities between US counties before and after the order of "staying at home". A significant reduction of inter-county colocation suggested a decrease of the transmission risk, validating the impact of such policies.
Another notable application extends to digital contact tracing. For example, in a university setting, researchers used Wi-Fi access point logs to reconstruct a physical proximity network and demonstrated that these data could predict COVID-19 cases and help in early outbreak detection [43].
These applications demonstrate the flexibility and relevance of colocation data in epidemiological frameworks. In the next section, we will show the structure of this dataset and the construction of the colocation matrix W.

5.3 Dataset Structure and Colocation Matrix

5.3.1 Dataset Structure

The data provided by Meta through INSERM are compressed in CSV partitions. Each file contains all the colocation feeds for the whole world in a given day (representative of an entire week). The files are structured as represented in Table 5.1.

Column name	**Type**	**Meaning**
Row	Int	Row index of the file
polygon1_id	String15	Alpha-numeric code of the first region
polygon1_name	String31	Name of the first region
polygon2_id	String15	Alpha-numeric code of the second region
polygon2_name	String31	Name of the second region
country	String3	3-letter abbreviation of the country of the 2 regions
polygon_level	Int64	Meta's internal region partition (roughly corresponds to NUTS-3 protocol)
is_home_tile_colocation	Bool	True if the 2 regions in the table are the same, False otherwise; indicator of colocation of individuals of the same region
weekly_measured_coobservation_rate	Float64	Probability that an individual from region 1 and another one from region 2 are observed in the same 5-minute bin, regardlessly of their location
weekly_measured_colocation_rate	Float64	Probability that an individual from region 1 and another one from region 2 are observed in the same 5-minute bin, in the same $(0.6km)^2$ area
weekly_colocation_rate	Float64	Probability that any individual from first region was in the same $(0.6km)^2$ area as an individual from second region during the day
ds	Date	UTC date (YYYY-MM-DD) representing a 24-h aggregation window (representative of a week)

Table 5.1: Structure of the raw colocation dataset (Meta Data for Good)

The values in column **weekly_colocation_rate** are obtained as the ratio between the values in column **weekly_measured_colocation_rate** and the ones in column **weekly_measured_coobservation_rate**, as result of a conditional probability estimate:

$$P\left(coloc|coobs\right) = \frac{P\left(coloc \cap coobs\right)}{P\left(coobs\right)} \tag{5.1}$$

where the l.h.s. represents the value in column **weekly_colocation_rate**, then, going to the r.h.s., we have at the numerator the value we find in column **weekly_measured_colocation_rate**, and the denominator is the value in column **weekly_measured_coobservation_rate**.
Each CSV-file contains about 18 million rows.

5.3.2 Derivation of the Colocation Matrix

Once we explain the structure of the colocation dataset, we are able to derive the colocation matrix W mentioned in Section 2. We take a certain number (depending on the dimension of the network we want to describe) of provinces in Italy, filtering the CSV file multiple times in order to obtain a table containing the desired information only. Each province will represent a node of the network, so that in the matrix W the element w_{ij} will be the value of the column **weekly_colocation_rate** corresponding to the row in which we have both i and j in the two corresponding columns, **polygon1_name** and **polygon2_name**.
Once we extract these values, we can construct the matrix for a given time window, since we recall that each of these data are representative of an entire week, so we would have:

$$W(t) = \begin{bmatrix} w_{11}(t) & w_{12}(t) & \cdots & w_{1N}(t) \\ w_{21}(t) & w_{22}(t) & \cdots & w_{2N}(t) \\ \vdots & \vdots & \ddots & \vdots \\ w_{N1}(t) & \cdots & \cdots & w_{NN}(t) \end{bmatrix} \tag{5.2}$$

where all $W(t)$ for $1 \leq t \leq 7$ would be the same, and so on for subsequent times. In this way, we construct a vector in time of colocation matrices ready to use in our algorithm. Figure 5.2 shows the weighted graph for a 10-provinces network, obtained through a Python script in which the adopted colocation matrix is recovered from the CSV file of date 2023-05-01. We also provide a heatmap for the same colocation matrix W, visible in Figure 5.3.

5.4 Assumptions of Colocation

Despite the utility of colocation data in many applications such as the present context of epidemiological modeling, it is crucial to understand the assumptions

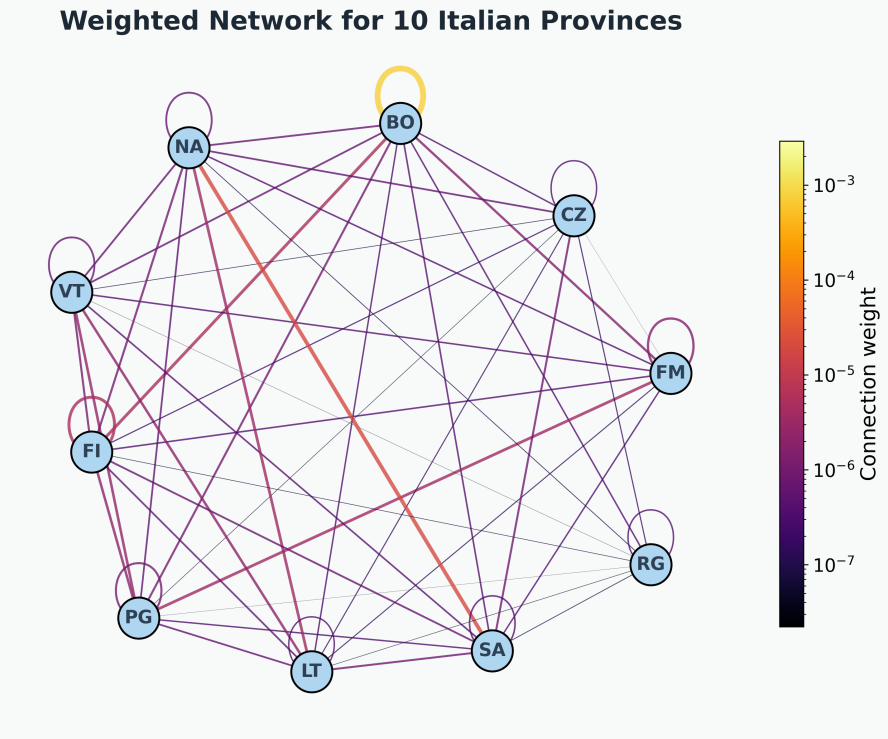

Figure 5.2: Weighted graph for a 10-nodes network. On the right the color map. Edges also have different thickness, directly proportional to their weight.

behind their construction. In this section, we will elucidate four key aspects: representativeness, within- vs. between-region colocation, contact heterogeneity, and temporal aggregation, as already done by Iyer et al. [40].

5.4.1 Representativeness

Like many datasets derived from online platforms, Colocation Maps rely on data from a specific user base, here, individuals who consented to location tracking through Meta's mobile apps. This raises concerns about demographic biases. For example, users enabling Location History (LH) and Background Collection (BC) may probably belong to a younger or more technological slice of the population. To mitigate this, Meta employs a two-stage demographic reweighting procedure:

- First, Facebook users are weighted to reflect the demographics of the on-ground population (e.g., by age, gender, wealth, and infrastructure);
- Second, the subset of users included in the colocation calculation is reweighted to resemble the overall Facebook population, incorporating additional attributes such as device type and Facebook activity level.

Despite this reweighting, differences may persist, so both raw and weighted versions of the data are shared with researchers. These allow robustness checks and adjustments in modeling.

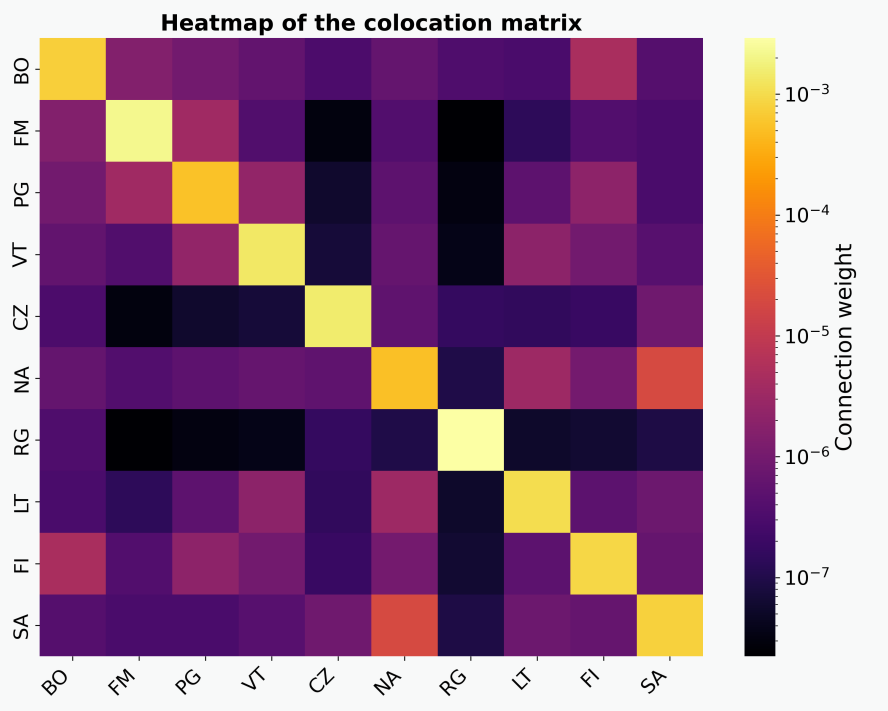

Figure 5.3: Logarithmic heatmap of the colocation matrix W between the 10 cities considered (Bologna, Fermo, Perugia, Viterbo, Catanzaro, Napoli, Ragusa, Latina, Firenze, Salerno). The chromatic scale represents the connection intensity: brighter colors indicate greater weights between nodes.

5.4.2 Within- vs. Between-Region Colocation

Colocation Maps distinguish between within-region and between-region colocation. The rates of colocation among individuals from the same region are generally orders of magnitude higher than those across regions. This difference likely reflects varying patterns of human activity, for instance, people living in the same household or neighborhood are naturally more likely to share space frequently.
However, high within-region colocation rates do not always correspond to elevated disease transmission risk. For example, individuals sleeping in the same building but in separate apartments may not interact closely. This suggests that some within-region colocation events may overstate the potential for direct contact transmission.

5.4.3 Contact Heterogeneity

Another critical consideration is the variability in colocation across different pairs of individuals. While most people are colocated for short durations with many others, a small fraction of pairs spend extensive time colocated, sometimes spanning multiple days. These few pairs contribute disproportionately to total colocation time.
This heterogeneity implies that average colocation rates may obscure important dynamics. In finer-grained models, such as agent-based or network-based simulations, it would be important to account for this variability explicitly. For instance, repeated contact between the same individuals may increase the likelihood of transmission, but such effects are typically not modeled in mean-field or metapopulation frameworks.

5.4.4 Temporal Aggregation and Homogeneity

Colocation rates in these maps are computed at weekly resolution, averaging over thousands of five-minute bins. While this temporal smoothing is computationally efficient and preserves privacy, it introduces assumptions of temporal homogeneity. That is, it treats all minutes within a week as statistically equivalent.
This may mask daily rhythms or time-specific behavior patterns (e.g., differences between work hours and nighttime). Although such smoothing is standard in mobility datasets, users should be cautious when applying weekly-aggregated rates to dynamic models that might be sensitive to diurnal patterns.
However, temporal smoothing improves privacy by blurring precise individual behaviors, one of the key motivations for Meta's approach.

5.5 Summary and Transition to Inference Implementation

The colocation data provided by Meta, through the collaboration with INSERM, offer a unique opportunity to build dynamic interaction networks at high spatio-temporal resolution. By estimating the probability that individuals from different regions are colocated in the same small area, these data provide the basis for constructing time-varying adjacency matrices that capture real-world mobility and contact patterns. Their use within metapopulation models, as shown throughout this chapter, allows a realistic parameterization of the infection kernel $w_{ij}(t)$, enabling us to move beyond homogeneous mixing assumptions.
We have illustrated the provenance, structure, and assumptions of the dataset, along with a detailed explanation of how the colocation matrix $W(t)$ is derived and interpreted. Despite inherent limitations, such as demographic biases and temporal

aggregation, the dataset remains a powerful tool when appropriately understood and used with care.

Having established the structure and interpretation of the interaction network, we are now ready to move toward the core of the inference algorithm. In the next chapter, we present the implementation of the algorithm based on the path-integral formalism, the mean-field approximation, and the saddle-point equations introduced earlier. We describe in detail how the algorithm has been coded in Julia, focusing on its structure, numerical challenges, and integration with real-world data.

Chapter 6

From Theory to Practice: Implementation of the Forward-Backward Algorithm

6.1 Overview of the Algorithmic Framework

In this chapter, we present the implementation details of the epidemic inference algorithm developed throughout the thesis. The entire computational workflow has been implemented in the Julia programming language and organized into a **custom package** called **MetaPopEpi**, which has been uploaded on GitHub as private for now.
The purpose of the algorithm is to reconstruct the most probable epidemic trajectories of epidemic variables (such as the number of exposed, infected, and recovered individuals in each region over time), based on a combination of:

- a stochastic SEIR Metapopulation model;
- first, fictitious data for the colocation matrix, and eventually real-world colocation data to model inter and intra-regional contacts;
- Bayesian Inference using a path integral formulation;
- saddle-point approximation to obtain deterministic-like equations.

The Julia package integrates all components needed for the inference:

- model parameterization and initialization;

- forward-backward inference routine to compute trajectories;
- other tools for several purposes (such as a neighboring function or a Monte Carlo simulation function).

The Julia package is built to maintain a balance between mathematical transparency and computational efficiency. The code is built on:

- typed structures to define models cleanly;
- array construction and vectorization for high-performance computation;
- external libraries for linear algebra, numerical optimization and plotting.

In the following sections, we describe in detail each component of the implementation, starting from the definition of core data structures and progressing through the numerical solution of the inference problem.

6.2 Package Tree Unpacking

In this section, we will unpack the custom Julia package in its main components and properly describe them. The package composition follows a precise tree structure for readability,ease of access and writing, and versioning.
Here we mention the main parts:

- A source directory (/src), which is the core part, containing all the .jl files, which in turn comprehend all the functions utilized in the simulations,
- a Project.toml file, which is a core component of the package, since it contains all the dependencies of the package itself (libraries like LinearAlgebra and so on),
- documentation files, useful for understanding the meaning and utility of the functions.

We now focus on the /src directory and describe each file it contains:

- types.jl, inside which we find the two main ***struct*** of the code and a new Type, the mutable structure EpidemicModel, the structure Node and the AbstractType InfectionModel respectively;
- utils.jl, which contains some of the main functions for general usage, and these are used for example for nodes formatting (creation of a complete initialized Vector of nodes where each node is a struct Node type), or for the patient zero detection and so on,

- sample.jl, this file contains the functions for the Monte Carlo simulations of the epidemic spread;
- a directory /models, which in turns contains the file SEIR.jl, that represents the core file with all the functions that simulate the dynamics of the epidemic system through the deterministic saddle-point equations;
- optimize.jl, which contains all the optimization algorithm developed and used for the main purpose of infer the most probable trajectories;
- MetaPopEpi.jl, in which we define the module of the entire package with the same name, we import all the main libraries involved in the simulations, export all the functions called in the worksheet, and eventually we include all the aforementioned files.

6.2.1 Types.jl File

This file defines the core data structures used to represent the epidemic model and its components. It begins with the definition of an 'AbstractType' for the infection model—currently instantiated as the SEIR model, but easily extendable to others.

The main structure, 'EpidemicModel', encapsulates the following elements:

- The infection model being used (e.g., SEIR);
- Spatial and temporal dimensions of the simulation: number of nodes and number of time steps;
- A vector specifying the population size at each node;
- A time-varying contact matrix, generated either synthetically via an Erdös–Renýi random graph or from real-world colocation data;
- A matrix of observations, where missing values are denoted with -1.0;
- Three parameters governing the initial conditions:
  - 'prior': initial number of infected individuals;
  - ϵ: regularization hyperparameter encouraging values of q_i close to 0 or 1;
  - δ: penalization hyperparameter encouraging sparsity in the set of initially infected nodes;
- A boolean flag 'converged' used to indicate whether the inference algorithm has converged.

A second structure, 'Node', is used to represent each node in the simulation graph. It includes:

- The node index i;
- A time-dependent vector of neighboring nodes, $\partial(t)$;
- Two matrices, 'margfwd' and 'margbwd', holding parameters for forward and backward inference respectively (e.g., $E_i(t)$, $I_i(t)$, $R_i(t)$ and $\theta_i^E(t)$, $\theta_i^I(t)$, $\theta_i^R(t)$);
- A node-specific standard deviation σ_i, representing uncertainty in observations, modeled via a Gaussian distribution centered on observed infection counts.

Tables 6.1 and 6.2 summarize the fields of the two structures.

Field	**Type**	**Description**
Disease	InfectionModel	The infection model used
M	Int	Number of nodes in the contact graph
T	Int	Number of time steps
Ns	Vector{Float64}	Number of individuals in each node
ws	Vector{SparseMatrixCSC{Float64,Int64}}	colocation matrices at each time step
prior	Float64	Initial number of infected individuals
ϵ	Float64	Hyperparameter for the regularization of the prior
δ	Float64	Hyperparameter for the penalization of the number of infected nodes
obs	Matrix{Float64}	Observation matrix
converged	Bool	Flag for convergence check

Table 6.1: EpidemicModel structure, each field is present according to the discussion made above

Field	Type	Description
i	Int	Index of the node
∂	Vector{Vector{Int}}	List of neighbors
margfwd	Matrix{Float64}	Matrix containing parameters for the forward algorithm
margbwd	Matrix{Float64}	Matrix containing parameters for the backward algorithm
σ	Float64	Observation weight for the state of the nodes

Table 6.2: Node structure fully explained

This design offers several key advantages:

- Julia's 'struct' system enables strongly typed, memory-efficient data containers, supporting high-performance execution through the JIT compiler;
- The use of an abstract infection model type allows easy extension to alternative compartmental models (e.g., SIR, SIS, SEIRS) with minimal changes to the codebase;
- Time-varying neighbor vectors $\partial(t)$ reflect the dynamic nature of the underlying contact network, allowing more realistic modeling of temporal interactions;
- Assigning a variable observation variance σ_i to each node provides flexibility and robustness in matching simulations with observed data, improving convergence.

The 'Types.jl' file thus defines the foundational data layer on which the entire inference framework is built. In the next sections, we will explore how these structures are leveraged in implementing the epidemic inference and simulation pipeline.

6.2.2 Utils.jl File

This file contains general-purpose utility functions that support various stages of the inference pipeline, including preprocessing, model initialization, and post-inference diagnostics. While these functions are not directly involved in the inference algorithm, they play a crucial role in organizing data, assigning weights, and extracting meaningful statistics from simulated or inferred epidemic trajectories. The main functionalities provided are:

- **Node Formatting and Network Initialization**: One of the initial tasks in the simulation pipeline is to extract the neighborhood structure from the

colocation matrix and format the nodes accordingly.
- The function `get_neighbors` derives the list of neighbors for each node by identifying non-zero entries in the colocation matrix.
- The function `nodes_formatting` initializes a `Vector` of `Node` structures, assigning a unique index to each and preallocating memory for time-dependent variables.

- **Observation Weighting**: The function `obs_weight` computes the likelihood of observing a given number of infected individuals under a Gaussian observation model, given the current simulation state. This allows probabilistic integration of noisy data into the inference.
  - An additional function dynamically updates σ_i for each node based on the empirical error between observed and inferred data.

- **Marginals Collection**: Several `get_marginals` functions are defined to extract marginal distributions from the `nodes` vector. These marginals can be used for both inference and Monte Carlo simulations.

- **Risk Assessment**: The module includes a function to assess infection risk in a selected region (node), using partial observations from other nodes.
  - This function performs multiple optimization runs under varying conditions (e.g., observation percentages, reference trajectories, and inference seeds).
  - Parallel computation via `Threads` accelerates this process, and results are aggregated to compute average normalized L_2 distances from ground-truth trajectories.
  - These diagnostics are later used to generate graphical representations of risk and model performance.

All functions in this module are intentionally decoupled from the core inference logic, following modular design principles. This separation enhances maintainability, facilitates testing and debugging, and allows for easy integration of new functionalities.

6.2.3 Sample.jl File

This file contains the functions for stochastic sampling of epidemic trajectories, which are fundamental for both simulating synthetic data and evaluating inference strategies in controlled environments. Its main purpose is to generate Monte Carlo simulations of the epidemic process with fixed known parameters, which are used as target trajectories (ground state) from which we take observations, and to test algorithmic performance.

- **Monte Carlo Sampling of Epidemic Trajectories:** The core function, `sample_metapop`, performs a forward stochastic simulation of the SEIR model.
  - Transitions between compartments (e.g., susceptible → exposed, exposed → infected) are modeled using Binomial sampling, conditioned on the current system state and the infection pressure derived from the contact network $W(t)$.
  - The function internally calls `sample_single_node!`, which performs the compartmental update for a single node.
  - The complete result is stored in a 3D trajectory tensor `traj[i, t, k]`, where i is the node index, t the time step, and k the compartment (E, I, or R).

  These trajectories are used for two main purposes:

  – To generate synthetic observations that will be fed into the inference algorithm.

  – To serve as **ground-truth references** for comparing inferred trajectories and evaluating the accuracy of source localization.

- **Score Function for Approximate Bayesian Computation (ABC):** A second important component in this file is the implementation of a score function used in ABC-based inference.
  - This score measures the discrepancy between a simulated trajectory and a reference trajectory (or observation data).
  - In the context of zero-patient inference, it quantifies how closely a trajectory generated with a candidate q_i (initial infection vector) matches the observed data.
  - The score is computed as the L_2 norm between the simulated and observed trajectories at observation points.
  - This raw distance is then weighted using a Gaussian distribution to assign likelihood scores, enabling a probabilistic ranking of nodes by their likelihood of being the origin of the epidemic.

This module is essential for the inference framework: it defines the ground-truth dynamics that the saddle-point forward-backward algorithm seeks to recover, given partial and noisy observations of the epidemic spread.

6.2.4 SEIR.jl File

Together with `optimize.jl`, this file represents the computational core of the package. It encodes the SEIR compartmental model dynamics and provides the mathematical machinery required for both simulation and inference. It builds on

the types defined in Section 6.2.1 and contains model-specific logic for forward and backward propagation, likelihood evaluation, and optimization interfacing.

- **SEIR Disease Struct:** The file begins with the definition of a struct that stores the key epidemiological parameters: infection rate β, transmission rate η, and recovery rate μ. This struct is a subtype of the abstract type `InfectionDisease`, defined in `types.jl`, enabling easy generalization and substitution of models (e.g., switching from SEIR to SIR).

- **Forward and Backward Propagation Functions:** These functions are central to the saddle-point inference algorithm:

  - The forward function, `update_forward!`, implements Equations (4.11)–(4.13), computing the forward marginals for E, I, and R. It uses current state variables at time t and conjugate fields at $t+1$ to evaluate transition probabilities, then updates the marginals at $t+1$ in-place. A convergence measure is computed via `compute_convergence`, and damping is applied for numerical stability.
  - The backward function, `update_backward!`, performs the reverse-time update of conjugate fields θ based on Equations (4.14)–(4.16). It clamps θ values to a numerical cutoff for stability, since unlike state variables (bounded in [0,1]), conjugate fields are unbounded.
  - A simplified forward function, `only_forward!`, is also provided, which omits θ and implements a mean-field approximation.

- **Stochastic Simulation:** The file also defines `sample_single_node!`, the core routine for single-node updates in the Monte Carlo simulations discussed in Section 6.2.3. This function precomputes transition probabilities, performs stochastic binomial sampling of daily increments ΔE, ΔI, ΔR, and updates state variables in-place for each time step.

- **Action Functional Computation:** The `action` function calculates the total action $\mathcal{S}^{tot}$ across multiple candidate seeds. For each initial seed, the inference algorithm is run and the resulting action value is stored in a dictionary indexed by seed. The output is then sorted to identify the most probable infection origin based on the minimum action value.

- **Optimization utilities:** The file includes several utility functions to enable gradient-based optimization via packages such as `NLopt.jl` or `Optim.jl`. Parameters are flattened into a single vector of the form:

$$[A[i,t], A[i,t+1], \ldots, A[i+1,t], A[i+1,t+1], \ldots, B[i,t], \ldots]$$

where A and B refer to model-specific variables (e.g., θ_E, θ_I). Core utilities include:

- `unpack_theta`: converts the flat parameter vector into structured matrices $\theta_i^E(t)$, $\theta_i^I(t)$, and $\theta_i^R(t)$ for use in the saddle-point update; this allows for a better manageability during the update step for single variables,
- `initial_guess`: constructs an initial point for the optimizer, either from mean-field solutions (if we want to optimize both state variables and θs) or via random perturbations (for θs only),
- `compute_loss_theta`: evaluates the loss function, i.e., the squared deviation from the saddle-point conditions over time and space. This function calls `simulate_forward!` to compute epidemic trajectories (without optimizing forward marginals), then computes the updated conjugate fields via backward propagation and accumulates the loss:

$$\begin{aligned} L &= \sum_{t=1}^{T}\sum_{i=1}^{M}\left[\left(\theta_E\left[i,t\right]-\theta_E^{update}\right)^2+\left(\theta_I\left[i,t\right]-\theta_I^{update}\right)^2+\left(\theta_R\left[i,t\right]-\theta_R^{update}\right)^2\right] \\ &\quad+\sum_{i=1}^{M}\Big[\left(\theta_E\left[i,T+1\right]\right)^2+\left(\theta_R\left[i,T+1\right]\right)^2 \\ &\quad+\left(\theta_I\left[i,T+1\right]-\delta_{i,i_o}\delta_{T+1,t_o}\left(N_i\frac{\left(I\left[i,T+1\right]-I_{obs}\left[i,T+1\right]\right)}{\sigma^2}\right)\right)^2\Big]. \end{aligned} \tag{6.1}$$

- **Automatic Differentiation (AD):** The function `loss_with_grad_theta` wraps the loss function and computes its gradient using AD. Two approaches are implemented for comparison:
  - `ForwardDiff.jl` (forward-mode): efficient for low-dimensional problems; derivatives are propagated alongside function evaluations using dual numbers.
  - `ReverseDiff.jl` (reverse-mode): suitable for high-dimensional parameter spaces with scalar loss output; gradients are accumulated during a reverse traversal of the computation graph.

`ForwardDiff` functions operate on vectors of type `AbstractVector{Real}`, while `ReverseDiff` functions require standard `Vector{Float64}` inputs. A comparative performance analysis is presented in Figure 6.1 and Table 6.3.

This file forms the computational backbone of the package, enabling saddle-point inference, trajectory simulation, and parameter optimization for SEIR epidemic modeling. The next section introduces `optimize.jl`, which completes the core algorithmic framework, followed by `MetaPopEpi.jl`, which integrates all components into a working pipeline.

6.2.5 Optimize.jl and MetaPopEpi.jl Files

The **optimize.jl** file contains the core routines for trajectory reconstruction through a forward-backward inference scheme. The file is structured to include both utility functions for intermediate updates and full procedures for inference and optimization.
We begin describing the essential functions that define the boundary behavior of the system:

- **Initial condition update** (`update_priors!`): updates the vector of initial infection probabilities q_i, according to Equation (4.28), using the results of the backward pass;

- **Final condition update** (`update_endpoint!`): sets the terminal values of the conjugate fields $\theta^E, \theta^I, \theta^R$ based on the last forward simulation. If an observation at final time is present, θ^I is assigned accordingly; otherwise, all θ fields are initialized to zero.

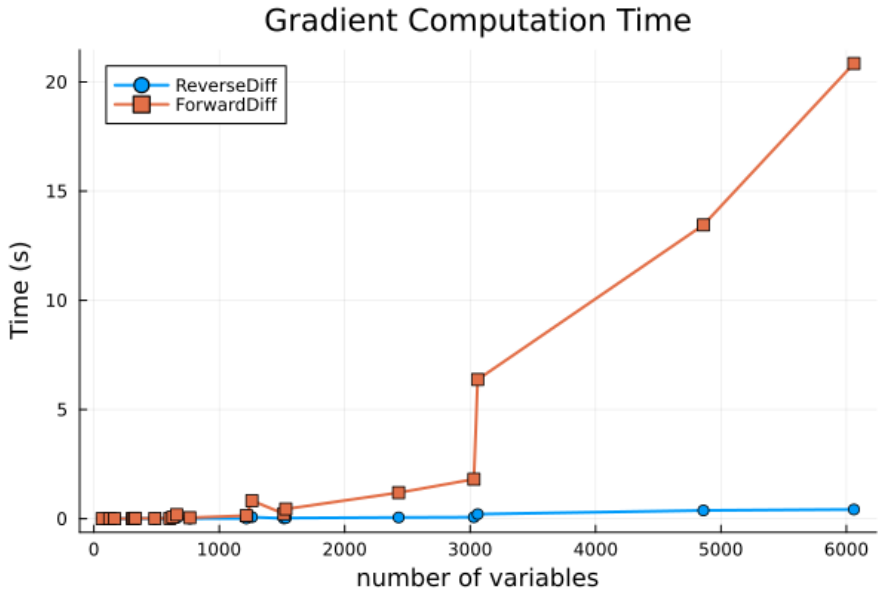

Figure 6.1: Comparison between gradients performed through ForwardDiff.jl (orange line) and ReverseDiff.jl (blue line). It is easy to notice that proceeding to augment the number of variables the forward differentiation starts to widely increase its computational time, instead of reverse differentiation, which works very well.
This graph has been obtained by evaluating the two gradients in the same conditions for the system through different combinations of nodes and time steps; the vector for the number of nodes is $M = [2,5,10,20]$, while for times we used $T = [10,20,50,80,100]$; infection parameters are $\beta = 0.36, \eta = 0.1, \mu = 0.2$. Computational times have been obtained through **BenchmarkTools.jl**.

The core of this file is the optimization algorithm, accompanied by other optimization functions which have been implemented for testing other approaches to the same problem, which are the aforementioned **NLopt.jl** and **Optim.jl** optimization tools:

- **Forward-Backward Saddle-Point Algorithm**: the function which performs the algorithm is `optimize_dynamics`, which combines both propagation directions to implement a full iteration of the inference algorithm, then it iterates this procedure many times until convergence:
  - Initialization of the vector `nodes` via `nodes_formatting`;
  - Update of the initial condition (q_i) via `update_priors!`;
  - Forward pass using `update_forward!`, which computes the trajectory of the epidemic states $E_i(t)$, $I_i(t)$, and $R_i(t)$;

  - Final condition update via `update_endpoint!`, providing a starting point for the backward computation;
  - Backward pass using `update_backward!`, which computes the conjugate fields $\theta_i^E(t)$, $\theta_i^I(t)$, and $\theta_i^R(t)$;
  - Convergence check: if convergence is not reached, the loop is repeated with updated q_i values derived from $\theta_i^I(t = 1)$.

  This iterative loop refines epidemic state estimations by alternating forward and backward updates, and solves the saddle-point equations introduced in Chapter 4. A schematic representation is shown in Figure 6.2.

- **Mean-Field Simulation**: a simplified approximation is provided by the `only_forward!` function, which applies forward propagation alone, omitting the backward pass. This yields a mean-field approximation of the system's dynamics and ignores fluctuations.

- **NLopt optimization algorithm**: in addition to the saddle-point iteration, the file provides two gradient-based optimization routines using the `NLopt.jl` library:

  - `nlopt_optimizer_`θ`_forward`: uses `ForwardDiff.jl` for forward-mode differentiation. Suitable for small to moderately sized problems;
  - `nlopt_optimizer_`θ`_reverse`: uses `ReverseDiff.jl` for reverse-mode differentiation. This implementation scales better with system size and is recommended for large-scale inference tasks.

  Both functions have a similar setup and rely on the **LD_LBFGS** method from the **NLopt.jl** library, which is a quasi-Newton method particularly useful for large-scale optimization problems with smooth gradients. The objective function minimized in both cases is the loss function. They proceed in the following way: Both functions rely on the `LD_LBFGS` algorithm, a quasi-Newton method effective for high-dimensional problems with smooth loss functions. The optimization process follows these steps:

  - Initialization of the vector of conjugate variables θ;
  - Definition of a wrapper function for logging, which encapsulates the loss and gradient computation;
  - Setup of the optimization problem: selection of algorithm, definition of parameter bounds, convergence criteria, and maximum iteration count;
  - Construction of the final θ matrices from the optimized result, which are used to reconstruct the epidemic trajectories.

M	**T**	**forward (ms)**	**reverse (ms)**
2	10	0.19	0.14
2	20	0.65	0.27
5	10	1.86	0.77
2	50	3,78	0.73
5	20	7.44	1.64
10	10	17.2	6.67
2	80	10.0	1.23
2	100	15.2	2.0
10	20	69.8	12.8
20	10	188.28	43.68
5	50	47.58	5.68
5	80	137.15	10.35
20	20	824.87	72.0
5	100	226.39	13.81
10	50	443.92	31.03
10	80	1192.04	56.88
10	100	1809.29	68.21
20	50	6372.61	207.73
20	80	13452.89	378.08
20	100	20844.94	421.84

Table 6.3: Computational time comparison between forward and reverse differentiation already shown in Figure 6.1; here the time values for the two method are compared (columns **forward** and **reverse**), along with the dimensions of the system (columns **M** and **T**). Rows are ordered according to increasing system dimensionality (total number of variables given by $3 * M * (T + 1)$).

Finally, the file **MetaPopEpi.jl** encapsulates all files mentioned above and all the functions and libraries that are utilized inside the python-notebook to perform all types of simulations and trials, representing the global module of the entire custom Julia package.
In the next section, we spend few lines for elucidating some technical concepts about some tools and packages utilized in the code.

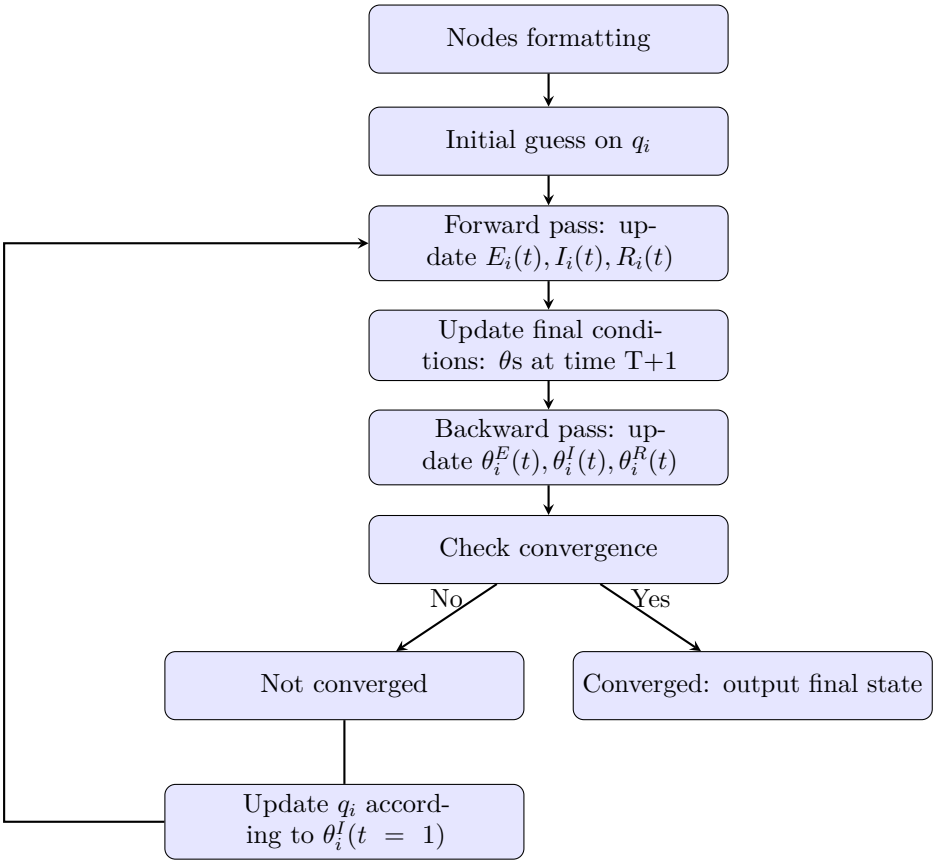

Figure 6.2: Schematic diagram of the forward-backward inference loop.

6.3 Technical Remarks and Implementation Tools

In this final section of the chapter, we provide further insight into technical components and packages employed throughout the implementation of the inference algorithm.

- **Automatic Diferrentiation (AD) in Julia**: as seen in previous sections, the minimization of the loss function requires to compute gradients with respect to a high-dimensional vector of parameters to optimize. For this purpose, it is highly recommended the use of **automatic differentiation** tools, and two major Julia packages are utilized in this context: the already mentioned **ForwardDiff.jl** and **ReverseDiff.jl**.
  The first one is based on dual number arithmetic and is accurately described in the work of Revels et al. (2016) [44], the second one on computational graph tracing, whose baselines can be understood from the work of Innes

(2018) [45], in which another useful package is described (**Zygote.jl**) but they share a similar functioning.

- **Optimization Libraries**: To find the optimal trajectories through the minimization of the loss function for θs, we rely on gradient-based optimization using **NLopt.jl**, a Julia wrapper for the NLopt library, which provides various and global optimization algorithms, such as the one we used, **LD_LBFGS**, a limited memory BFGS quasi-Newton method [46]. It is particularly suited for large-scale problems. It iteratively updates the parameter vector by approximating the inverse Hessian (second derivative matrix), using only gradient information.

- **Wrapper Usage in Optimization**: in order to monitor convergence and log loss values at each iteration, we defined a custom wrapper around the loss function. This procedure is quite common in iterative solvers where intermediate diagnostics are not present by default. The use of **Ref()** objects, dynamic arrays for loss history, and the separation of the gradient computation inside a wrapper are consistent with the usual setting in optimization workflows.

Finally, it is worth clarifying the nature of the optimization methods used in this work. The algorithms employed through the **NLopt.jl** library, with the **LD_LBFGS** method, belong to the family of **gradient-based optimization algorithms**. These methods are based on the fundamental principle of **gradient descent**, in which a function is minimized by iteratively moving in the opposite direction of its gradient:

$$x_{t+1} = x_t - \eta \nabla \mathcal{L}(x_t),$$

where η is the learning rate and $\mathcal{L}(x)$ is the loss function.
However, unlike simple gradient descent, quasi-Newton methods like the one we used works with approximations to the Hessian matrix to improve convergence rate and robustness. This methods are more suitable for high-dimensional spaces, which is our case, where reverse or forward differentiation allows for the implementation of gradient-based inference even in complex models with many parameters.
As one can read in the review by Ruder (2016) on optimization in deep learning, "gradient descent and its variants lie at the heart of optimization for machine learning models", and their extensions, including quasi-Newton methods, have been crucial for practical performance in several working fields, from deep learning to biology and statistical inference [47].

6.4 Towards Inference Results

Throughout this chapter, we have presented the implementation details of the inference algorithm developed for this thesis work. We have organized the Julia

code into modular files, each one with a specific role as model definition, simulation, optimization and so on, in such a way to build a robust and flexible framework for epidemic inference over mobility-driven networks.
We also introduced techniques like **automatic differentiation**, **MonteCarlo sampling**, **gradient-based optimization**, as well as the use of some essential libraries like **ForwardDiff.jl** and **ReverseDiff.jl** to implement efficient and scalable computations.
Having established the algorithmic and technical framework, we are now ready to evaluate the performance of the proposed method, showing some results. In the next chapter, we will present a series of experiments and numerical results that highlight the capabilities of our inference procedure. These include first of all reconstruction of trajectories according to some observation information.

Chapter 7

Results and Performance Evaluation

Despite the theoretical robustness of the proposed framework and the completeness of the algorithmic pipeline described in the previous chapters, the numerical experiments conducted as part of this thesis have yielded only partially satisfactory results.
Even though the implementation successfully reconstructs plausible epidemic trajectories under controlled conditions, in particular using the NLopt optimization library, some challenges persist. In particular, a strong dependence on the choice of initial conditions is evident, especially the initialization of q_is.
So, inference seems to be effective under favorable prior assumptions. In some cases, especially when analyzing atypical trajectories (trajectories that mean-field approximation fails or struggles to approximate), one or more nodes fall into what appears to be a **local minimum** during optimization, failing to correctly adapt their state variables to the correct trajectories, even though in the observation points there is good agreement.
This behavior highlights a limitation of the current method in escaping suboptimal configurations, despite the use of gradient-based methods and the introduction of a Gaussian noise on the problematic nodes to escape the local minimum. Nevertheless, these results are worthy, since they show that the developed framework is capable of producing meaningful results, but they also underline the importance of initialization and the risk of local minima.
We will now describe:

- the experimental setup,
- the results obtained with different optimization strategies,
- a final reflection on the limitations observed (which will be deeply explored in

the final chapter, along with the future directions on this topic).

7.1 Experimental Setup

To evaluate the performance of the inference algorithm, we construct a reference scenario, where the true epidemic trajectories are known by construction. These reference trajectories are obtained through Monte Carlo simulations of the stochastic SEIR metapopulation model introduced in Chapter 2.
The dynamics evolves over a finite time horizon starting from an initial condition in which a single node is infected, while all the others are fully susceptible. The interaction network is constructed through an Erdös-Rényi random graph, as introduced in Section 2.1.
The colocation matrix is then directly constructed from the adjacency matrix of the chosen graph, assigning weights to the edges to reflect contact intensity.
From each simulation, we extract observations by simply updating the observation matrix with the values of the infected trajectory at certain time points for some (or all) nodes. These values are then utilized to compute the θs using a Gaussian weight. This observation model matches the likelihood structure described in Section 2.3.3 and is integrated into the inference framework through the forward-backward algorithm (Chapter 6).
Before running the inference algorithm, the colocation matrix is rescaled in order to match the setup of the epidemic model used (which is the rescaled SEIR of Section 4.2.2). Finally, the inference algorithm starts.

7.2 Results from NLopt Optimization

To solve the inference problem depicted in the previous chapters, we implemented a numerical minimization of the total loss function using the NLopt library. This allows to identify the most probable epidemic trajectory consistent with the given observations, using gradient-based methods to efficiently explore the the high-dimension configuration space.
The results shown in this section correspond to the best-performing runs in terms of convergence, accuracy, and stability.

7.2.1 Optimization Setup

The chosen algorithm in the NLopt optimizer is LBFGS, selected for its performance in large-scale smooth optimization problems. The gradients of the loss function with respect to the variables to optimize are computed automatically through the library ReverseDiff.jl, which was already mentioned and described in the previous

chapter.
The optimization ends either when a maximum number of iterations is reached or when the relative variation in the loss function or in the optimization variables drops below a predefined threshold (which we set to $10^{-5} \div 10^{-6}$).

7.2.2 Reconstruction of Epidemic Trajectories

In Figures 7.1–7.4 we show the comparison between the ground-truth trajectories generated by the MonteCarlo simulation and the inferred ones obtained through optimization. In particular, we show only few nodes to give an idea of the algorithm performance. Trajectories represent the evolution in time of the number of infected individuals.

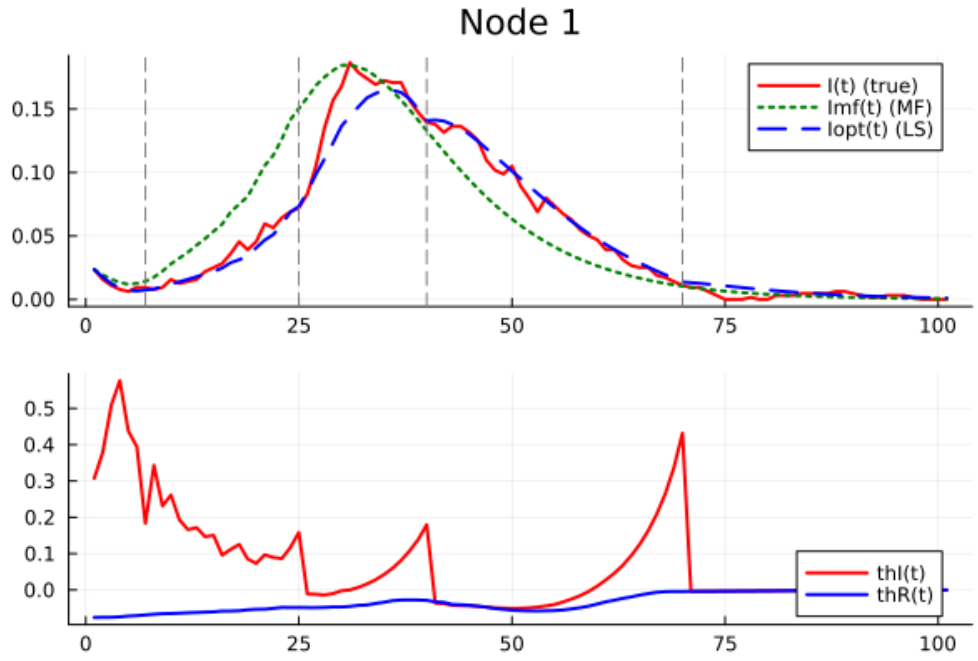

Figure 7.1: Evolution of the infected individuals in time (top graph): red curve represents the MonteCarlo trajectory, green curve represents the mean-field approximation, blue curve is the inferred trajectory. Backward evolution of θ^I and θ^R in time (bottom graph): red curve is θ^I, blue curve is θ^R. The curves are obtained for a 10-nodes network with epidemic parameters $\beta = 0.36, \eta = 0.1, \mu = 0.2$, the prior is set into node 1, the observations are taken for all nodes at time steps t = 7,25,40,70 and are represented by vertical dashed lines in the figure. The same setup is extended to the following figures, which represent curves for other nodes.

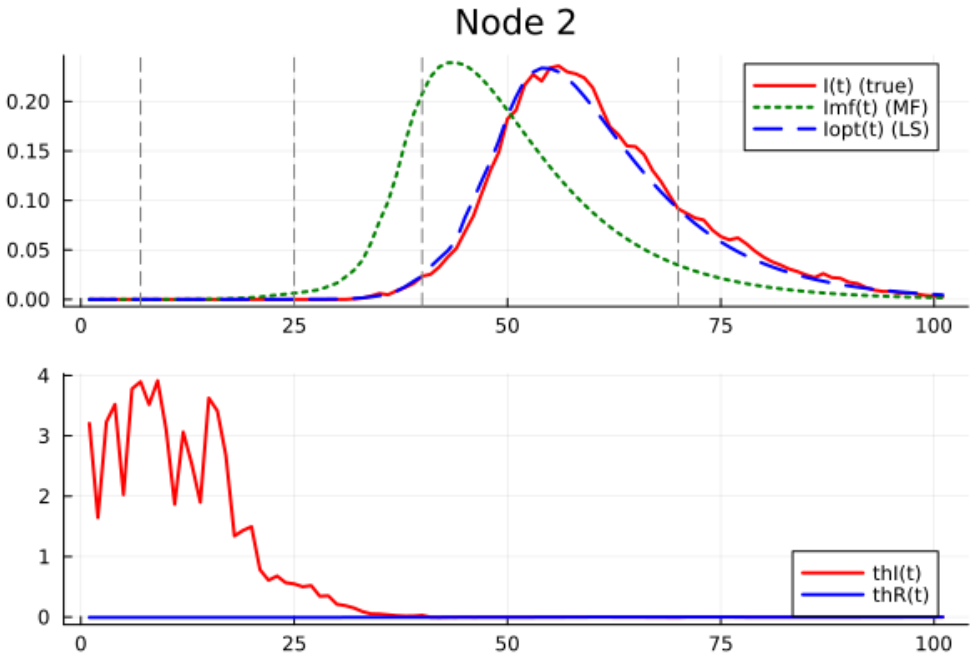

Figure 7.2: Evolution of node 2.

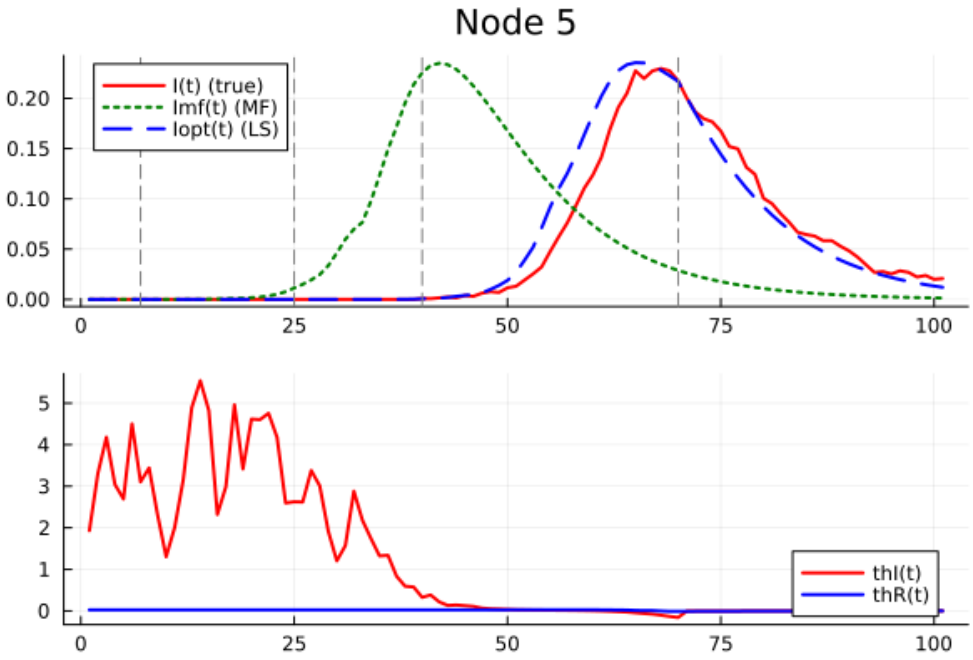

Figure 7.3: Evolution of node 5.

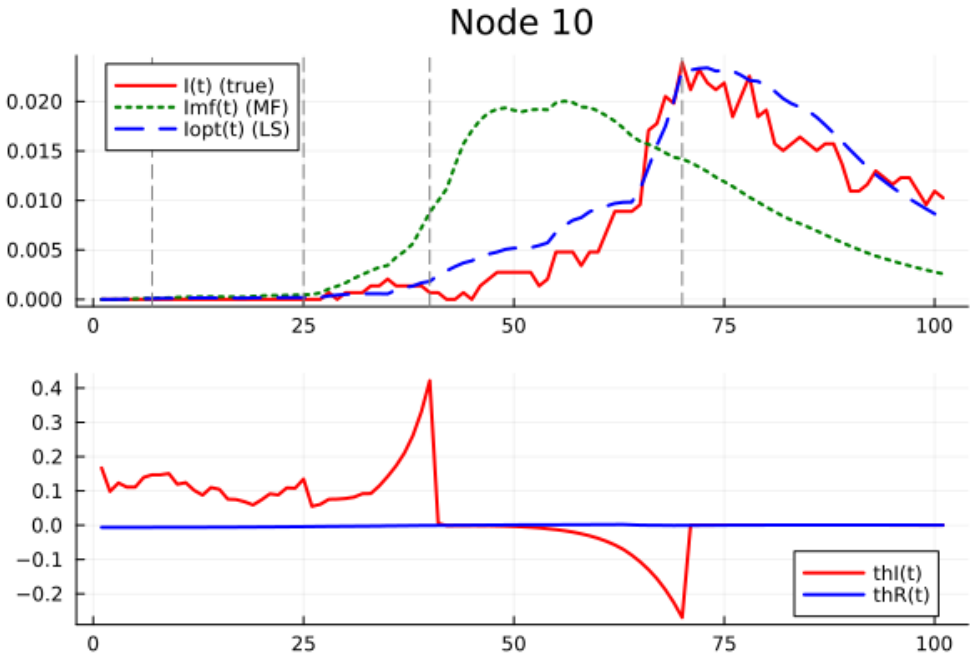

Figure 7.4: Evolution of node 10.

In these figures, one can see that, despite the atypicality of MonteCarlo trajectories, the inference algorithm is able to produce curves which greatly approximate the reference ones, instead of the mean-field approximation, which hardly recognize the right pattern of the time evolution.
Despite the good agreement of these curves for the nodes in the figures above, some challenges are still present, as already mentioned in the sections above, for example for node 6, which is depicted in Figure 7.5.

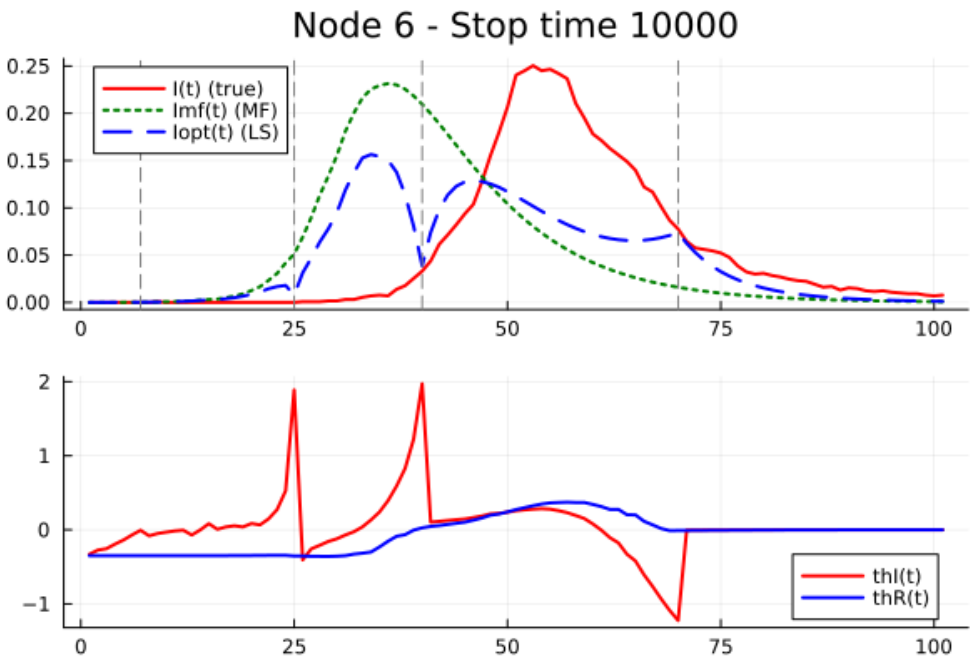

Figure 7.5: Evolution of node 6 (bad node): it is evident that in the observation points (indicated by grey vertical dashed lines) the inferred curves are forced to converge on the correct value of infected individuals, but it can't correctly reproduce the epidemic evolution.

This behavior is related to the atypicality of the reference trajectories, since it is not always present. For example, generating another MonteCarlo trajectory with a different **RNG seed** (Random Number Generator), we noticed that each node trajectory is correctly inferred, as one can see in Figures 7.6–7.8, where we show only few nodes.
An interesting point of view can be related to risk assessment, which is the possibility to infer crucial information about certain unobserved nodes, based on known information about other observed nodes. In fact, back to the MonteCarlo trajectories of Figures 7.1–7.5 (where node 6 (Fig. 7.5) is badly inferred), we changed the observation setup by reducing the number of observed nodes, and increasing the time steps of observation; in this way, it is possible to understand some interesting information about the inner functioning of the algorithm.

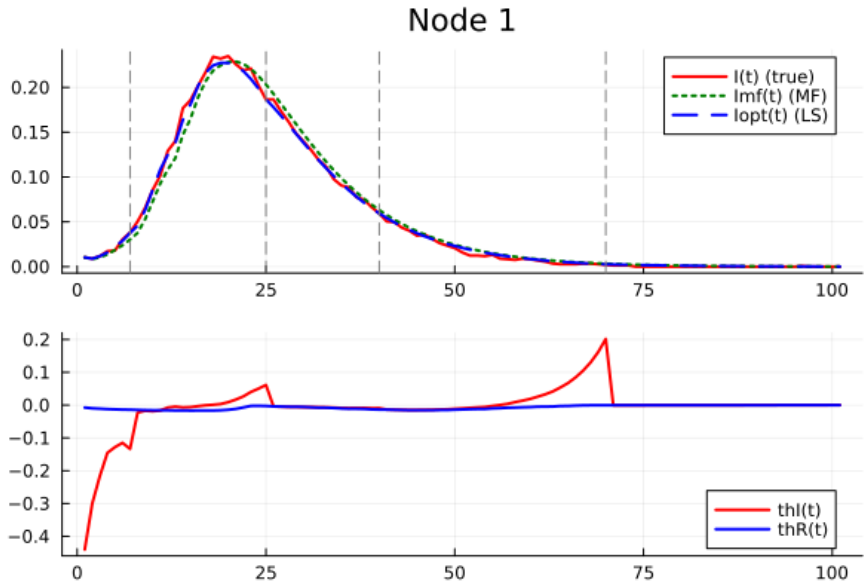

Figure 7.6: Evolution of node 1 for a typical MonteCarlo trajectory: also mean-field approximation is in good agreement.

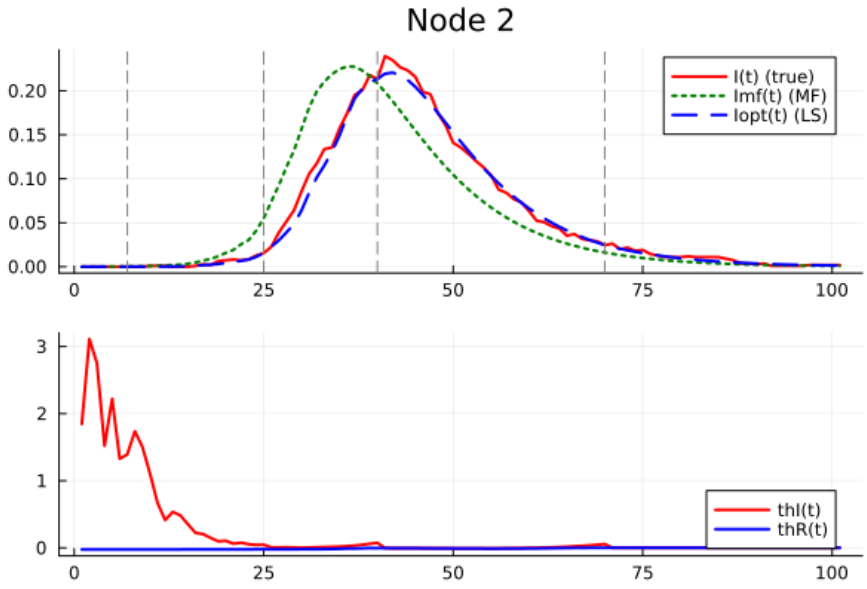

Figure 7.7: Evolution of node 2 for a typical MonteCarlo trajectory.

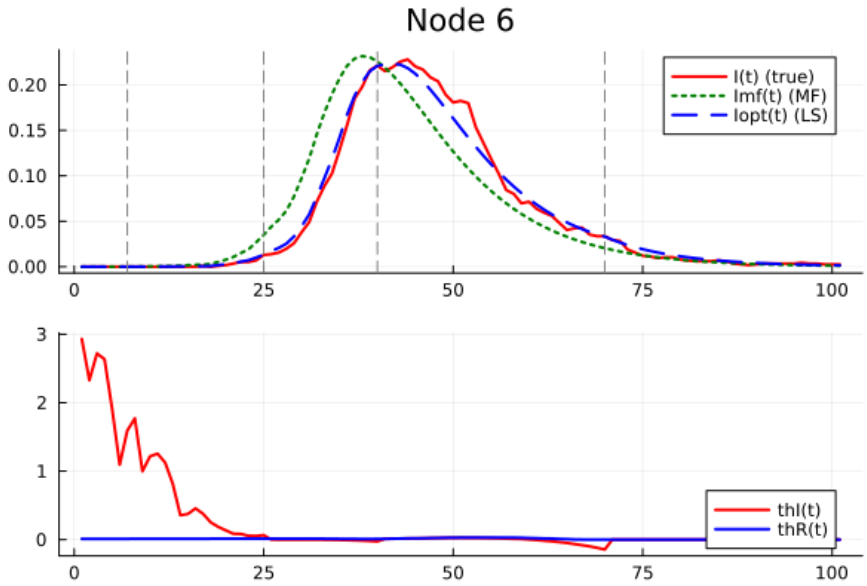

Figure 7.8: Evolution of node 6 for a typical MonteCarlo trajectory.

The results for the risk assessment are shown in Figures 7.9–7.12, where it is evident that the inferred trajectory of node 6 (Fig. 7.11) is very consistent with the MonteCarlo simulation, since it is one of the observed nodes. Observing the inferred trajectories of unobserved nodes (figs. 7.10 and 7.12), we can deduce that the algorithm works quite well, also for atypical trajectories, and the quality of curves is related to the connectivity between observed and unobserved nodes.

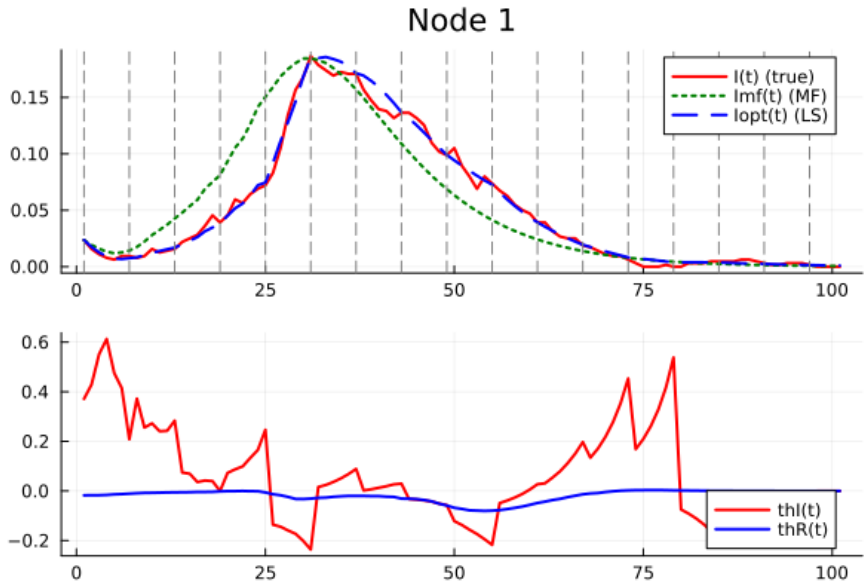

Figure 7.9: Evolution of node 1 (observed): observation points are indicated by grey vertical dashed lines

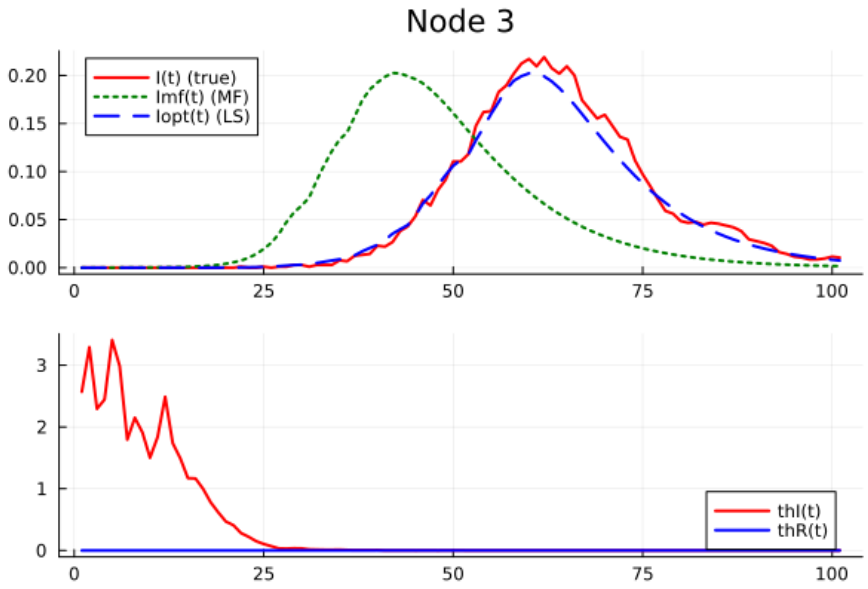

Figure 7.10: Evolution of node 3 (unobserved): despite the fact that we don't take information from the reference trajectory, the inference is well done.

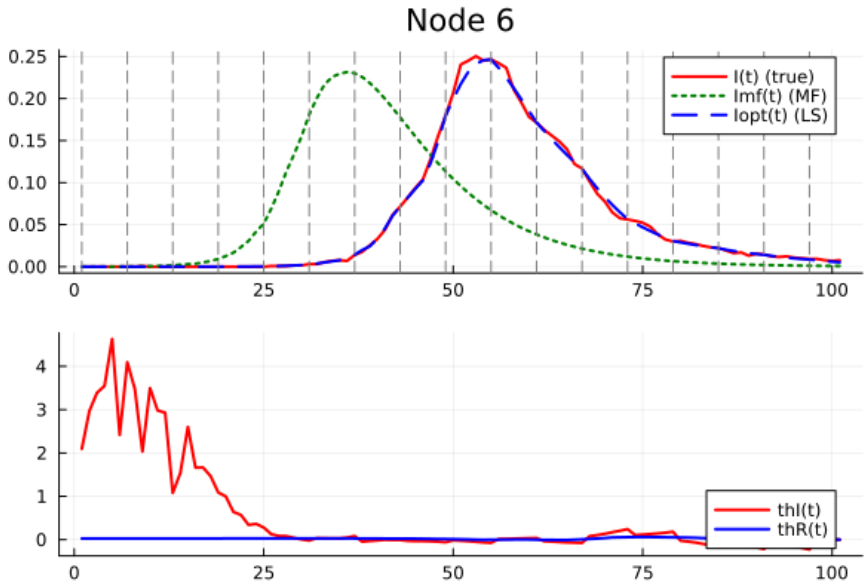

Figure 7.11: Evolution of node 6 (observed): we can see that the inference works well, as opposed to Figure 7.5.

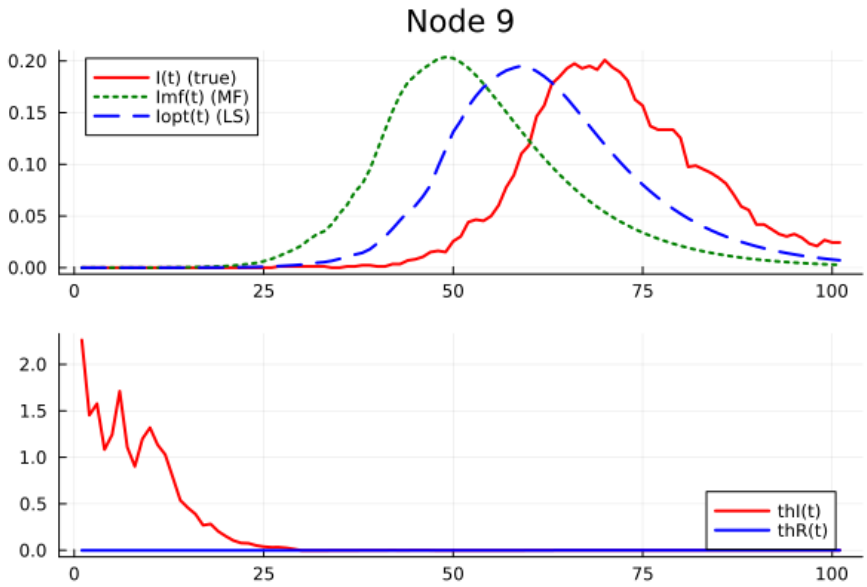

Figure 7.12: Evolution of node 9 (unobserved): in this case the inference places in the half between the mean-field trajectory and the MonteCarlo trajectory.

A more detailed analysis about risk assessment is addressed in the following section.

7.3 Risk-Assessment

In this section, we will focus on describing the capability of the inference algorithm in evaluating the epidemic risk in certain provinces. As already mentioned above, risk assessment consists in inferring the most probable epidemic outcome in some unobserved regions (to be precise, regions we are unable to obtain information about), knowing how the epidemic has evolved until that moment in other regions, especially if there is a good degree of correlation between unobserved and observed regions.
The risk assessment procedure follows these steps:

- **Sampling of the reference trajectories** through MonteCarlo simulation,
- **Capturing observations** for few nodes only,
- **Running the optimization algorithm** and observing the inferred trajectories, especially on unobserved nodes.

One of the results of this procedure is shown in Figures 7.9–7.12. As an additional experiment, we investigated how the correlation structure between nodes, encoded through the contact strength in the colocation matrix, affects the reliability of the risk assessment.
In particular, we observed that when the initial inference of a specific node was poor, increasing its contact strength with one of the observed nodes systematically improved the quality of the inference for that node. To test this effect, we performed a controlled study in which we increased the weight of the contact between the poorly observed node (node 7) (Fig. 7.13) and one of the observed nodes (node 8) (Fig. 7.14) by one order of magnitude.
We then evaluated the resulting changes in the inferred trajectories. The results showed a clear improvement in the stability and accuracy of the risk assessment in the unobserved node (Fig. 7.15) as its connectivity to observed nodes (Fig. 7.16) increased. This finding highlights the crucial role of network structure in propagating information during inference, and suggests that regions with weak observational data can still benefit from indirect observations if they are sufficiently connected to monitored areas.
These findings are well represented in Figures 7.13–7.16, where we compare the two aforementioned nodes before and after the weight increase. In addition, we show the associated colocation matrices for the two setups in Figures 7.17–7.18, focusing on one time step only. For readability purposes, we set the diagonal of the matrix (corresponding to self-colocation rates) to 0, since the diagonal is way larger than the other entries and it would be dominant in the heatmap, making hard to recognize changes in the magnitude of the modified entries.

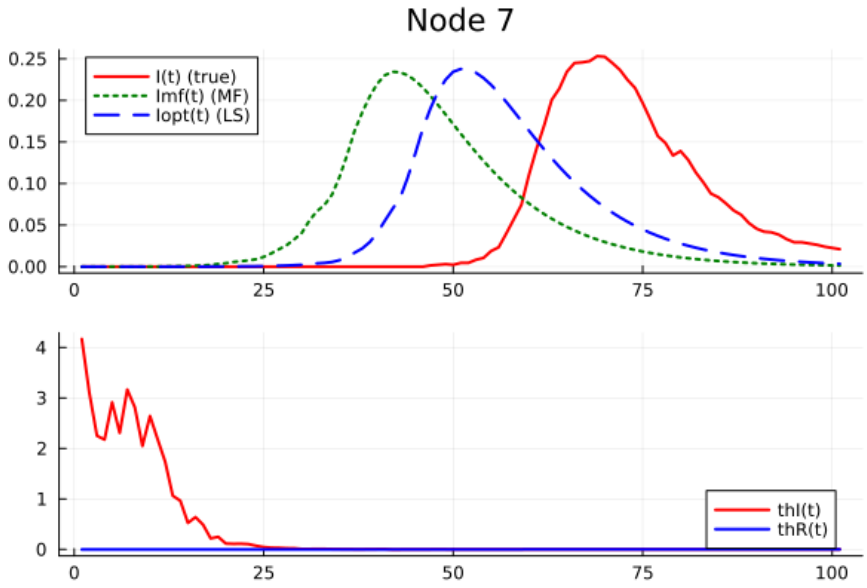

Figure 7.13: Evolution of node 7 before the increase of the connectivness between nodes 7 and 8: the inferred trajectory is quite distant from the reference MonteCarlo trajectory.

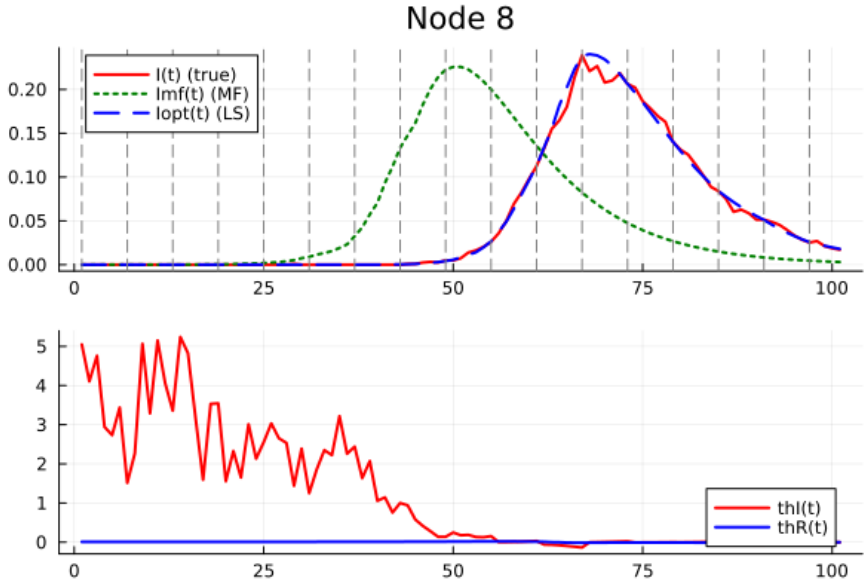

Figure 7.14: Evolution of node 8 before the increase of the connectivness between nodes 7 and 8: the inferred trajectory is consistent with the reference trajectory, since it is an observed node.

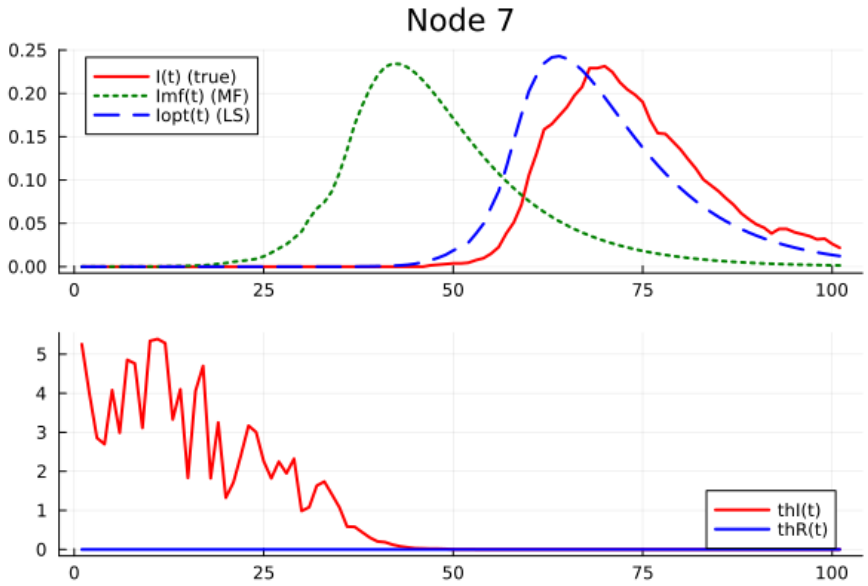

Figure 7.15: Evolution of node 7 after the increase of the connectivness between nodes 7 and 8: the inferred trajectory has systematically improved, significantly reducing the distance from the reference MonteCarlo trajectory.

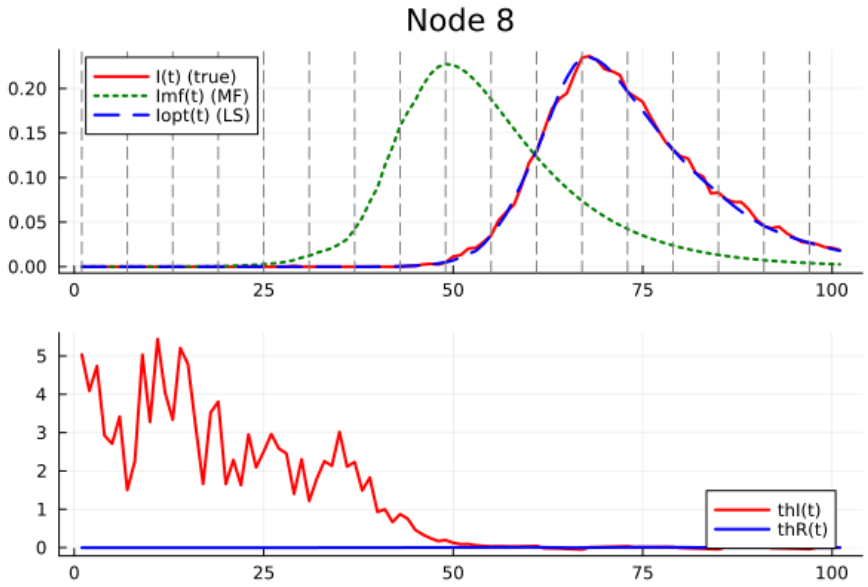

Figure 7.16: Evolution of node 8 after the increase of the connectivness between nodes 7 and 8: no significant changes are present.

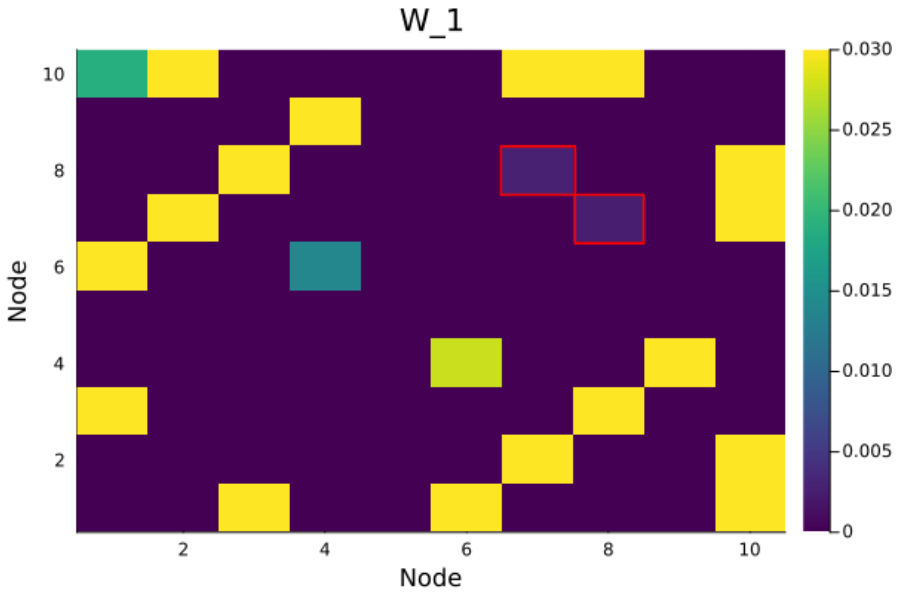

Figure 7.17: Heatmap for the colocation matrix at time $t = 1$ before the weight increase: marked cells (red contour) represent the matrix entries for connectivness between node 7 and 8.

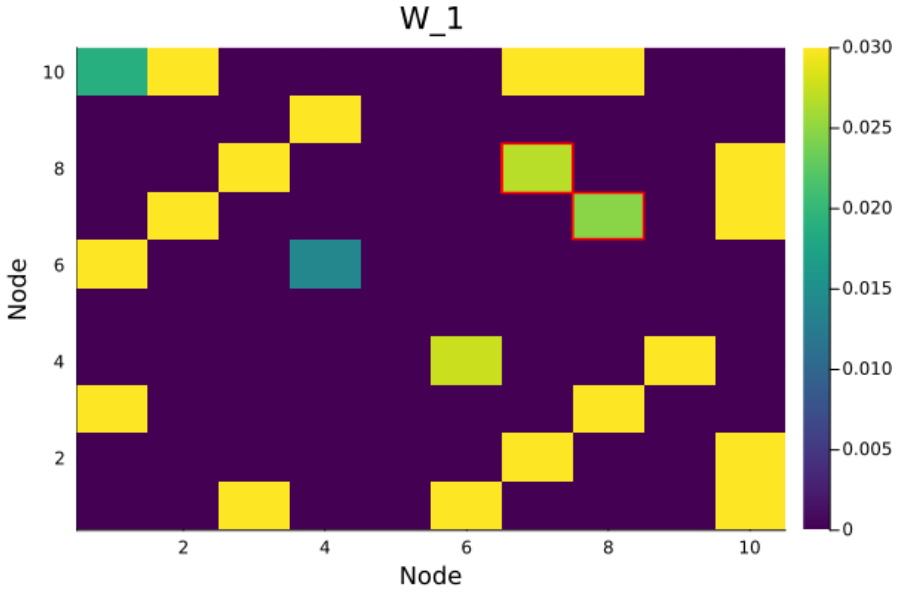

Figure 7.18: Heatmap for the colocation matrix at time $t = 1$ after the weight increase: the increase in contact strength is evident, highlighted by the change in color of the two involved cells.

7.3.1 Forward-Backward Optimization Algorithm vs. Monte Carlo Inference

To further quantify the accuracy of the inference algorithm, we computed the average $L2$ Norm (Fig. 7.19) between the inferred trajectories and the ground-truth Monte Carlo trajectories, across all time points and nodes. As a term of comparison, we also performed a naive Monte Carlo inference (Fig. 7.20) by simply simulating an ensemble of forward Monte Carlo trajectories, and computing again an average $L2$ Norm with respect to the ground-truth trajectories, measuring the distance of each of the sampled trajectories from the reference ones. In this way we are able to retrieve a distribution of the average distances. The setup is described in Section 6.2.2 and the results are shown in Figures 7.19–7.20, where we compare the average Norm for the inferred trajectories as a function of the percentage of observed nodes with the distribution of the Monte Carlo inference distances.

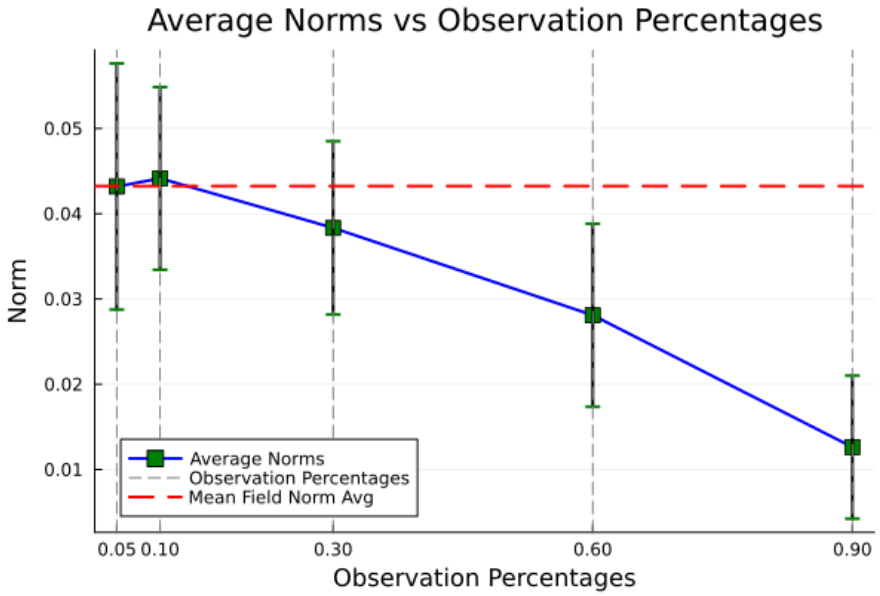

Figure 7.19: Average $L2$ Norm of the inferred trajectories w.r.t. the ground-truth trajectories: blue line with green squares represents the average norm values for each observation percentage, along with vertical green line around the mean values representing the standard deviation, gray dashed vertical lines indicate the considered observation percentages, red dashed horizontal line indicates the average $L2$ Norm between mean-field trajectory and the ground-truth trajectories.

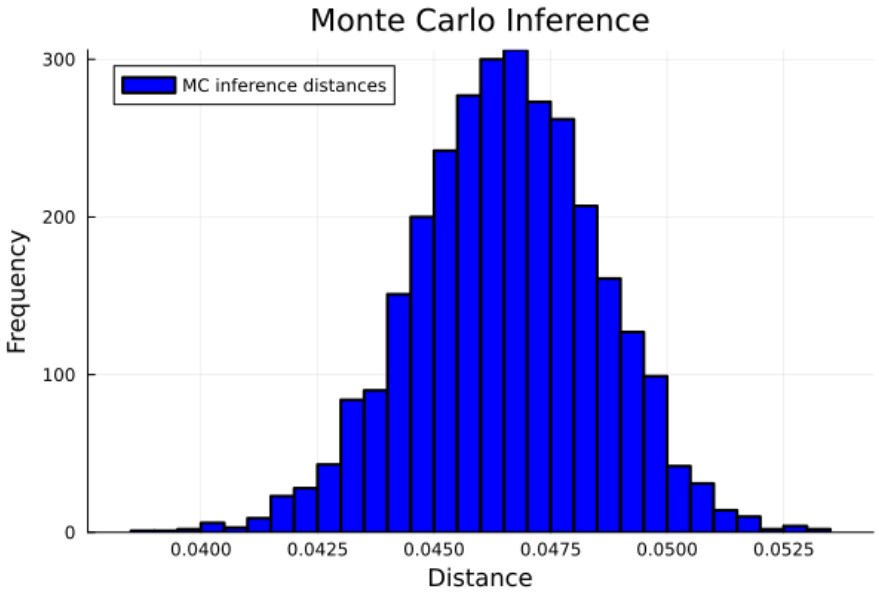

Figure 7.20: Histogram of the average $L2$ Norm of the Monte Carlo Inference trajectories w.r.t. the ground-truth trajectories: we have a distribution of the distances with most probable value ≈ 0.0465

One can see that with the 30% of observed nodes only, the average distance of the forward-backward optimization is considerably smaller than the most probable distance of the MC Inference. Indeed, we can say that the optimization algorithm places in the left tail of the MC distances distribution.

7.4 Real Colocation Data

In this section, we present the introduction in the model of real-world colocation data, which will affect the structure of the network. First, we will work with larger metapopulations, passing from $N \sim 10^3$ to $N \sim 10^6$.
Since in the limit of large populations the mean-field approximation holds, it is evident that the MonteCarlo trajectories are highly mean-field-like, and inference follows this trend too. Figures 7.21–7.23 show this concept, highlighting the fact that, since real-world colocation data present a huge difference in magnitude between intra-region colocation rates and inter-region colocation rates, trajectories tend to be quietly isolated, as if each metapopulation has a separate epidemic evolution.
This is also demonstrated by the fact that, if we do not have observations for a certain node, that node will mostly evolve according to the mean-field approximation, since information from other nodes will poorly affect the value of θs, due to the presence

of the corresponding colocation matrix entry in front of the term that accounts for the contribution of all neighboring nodes, as one can see in Equation (4.15).

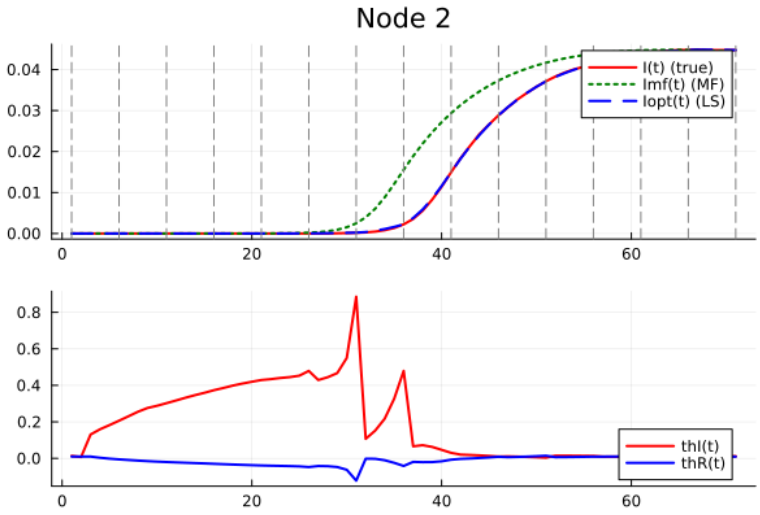

Figure 7.21: Evolution of node 2 (observed node): it is evident the superposition of the inferred and the ground-truth trajectories, due to a discrete number of observations;despite the slight advance of the mean-field approximation w.r.t. the ground-truth(red curve), the shape is the same, evidencing the typicality of the curves.

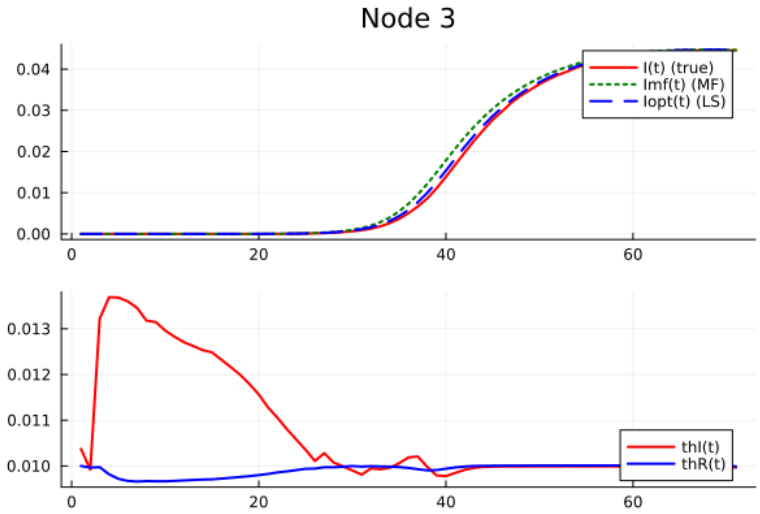

Figure 7.22: Evolution of node 3 (unobserved node): the three curves are almost overlapping.

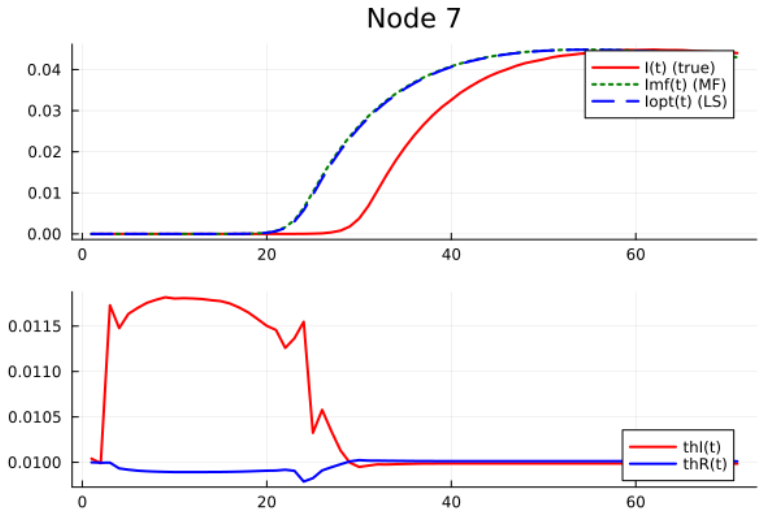

Figure 7.23: Evolution of node 7 (unobserved node): the inferred curve is almost overlapping with the mean-field approximation; the reason for this behavior is to be found again in the colocation matrix. Indeed, node 7 is the most self-connected node, producing a dynamic almost separate from the others. See Map 7.24

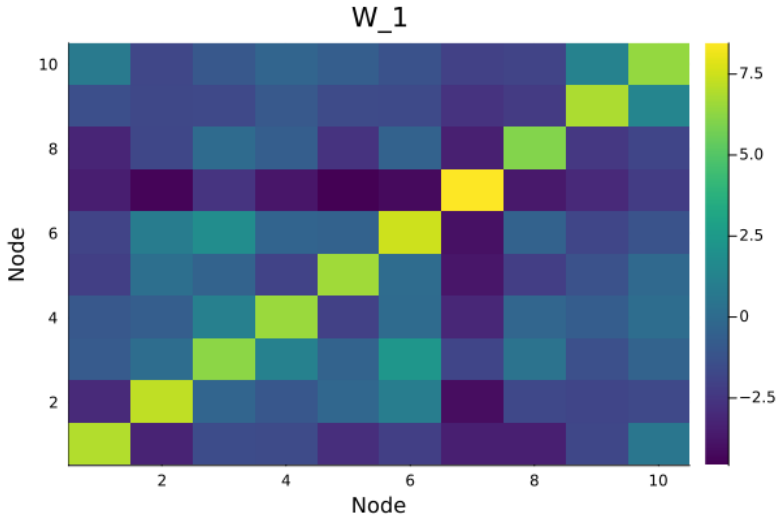

Figure 7.24: Heatmap of the colocation matrix at time step $t = 1$ in log-scale: node 7 is the most self-connected node and also poorly connected with other nodes.

7.5 Larger Number of Metapopulations

7.5.1 Network with 20 Nodes

We go back to the analysis through the fictitious colocation matrix. We tried to increase the number of metapopulations involved in the study, raising M from 10 to 20. The dynamics mostly follows the trend of the smaller case, including atypical behavior of some nodes, which will now be more than one. However, the overall inference is quite good and useful to understand valuable insights into the spread of the network epidemic, as one can see in Figures 7.25–7.27, where it is evident that there are again some criticalities in the behavior of certain nodes, that can be linked to some local minimum.

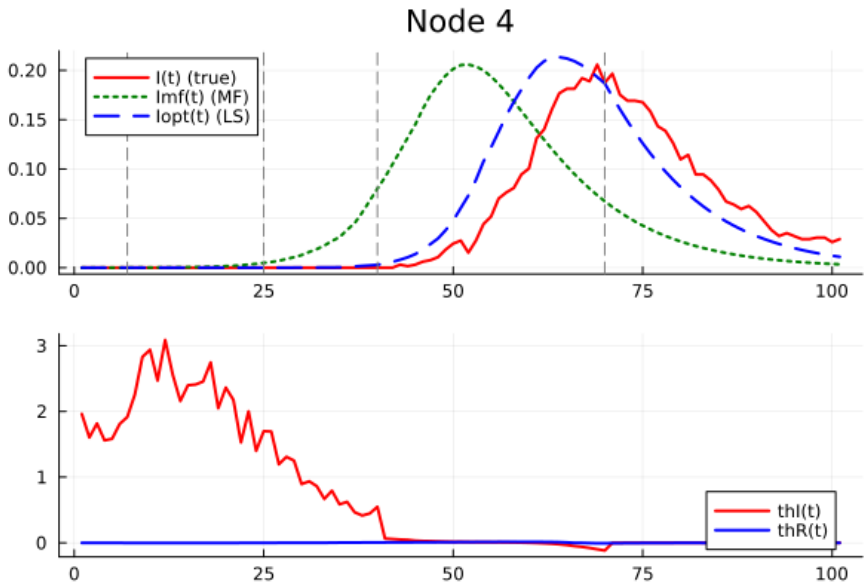

Figure 7.25: Evolution of node 4: the inferred trajectory is consistent with the ground-truth, despite the advance of the mean-field approximation.

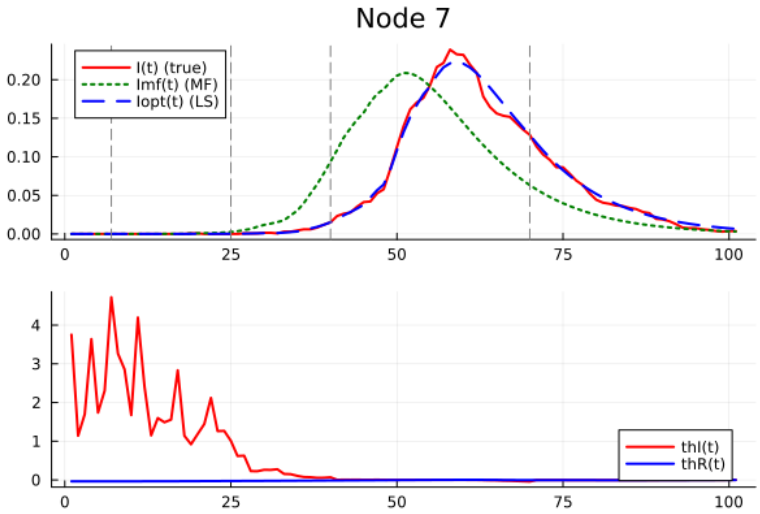

Figure 7.26: Evolution of node 7: again, the inference is consistent.

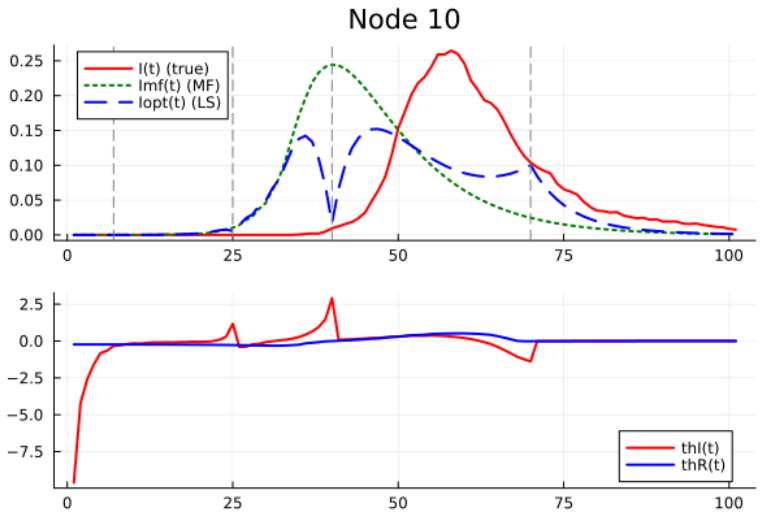

Figure 7.27: Evolution of node 10: it is evident the discrepancy, along with the inconsistency of the curve; the reason is to be found in bad conditioning (poorly informative observations), in the atypicality of the ground-truth, and in some local minimum.

7.5.2 Network with 30 Nodes

We further increased M to 30 nodes, and the trend is reproduced, as one can see in Figures 7.28–7.29. Even in this case, few nodes show strange behavior, which can be managed by increasing the number of observations.

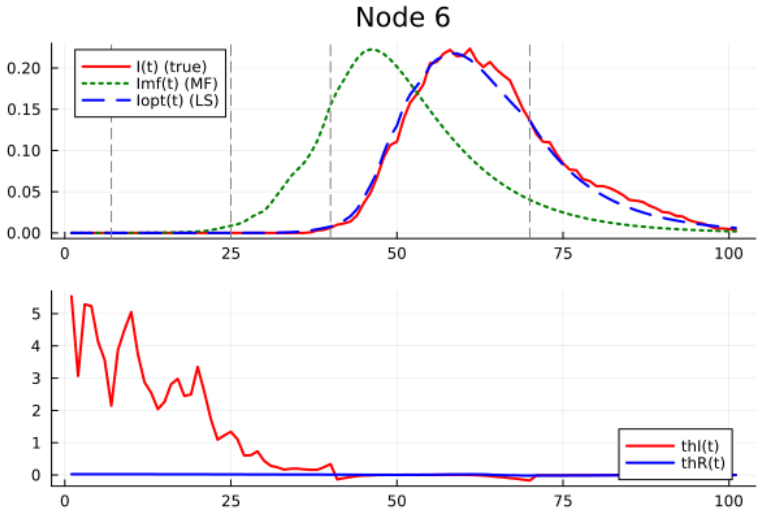

Figure 7.28: Evolution of node 6 in a Network with $M = 30$ nodes.

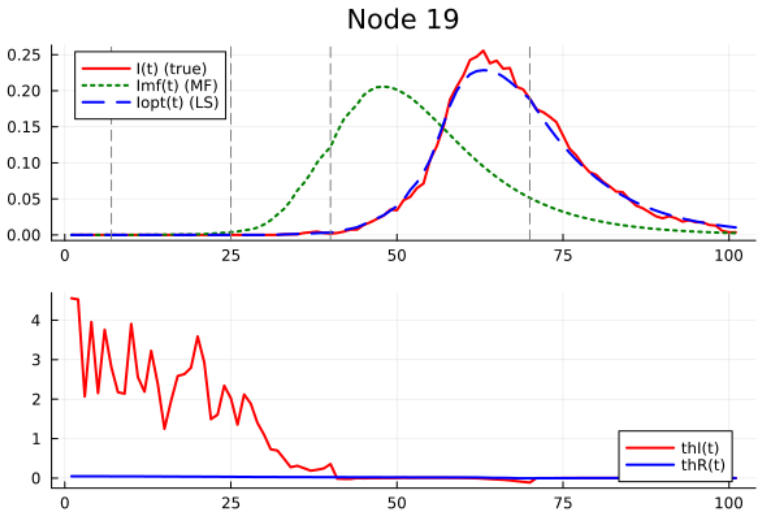

Figure 7.29: Evolution of node 19 in a Network with $M = 30$ nodes.

7.6 Scalability of The Algorithm

One crucial aspect of the proposed inference algorithm is its scalability with the number of nodes that constitute the network under consideration. In fact, while the present study considered only networks with a discrete number of nodes (up to 30 but concentrating on 10 nodes networks), more realistic applications involve hundreds or thousands of subpopulations, based on geographical resolution.
Analyzing the time profile of computations, we noticed that the bottleneck is located in the **ReverseDiff gradient** function, which in turn depends on the scaling of the loss function. A statistical analysis on the scalability of the model has been performed, highlighting the fact that the algorithm can be further improved (see Figures 7.30–7.31).

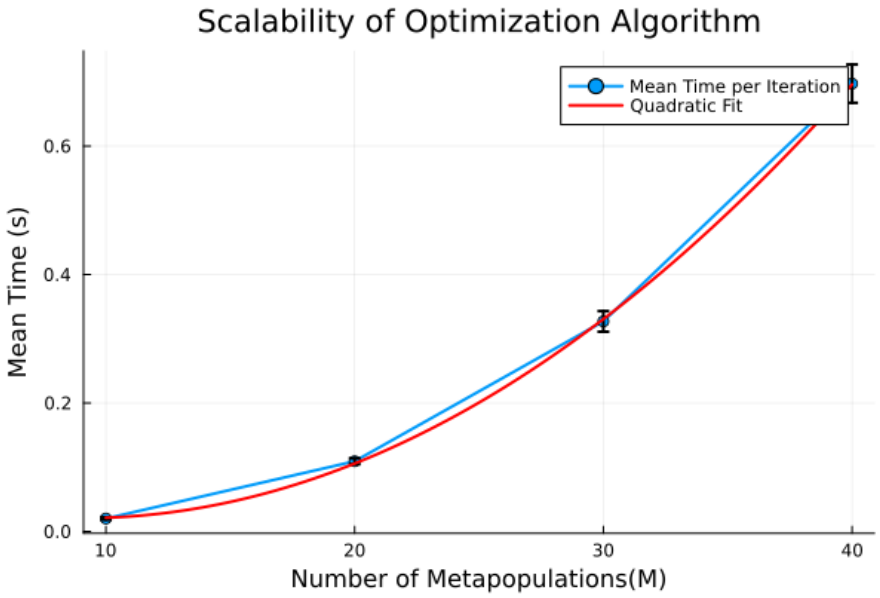

Figure 7.30: Quadratic fit of the scalability curve: blue line represents the curve obtained by considering the mean over 5 simulations of the computational time per iteration, red curve represents the quadratic polynomial which better reproduce the blue line.

The two polynomials seem to equally reproduce the real curve, but from the analysis of **RSS (Residual Sum of Squares** it emerges that the cubic fit is more precise, showing that the algorithm has computational complexity $O(M^3)$.

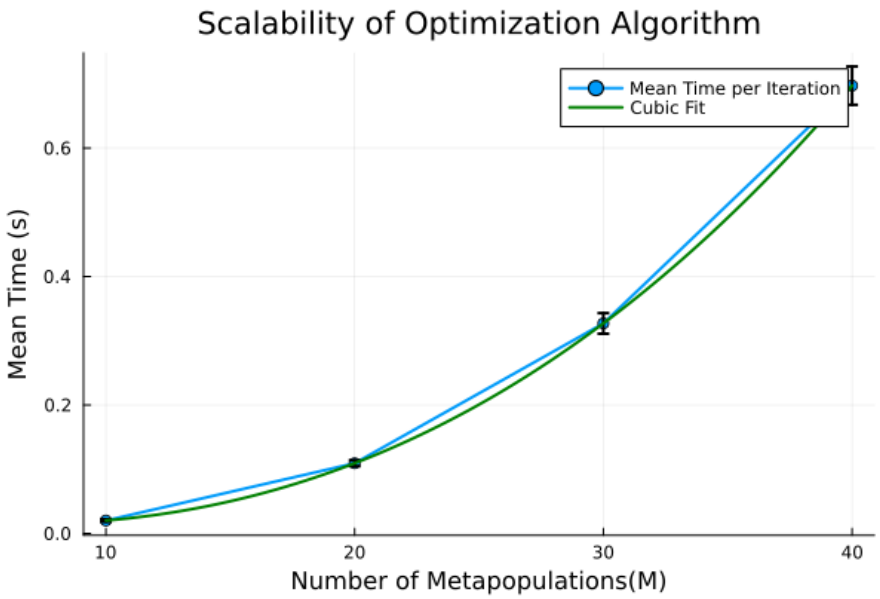

Figure 7.31: Cubic fit of the scalability curve: same setup as above; green line is the cubic polynomial which better approximate the blue line.

Chapter 8

Conclusions and Future Perspectives

8.1 Conclusions

This thesis has developed a novel inference framework for spatially structured epidemic models, combining a stochastic SEIR metapopulation description with a path-integral formulation inspired by statistical physics. Through a saddle-point approximation, we derived forward-backward equations that allow the reconstruction of the most probable epidemic trajectories consistent with sparse and noisy observations.
The implementation has been performed first via a forward-backward algorithm, then, due to numerical instability, via a gradient-based optimization through the Julia library **NLopt.jl**. The choice to rely on optimization has reduced the numerical instability of the first method, but has paid the price of increasing the computational time.
The implementation demonstrated the ability to reconstruct epidemic trajectories with good accuracy on synthetic data, provided that a reasonable prior and appropriate initial conditions were chosen. The experiments showed that the method can recover key epidemic indicators such as infection curves, the epidemic peak, and the possibility to evaluate risk on unobserved nodes in most scenarios.
Moreover, we extended the experiments to real colocation data provided by Meta (Data for Good), applying the framework to mobility patterns derived from actual user movements. These tests confirmed the framework's feasibility, but also revealed that the epidemic dynamics on the real-world colocation network turns out to be rather well described by deterministic single-population mean-field-like trajectories, due to the high level of mixing and the relative homogeneity of contacts encoded in the empirical colocation matrix. As a consequence, the gain from a spatially

resolved inference was moderate, suggesting that in highly connected mobility networks, mean-field-like approximations may still be a reasonable first-order model. The study also revealed several limitations and criticalities:

- a strong sensitivity to the choice of initialization, especially for the seed probabilities q_i;
- the risk of local minima in the optimization landscape, particularly in more complex or atypical epidemic trajectories;
- scalability challenges when the network size grows, due to the increased number of variables and the difficulty of maintaining stable convergence.

Nevertheless, the proposed methodology provides a promising direction for epidemic inference, as it brings together rigorous stochastic modeling and a principled Bayesian framework.

8.2 Future Perspectives

Several directions for further research emerge from this work, laying the foundation for important improvements:

- **Refined integration of real data**: explore ways to preprocess and cluster the colocation data to enhance spatial heterogeneity, for example grouping together nodes (metapopulations) with similar behavior and colocation patterns in such a way to have a reduced graph composed by "supernodes", making the inference more informative than a simple mean-field approximation;
- **Advanced optimization strategies**: adopt global optimization to reduce local minima issues;
- **Regularization of conjugate fields**: try to reduce the noise around the θ-variables for a better estimate of the correct behavior if the system, including patient zero inference;
- **Increase robustness of the forward-backward algorithm**: design improved numerical schemes to make the forward–backward propagation more stable, allowing it to be reused effectively in future implementations;
- **Joint parameter inference**: extend the framework to estimate epidemiological parameters (β, η, μ) together with state trajectories, since in the actual framework they are fixed. In particular, one can implement a Markov Chain inference based on the action functional $\mathcal{S}$ as a function of the parameters: a

set of parameters is chosen, the action is evaluated, and eventually the parameters are accepted or rejected. The procedure is the same as **MCMC (Monte Carlo Markov Chain)**, except that we don't have to sample thousands of trajectories, but we evaluate the action of the system;

- **Patient zero identification**: further develop techniques to infer the epidemic seed location (patient zero) in a robust way, since initial experiments showed this task to be more challenging than expected, due to the fact that θ-variables are very noisy and the seed probability q_i is strongly dependent on the value of θ^I at time $t = 1$, leading us to provide the seed as an input for now;
- **Real epidemic applications**: apply the method, once it is further improved, to validated epidemiological datasets with confirmed case counts, to support public health decision-making;
- **Computational scalability**: investigate approximate representations or relax some constraints to speed up computations and optimization and, by consequence, handle networks with a larger number of nodes efficiently; explore an efficient way to compute a manual gradient to replace the automatic differentiator **ReverseDiff.jl**, which is the one responsible for the high computational cost.

In conclusion, while several challenges remain in terms of robustness, scalability, and the integration of real-world data, this thesis demonstrates the feasibility of a principled, well-posed, and physics-inspired framework for epidemic inference. Its further improvement may contribute to more effective monitoring and control of future infectious disease outbreaks, opening the way for data-driven, adaptive public health strategies.

Bibliography

[1] William Ogilvy Kermack and Anderson G McKendrick. «A contribution to the mathematical theory of epidemics». In: *Proceedings of the Royal Society of London. Series A, Containing papers of a mathematical and physical character* 115.772 (1927), pp. 700–721 (cit. on pp. 1, 9, 10).

[2] Tom Britton and Federica Giardina. «Introduction to statistical inference for infectious diseases». In: *Journal de la société française de statistique* 157.1 (2016), pp. 53–70. URL: https://www.numdam.org/item/JSFS_2016__157_1_53_0/ (cit. on p. 1).

[3] Jian-Xin Pan and Kai-Tai Fang. «Maximum Likelihood Estimation». In: *Growth Curve Models and Statistical Diagnostics*. New York, NY: Springer New York, 2002, pp. 77–158. ISBN: 978-0-387-21812-0. DOI: 10.1007/978-0-387-21812-0_3. URL: https://doi.org/10.1007/978-0-387-21812-0_3 (cit. on p. 1).

[4] Udo von Toussaint. «Bayesian inference in physics». In: *Rev. Mod. Phys.* 83 (3 Sept. 2011), pp. 943–999. DOI: 10.1103/RevModPhys.83.943. URL: https://link.aps.org/doi/10.1103/RevModPhys.83.943 (cit. on p. 1).

[5] Alfredo Braunstein. «Notes for the class of Algorithms for Optimization,Inference and Learning». Notes distributed during the course, unpublished. 2024 (cit. on p. 1).

[6] Christos Karras, Aristeidis Karras, Markos Avlonitis, and Spyros Sioutas. «An Overview of MCMC Methods: From Theory to Applications». In: *Artificial Intelligence Applications and Innovations. AIAI 2022 IFIP WG 12.5 International Workshops*. Ed. by Ilias Maglogiannis, Lazaros Iliadis, John Macintyre, and Paulo Cortez. Cham: Springer International Publishing, 2022, pp. 319–332. ISBN: 978-3-031-08341-9 (cit. on p. 1).

[7] A. S. Talawar and U. R. Aundhakar. «Parameter Estimation of SIR Epidemic Model Using MCMC Methods». In: *Global Journal of Pure and Applied Mathematics* 12.2 (2016), pp. 1299–1306. ISSN: 0973-1768. URL: http://www.ripublication.com (cit. on p. 2).

[8] Joseph Dureau, Konstantinos Kalogeropoulos, and Marc Baguelin. «Capturing the time-varying drivers of an epidemic using stochastic dynamical systems». In: *Biostatistics* 14.3 (2013), pp. 541–555. DOI: 10.1093/biostatistics/kxs052 (cit. on p. 2).

[9] Luís M. A. Bettencourt and Ruy M. Ribeiro. «Real Time Bayesian Estimation of the Epidemic Potential of Emerging Infectious Diseases». In: *PLoS ONE* 3.5 (2008), e2185. DOI: 10.1371/journal.pone.0002185 (cit. on pp. 2, 16).

[10] S. Flaxman et al. «Estimating the effects of non-pharmaceutical interventions on COVID-19 in Europe». In: *Nature* 584.7820 (2020), pp. 257–261. DOI: 10.1038/s41586-020-2405-7 (cit. on pp. 2, 16).

[11] C. Fritz, E. Dorigatti, and D. Rügamer. «Combining graph neural networks and spatio-temporal disease models to improve the prediction of weekly COVID-19 cases in Germany». In: *Scientific Reports* 12 (2022), p. 3930. DOI: 10.1038/s41598-022-07757-5 (cit. on p. 2).

[12] K. Song, H. Park, J. Lee, and et al. «COVID-19 infection inference with graph neural networks». In: *Scientific Reports* 13 (2023), p. 11469. DOI: 10.1038/s41598-023-38314-3 (cit. on p. 2).

[13] Chintan Shah, Nima Dehmamy, Nicola Perra, Matteo Chinazzi, Albert-László Barabási, Alessandro Vespignani, and Rose Yu. *Finding Patient Zero: Learning Contagion Source with Graph Neural Networks*. 2020. arXiv: 2006.11913 [cs.SI]. URL: https://arxiv.org/abs/2006.11913 (cit. on p. 2).

[14] Angelika Steger and Nicholas C. Wormald. «Generating random regular graphs quickly». In: *Proceedings of the 6th International Conference on Graph Theory* (1999), pp. 239–249. DOI: 10.1145/780542.780576 (cit. on p. 7).

[15] Paul Erdős and Alfréd Rényi. «On Random Graphs I». In: *Publicationes Mathematicae* 6 (1959), pp. 290–297. URL: https://snap.stanford.edu/class/cs224w-readings/erdos59random.pdf (cit. on p. 8).

[16] GeeksforGeeks. *Erdos Renyi Model - Generating Random Graphs*. Accessed: 2025-05-18. 2023. URL: https://www.geeksforgeeks.org/erdos-renyl-model-generating-random-graphs/ (cit. on p. 9).

[17] Vittoria Colizza, Romualdo Pastor-Satorras, and Alessandro Vespignani. «Reaction–diffusion processes and metapopulation models in heterogeneous networks». In: *Nature Physics* 3.4 (2007), pp. 276–282 (cit. on pp. 9, 13).

[18] Herbert W. Hethcote. «The Mathematics of Infectious Diseases». In: *SIAM Review* 42.4 (2000), pp. 599–653. DOI: 10.1137/S0036144500371907 (cit. on pp. 10, 12).

[19] Roy M. Anderson and Robert M. May. *Infectious Diseases of Humans: Dynamics and Control*. Oxford University Press, 1991 (cit. on p. 11).

[20] Matt J. Keeling and Pejman Rohani. *Modeling Infectious Diseases in Humans and Animals*. Princeton University Press, 2008 (cit. on p. 11).

[21] Maia Martcheva. *An Introduction to Mathematical Epidemiology*. Vol. 61. Texts in Applied Mathematics. Springer, 2015. DOI: 10.1007/978-1-4899-7612-3 (cit. on p. 11).

[22] Linda JS Allen. «An Introduction to Stochastic Epidemic Models». In: *In Mathematical Epidemiology* (2008), pp. 81–130. DOI: 10.1007/978-3-540-78911-6_3 (cit. on p. 12).

[23] Vitaly Belik, Theo Geisel, and Dirk Brockmann. «Natural human mobility patterns and spatial spread of infectious diseases». In: *Physical Review X* 1.1 (2011), p. 011001. DOI: 10.1103/PhysRevX.1.011001 (cit. on p. 13).

[24] Richard P. Feynman and Albert R. Hibbs. *Quantum Mechanics and Path Integrals*. McGraw-Hill, 1965 (cit. on p. 19).

[25] P. C. Martin, E. D. Siggia, and H. A. Rose. «Statistical Dynamics of Classical Systems». In: *Phys. Rev. A* 8 (1 July 1973), pp. 423–437. DOI: 10.1103/PhysRevA.8.423. URL: https://link.aps.org/doi/10.1103/PhysRevA.8.423 (cit. on pp. 19, 40).

[26] Luca Dall'Asta. «Notes for the class of Field Theory and Critical Phenomena». Notes distributed during the course, unpublished. 2023 (cit. on p. 19).

[27] Jozef Strecka and Michal Jascur. *A brief account of the Ising and Ising-like models: Mean-field, effective-field and exact results*. 2015. arXiv: 1511.03031 [cond-mat.stat-mech]. URL: https://arxiv.org/abs/1511.03031 (cit. on p. 26).

[28] Pierre Weiss. «L'hypothèse du champ moléculaire et la propriété ferromagnétique». In: *J. Phys. Theor. Appl.* 6.1 (1907), pp. 661–690 (cit. on p. 26).

[29] E. T. Jaynes. «Information Theory and Statistical Mechanics». In: *Phys. Rev.* 106 (4 May 1957), pp. 620–630. DOI: 10.1103/PhysRev.106.620. URL: https://link.aps.org/doi/10.1103/PhysRev.106.620 (cit. on p. 26).

[30] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. «An introduction to variational methods for graphical models». In: *Machine Learning* 37.2 (1999), pp. 183–233. DOI: 10.1023/A:1007665907178 (cit. on p. 27).

[31] J.S. Yedidia, W.T. Freeman, and Y. Weiss. «Constructing free-energy approximations and generalized belief propagation algorithms». In: *IEEE Transactions on Information Theory* 51.7 (2005), pp. 2282–2312. DOI: 10.1109/TIT.2005.850085 (cit. on p. 27).

[32] Dalton A. R. Sakthivadivel. «Magnetisation and Mean Field Theory in the Ising Model». In: *SciPost Phys. Lect. Notes* 35 (2022). DOI: 10.21468/SciPostPhysLectNotes.35. URL: https://scipost.org/10.21468/SciPostPhysLectNotes.35 (cit. on p. 27).

[33] Fabio Altarelli, Alfredo Braunstein, Luca Dall'Asta, and Riccardo Zecchina. «Bayesian inference of epidemics on networks via belief propagation». In: *Physical Review Letters* 112.11 (2014), p. 118701. DOI: 10.1103/PhysRevLett.112.118701 (cit. on p. 30).

[34] Erik Aurell and Kim Sneppen. «Epigenetics as a first exit problem». In: *Physical Review Letters* 88.4 (2002), p. 048101. DOI: 10.1103/PhysRevLett.88.048101 (cit. on p. 31).

[35] Steven Gratton. *Path Integral Approach to Uncertainties in SIR-type Systems*. 2020. arXiv: 2006.01817 [q-bio.PE]. URL: https://arxiv.org/abs/2006.01817 (cit. on p. 31).

[36] Meta Data for Good. *Facebook Data for Good: Colocation Maps*. 2021. URL: https://dataforgood.facebook.com/dfg/tools/colocation-maps (cit. on p. 41).

[37] Thomas House. «Modelling epidemics on networks». In: *Contemporary Physics* 53.3 (2012), pp. 213–225 (cit. on p. 42).

[38] Jessica Dimka, Carolyn Orbann, and Lisa Sattenspiel. «Applications of agent-based modelling techniques to studies of historical epidemics: The 1918 flu in newfoundland and labrador». In: *Journal of the Canadian Historical Association* 25.2 (2014), pp. 265–296 (cit. on p. 42).

[39] Stephen Eubank, Hasan Guclu, VS Anil Kumar, Madhav V Marathe, Aravind Srinivasan, Zoltan Toroczkai, and Nan Wang. «Modelling disease outbreaks in realistic urban social networks». In: *Nature* 429.6988 (2004), pp. 180–184 (cit. on p. 42).

[40] Shankar Iyer et al. «Large-scale measurement of aggregate human colocation patterns for epidemiological modeling». In: *Epidemics* 42 (2023), p. 100663 (cit. on pp. 42, 46).

[41] Clément Dubost, David Dupré, et al. «Predictive performance of network-based metrics derived from mobile colocation data for forecasting COVID-19 incidence in France». In: *International Journal of Infectious Diseases* 112 (2021), pp. 146–155. DOI: 10.1016/j.ijid.2021.09.031 (cit. on p. 43).

[42] Serina Chang, Emma Pierson, Benjamin PW Koh, et al. «Mobility network modeling explains higher SARS-CoV-2 infection rates among disadvantaged groups and informs reopening strategies». In: *Nature* 589.7840 (2021), pp. 82–87. DOI: 10.1038/s41586-020-2923-3 (cit. on p. 43).

[43] Ramin Mohammadi, Dhruv Tiwari, Srinivasan Venkatramanan, et al. «WiFi mobility models for predicting COVID-19 cases on a university campus». In: *arXiv preprint* (2022). arXiv: `2201.10641 [cs.SI]` (cit. on p. 43).

[44] Jarrett Revels, Miles Lubin, and Theodore Papamarkou. «Forward-mode automatic differentiation in Julia». In: *arXiv preprint arXiv:1607.07892* (2016) (cit. on p. 63).

[45] Michael Innes. «Don't unroll adjoint: Differentiating SSA-form programs». In: *arXiv preprint arXiv:1810.07951* (2018) (cit. on p. 64).

[46] Jorge Nocedal. «Updating quasi-Newton matrices with limited storage». In: *Mathematics of computation* 35.151 (1980), pp. 773–782 (cit. on p. 64).

[47] Sebastian Ruder. «An overview of gradient descent optimization algorithms». In: *arXiv preprint arXiv:1609.04747* (2016) (cit. on p. 64).