Bayesian Inference and Modeling for Point Processes with Applications From Neuronal Activity to Scene Viewing

Noa Malem

Dissertation

zur Erlangung des akademischen Grades
“doctor rerum naturalium” (Dr. rer. nat.)
in der Wissenschaftsdisziplin “Numerische Mathematik”

eingereicht an der
Mathematisch-Naturwissenschaftlichen Fakultät
Institut für Mathematik der Universität Potsdam

November 6th, 2023

1. Reviewer Prof. Dr. Sebastian Reich
2. Reviewer Prof. Dr. Manfred Opper
3. Reviewer Prof. Dr. Simon Barthelme
Supervisors Prof. Dr. Sebastian Reichand Prof. Dr. Ralf Engbert

Published online on the
Publication Server of the University of Potsdam:
https://doi.org/10.25932/publishup-61495
https://nbn-resolving.org/urn:nbn:de:kobv:517-opus4-614952

Abstract

Point processes are a common methodology to model sets of events. From earthquakes to social media posts, from the arrival times of neuronal spikes to the timing of crimes, from stock prices to disease spreading – these phenomena can be reduced to the occurrences of events concentrated in points. Often, these events happen one after the other defining a time–series.

Models of point processes can be used to deepen our understanding of such events and for classification and prediction. Such models include an underlying random process that generates the events. This work uses Bayesian methodology to infer the underlying generative process from observed data. Our contribution is twofold – we develop new models and new inference methods for these processes.

We propose a model that extends the family of point processes where the occurrence of an event depends on the previous events. This family is known as Hawkes processes. Whereas in most existing models of such processes, past events are assumed to have only an excitatory effect on future events, we focus on the newly developed nonlinear Hawkes process, where past events could have excitatory and inhibitory effects. After defining the model, we present its inference method and apply it to data from different fields, among others, to neuronal activity.

The second model described in the thesis concerns a specific instance of point processes — the decision process underlying human gaze control. This process results in a series of fixated locations in an image. We developed a new model to describe this process, motivated by the known Exploration–Exploitation dilemma. Alongside the model, we present a Bayesian inference algorithm to infer the model parameters.

Remaining in the realm of human scene viewing, we identify the lack of best practices for Bayesian inference in this field. We survey four popular algorithms and compare their performances for parameter inference in two scan path models.

The novel models and inference algorithms presented in this dissertation enrich the understanding of point process data and allow us to uncover meaningful insights.

Zusammenfassung

Punktprozesse sind eine gängige Methode zur Modellierung von Ereignismengen. Von Erdbeben bis zu Social-Media-Posts, von den neuronalen Spikes bis zum Zeitpunkt von Verbrechen, von Aktienkursen bis zur Ausbreitung von Krankheiten - diese Phänomene lassen sich auf das Auftreten von Ereignissen reduzieren, die in Punkten konzentriert sind. Häufig treten diese Ereignisse nacheinander auf und bilden eine Zeitreihe.

Modelle von Punktprozessen können verwendet werden, um unser Verständnis solcher Ereignisse für Klassifizierung und Vorhersage zu vertiefen. Solche Modelle umfassen einen zugrunde liegenden Zufallsprozess, der die Ereignisse erzeugt. In dieser Arbeit wird die Bayes'sche Methodik verwendet, um den zugrunde liegenden generativen Prozess aus den beobachteten Daten abzuleiten. Wir leisten einen doppelten Beitrag: Wir entwickeln neue Modelle und neue Inferenzmethoden für diese Prozesse.

Wir schlagen ein Modell vor, das die Familie der Punktprozesse erweitert, bei denen das Auftreten eines Ereignisses von den vorherigen Ereignissen abhängt. Diese Familie ist als Hawkes-Prozesse bekannt. Während in den meisten bestehenden Modellen solcher Prozesse davon ausgegangen wird, dass vergangene Ereignisse nur eine exzitatorische Wirkung auf zukünftige Ereignisse haben, konzentrieren wir uns auf den neu entwickelten nichtlinearen Hawkes-Prozess, bei dem vergangene Ereignisse exzitatorische und hemmende Wirkungen haben können. Nach der Definition des Modells stellen wir seine Inferenzmethode vor und wenden sie auf Daten aus verschiedenen Bereichen an, unter anderem auf die neuronale Aktivität.

Das zweite Modell, das in dieser Arbeit beschrieben wird, betrifft einen speziellen Fall von Punktprozessen - den Entscheidungsprozess, der der menschlichen Blicksteuerung zugrunde liegt. Dieser Prozess führt zu einer Reihe von fixierten Positionen in einem Bild. Wir haben ein neues Modell entwickelt, um diesen Prozess zu beschreiben, motiviert durch das bekannte Exploration-Exploitation-Dilemma. Neben dem Modell stellen wir einen Bayes'schen Inferenzalgorithmus vor, um die Modellparameter abzuleiten.

Wir bleiben auf dem Gebiet der menschlichen Szenenbetrachtung und stellen fest, dass es in diesem Bereich keine bewährten Verfahren für die Bayes'sche Inferenz gibt. Wir geben einen Überblick über vier gängige Algorithmen und vergleichen ihre Leistungen bei der Ableitung von Parametern für zwei Scanpfadmodelle.

Die in dieser Dissertation vorgestellten neuen Modelle und Inferenzalgorithmen bereichern das Verständnis von Punktprozessdaten und ermöglichen es uns, sinnvolle Erkenntnisse zu gewinnen.

Acknowledgements

Completing this dissertation is a significant milestone in my academic and personal journey. I am deeply grateful to the people who have supported me throughout this, and I dedicate this page to express my deep gratitude to them.

First and foremost, I would like to thank my supervisor Sebastian Reich. Your expertise, critical feedback, and encouragement have been instrumental in shaping my research and ensuring the quality of this dissertation. My sincere gratitude also goes to my co-supervisor Ralf Engbert for his guidance and support in applying my research to dynamic eye movement models. I would like to express my appreciation for the support and guidance of my mentor Manfred Opper. Working with you for the past years has been a fantastic opportunity and always a great pleasure.

I want to thank my colleagues who accompanied me through my PhD. To César and Chrisitan, for the scientific discussion and active support of my research. To my office mates Christian, Sahani, and Paul, who were my motivation to come to our office even in the age of home office. To the past and present members of AG Reich, Jana, Maria, David, Jakiw, Maia, Jin, and Diksha. Thank you for the coffee breaks and always bringing the best snacks from conferences.

I have been lucky to conduct my research as part of the collaborative research center 1294 for data assimilation. It provided many excellent opportunities to collaborate with and be inspired by other scientists from adjacent projects—particularly Lisa, who has been a fantastic project–buddy and a great friend.

I would like to thank my friends who were there through the ups and down. Thank you for listening to my complaints and celebrating my achievements with me. I could not have completed this journey without you.

Finally, I am genuinely thankful for my family. For my mom, thank you for always being my default support system and reminding me to stay curious. For Malte, your love, support, and patience have been a constant source of comfort and motivation during the challenging moments. Last but not least, for Ella, for teaching me everything that is really important in life.

Publications

Parts of this dissertation have been published in the following original articles:

Chapter 3:

Noa Malem-Shinitski, César Ojeda, and Manfred Opper. „Variational Bayesian Inference for Nonlinear Hawkes Process With Gaussian Process Self-Effects“. In: *Entropy* 24.3 (2022), p. 356

Chapter 4:

Noa Malem-Shinitski, Manfred Opper, Sebastian Reich, Lisa Schwetlick, Stefan A Seelig, and Ralf Engbert. „A Mathematical Model of Local and Global Attention in Natural Scene Viewing“. In: *PLOS Computational Biology* 16.12 (2020), e1007880

The general research concepts were formulated, and ongoing discussions were held with all co-authors. The author of this dissertation developed the specific research objectives and methodologies, performed formal analysis, created the software implementation, and validated the results herself. The co-authors thoroughly reviewed the articles.

All co–authors of these articles agreed to use content, figures, and results from the articles above for this dissertation.

Contents

List of Figures

List of Tables

Introduction

1

1.1 Modeling

Most people go through their days without thinking even once about the concept of *modeling*. Nevertheless, modeling is a crucial feature of human cognitive processing. The term *mental modeling* first appeared in 1943 in a book by the Scottish psychologist and philosopher Kenneth Craik [33]. The term refers to the hypothesis that the brain creates "small scale models" of reality. These models are then used to form predictions about the world, shape behavior and solve problems.

Some people, primarily keen on numbers and computers, spend some of their days actively thinking about modeling. They look at the world around them and seek to create an explicit model explaining or predicting it. These models usually take the form of mathematical equations (mathematical modeling) or computer pseudo–code (computational modeling). Nowadays, the two are almost synonymous, and we use the terms interchangeably.

As the world is vast and complex, attempting to model all of it is doomed to fail, and models are usually created to explain some phenomena in a specific domain. Lewis Fry Richardson created one of the first mathematical models for weather forecasting in the $1920s$ [146]. The model's results were severely wrong due to the crude approximations involved in the model. Nowadays, computational models are executed with ever more computing power, making them more complex and accurate.

Most computational models can be categorized as either *parametric* or *nonparametric*. Parametric models rely on a set of parameters that characterize the equations behind the model. Often, these parameters can be traced back to the physical features of the problem being modeled. Furthermore, parametric models are often used when we want to introduce to the model some prior knowledge we have. In this sense, the model's equations represent certain hypotheses about the phenomenon we wish to model. These hypotheses could be very exact, making the model usable only for a specific application.

Nonparametric models do not rely on concrete assumptions about the data we model but on the statistical features of the data itself. A popular example of nonparametric modeling is kernel density estimation [150, 76, 45], which is used to approximate the underlying density function from data. Artificial neural networks are another popular nonparametric model [151]. Although these models do have a finite set of parameters, they have so many of them that they are considered to emulate non–parametric methods.

In this work, we look at *generative* models. This class of models differs from other models used to classify or cluster data. The generative models produce *time series* data in our case. More specifically, in this work, we model time series where the events are characterized as points in time and space. A valuable framework for this kind of modeling is *point processes*, and we present it in the next chapter.

The generative models represent our understanding of the process underlying the observations, and we expect that the synthetic data that the model generates is similar to the observed data. Setting the model parameters "blindly" or "by hand" will likely result in synthetic data that is not similar to the observed data we wish to model. Thus, after setting the model, a crucial step is adapting the model's parameters according to the observed data. This is often referred to as *learning* or *inference*.

1.1.1 Probabilistic Modeling

In many cases, we observe different outcomes given the same initial conditions. For example, following an earthquake of a certain amplitude in a specific location, we cannot predict precisely when and where the aftershocks will appear. This is due to inherent noise in the system and our imperfect knowledge about the problem, which limits our modeling capabilities.

When choosing to address the uncertainty in the data, we use the framework of *probabilistic modeling*. Rather than modeling a specific outcome, we model a distribution. In this framework, the model is defined by the *likelihood* function, which is the probability of observed data given the model's parameter values. Thus, when generating data from the model, we effectively sample from the likelihood, and given the same initial conditions, different synthetic data–sets are produced. We do not expect the data generated from a probabilistic model to be identical to the observed data, and we use statistical methods to establish the model's accuracy.

One approach to statistical modeling relies on the view of the probability of an event as the relative occurrence frequency of the event in repeated experiments. It is known as *frequentist*. In this framework, probabilities are associated only with observed quantities, e.g., the data.

Another approach assumes that uncertainty is also associated with unobserved quantities, e.g., the model's parameters. According to this approach, the parameters of the models also come from a particular distribution, and as we cannot infer their exact values, we estimate their distribution. This approach is known as *Bayesian*. In this work, we use the Bayesian framework and introduce it in the next section.

1.2 The Bayesian framework

The origins of Bayesian inference are tracked back to the 18th century and the influential work by Thomas Bayes, "An Essay towards solving a Problem in the Doctrine of Chances" [6]. This paper was published two years after the death of its author, edited by Richard Price. In his essay, Bayes presented a theorem that calculated the probability of an occurrence of an event given the prior knowledge that another event took place. However, Bayes' theorem was not widely accepted then, and only in the 20th century, it began to be commonly used in statistical analysis.

Bayesian inference has been utilized in many fields. It can be applied to a broad range of problems, from "simple" problems such as estimating the probability of a coin being fair based on the observed frequency of heads and tails, to estimating the likelihood of a patient having a disease based on their symptoms and tests results and predicting the stock price of a company based on its financial performance and market conditions.

As stated before, unlike the frequentist approach, the Bayesian approach assumes that the model's parameters also have uncertainty associated with them. This allows one to incorporate prior knowledge and beliefs into the analysis, which can be helpful in situations with limited or noisy data. This comes into play as *prior distributions* over the model's parameters.

One of the main challenges when using Bayesian inference is the need to specify the prior distributions. Bayesian inference is sometimes referred to as *subjective*. The priors are considered non–objective knowledge, as they are not directly based on the observations. This can be addressed using *non–informative* prior distributions. But in this work, we remain in the field of subjective Bayes and use informative priors.

Generally, Bayesian inference framework follows three steps:

1. Setting up the full probability model over all the observed (data) and the unobserved (model's parameters) quantities in the form of likelihood and prior distributions.

2. Estimating the probability of the model's parameters conditioned on the observed data, known as the *posterior* distribution.

3. Evaluating the model.

The contribution of this dissertation involves all three steps of the Bayesian framework. We introduce novel Bayesian models in different application fields, exploring both parametric and nonparametric approaches. Moreover, we develop efficient inference algorithms for learning these models' parameters and evaluate the inference results.

1.3 Outline

In the next chapter, we aim to cover the fundamental mathematical knowledge needed for the rest of the dissertation. We elaborate on Bayesian inference, introduced in the previous section, and present some of the most common inference algorithms. For the use case of models for which the inference is not directly available, we introduce the concept of model augmentation and go through the details of one augmentation scheme we use later in our work. Last, we go through the general details of point processes, which are the type of statistical models that we develop in this dissertation.

Chapter 3 includes our contribution to modeling and inference in point processes. We present a novel semi–parametric model named the Nonlinear Hawkes process with Gaussian Process Self–effects (NHGPS), a nonlinear extension to the known Hawkes process. We describe the innovative inference approach for the model's parameters, which utilizes the augmentation scheme described in Chapter 2. The augmentation allows us to derive a Gibbs sampler and a Variational Inference algorithm for our model. We first demonstrate the performance of the inference on artificially generated data and then use the model in applications in different domains - from neuroscience to crime prediction.

In Chapter 4, we dive into a specific application domain for point processes, namely human natural scene viewing. We develop a new hypotheses–based parametric model to generate human–like scan paths given an image. The idea underlying our approach is that each saccade can be categorized as "exploratory" or "exploitative," and we formalize this assumption in terms of fixation location probability. We name our model Exploration–Exploitation model for scene viewing. Once we set the new model, we describe a Bayesian inference approach relying on the PG augmentation scheme, which is novel in the field. Similarly to the steps in Chapter 3, we first validate the correctness of the inference scheme using artificial data generated from the model. Then we investigate the performance of the model on real data.

Chapter 5 addresses the choice of an inference algorithm for a Bayesian model. We remain in the field of scan path modeling and compare the performances of different sampling algorithms for models with explicit likelihood functions. We use the Exploration–Exploitation model presented in the previous chapter and the known Scenewalk model as test cases.

Finally, we conclude the dissertation in Chapter 6. We summarize the main contribution of our work and suggest further developments and research directions.

Fundamentals

2

2.1 Bayesian Inference

The first building block of the Bayesian framework is the model. In this work, a model is defined by a likelihood function which is the probability density function associated with the chance of generating specific data given the model parameters. We use the notation y for data and θ for model parameters. The likelihood function of the model is then

$$p\left(y|\theta\right). \tag{2.1}$$

The likelihood also depends on the choice of the model $\mathcal{M}$. This is often denoted as $p\left(y|\theta,\mathcal{M}\right)$ or $p\left(y|\theta\right)$, but here we keep this dependency implicit and use the notation in Equation (2.1). The model is further specified with the full probability density of the model, which also includes the prior over the model parameters

$$p\left(y,\theta\right) = p\left(y|\theta\right)p\left(\theta\right). \tag{2.2}$$

The next step in the Bayesian framework is estimating the posterior distribution over the model parameters given observed data. This is done using Bayes rule [6]

$$p\left(\theta|y\right) = \frac{p\left(y|\theta\right)p\left(\theta\right)}{p\left(y\right)}. \tag{2.3}$$

The expression in the denominator on the right–hand side of the equation is known as the *evidence*. In practice calculating the evidence analytically is usually impossible and estimating it is very computationally costly. In practice, when we "estimate" the posterior, we estimate a quantity that is only proportional to it.

The posterior probability of the model parameters can be estimated directly as an analytical expression only for a very limited family of distributions. This is usually the case when the prior and likelihood of the model are conjugated and the posterior distribution has the same functional form as the prior distribution. Some common examples of conjugate distributions include: 1. Beta-binomial: If the prior distribution is a beta distribution and the likelihood is a binomial distribution, then the posterior distribution is also a beta distribution. 2. Normal-normal: If the prior distribution is a normal distribution and the likelihood is also a normal distribution, then the posterior distribution is also a normal distribution. 3. Gamma-Poisson:

If the prior distribution is a gamma distribution and the likelihood is a Poisson distribution, then the posterior distribution is also a gamma distribution.

In cases where the posterior distribution cannot be derived analytically, a common approach is to use sampling methods. By sampling, we refer to the process of drawing a set of random points that approximate the posterior distribution. With these points, we can calculate the summary statistics of the posterior, such as the mean or median, or visualize the distribution using techniques like histograms or kernel density plots.

A key element of this process is the proposal distribution $q(\theta)$, which is simple enough to sample from it. Different samplers differ from each other by the construction of $q(\theta)$ and the construction of the sequence of samples known as the *chain*.

Another approach is deterministic approximate inference methods. This approach also includes a proposal distribution $q(\theta)$, but instead of sampling from it, it is tuned to be as similar as possible to the posterior. Common deterministic approximate inference methods are *expectation propagation* [123] and *variational inference* [15]. The latter is introduced and used in Chapter 3 of this dissertation.

Next, we introduce the sampling algorithms used in several of the following chapters. We first present a brief history of the development of Monte Carlo methods and then dive into the mathematical framework of these methods.

2.1.1 Markov Chain Monte Carlo

Background

The origins of Monte Carlo (MC) methods are often tracked back to Metropolis and Ulam [121], and the method is named after the famous Monte Carlo casino. Following the rapid development of the electronic computer in the $1940s$, statistical sampling was revisited after being considered too tedious to calculate. MC was originally developed for estimating the state of a physical particle system, where the state cannot be evaluated analytically, but the number of particles is not high enough to apply methods from statistical mechanics. Metropolis and Ulam suggested randomly choosing a sub–set of particles and propagating them, creating a chain until the system's final state is reached. Interestingly, it was reported that the nuclear physicist Enrico Fermi developed MC independently already in the $1930s$, but did not publish on the subject [156].

In the upcoming years, MC methods remained a tool for physicists to simulate particle systems. In 1953 Metropolis, Rosenbluth, Rosenbluth, Teller, and Teller [120] presented the first Markov Chain Monte Carlo (MCMC) algorithm, which became known as the *Metropolis* algorithm. While using terminology from statistical physics, de facto Metropolis et al. presented an algorithm to simulate any given distribution of random vectors. The Metropolis algorithm was generalized further in Hastings [65] and was named the Metropolis–Hastings (MH) algorithm. In this work, Hastings extended the usage of MCMC methods beyond statistical mechanics to general high–dimensional statistical problems.

Following the pioneering work of the $1950s$ and $1970s$, the following decades brought much attention to using MCMC methods for Bayesian inference. New MCMC algorithms were developed [56, 62] and applied to real–world problems in different fields [9, 12, 58]. Furthermore, MCMC methods were used for different Bayesian computations, such as calculating marginal densities [53] and exploring and summarizing posterior distributions [159, 173]. Last, a rigorous mathematical framework was constructed to assess the validity of those methods [10, 11].

MCMC

Here, we are interested in using MCMC methods to estimate a complex posterior distribution $p\left(\cdot\right)$. In MCMC, the distribution we wish to estimate is called the *target* distribution. As we cannot sample directly from the posterior, we sample from a *proposal distribution* $q\left(\cdot\right)$. If the MCMC algorithm is well constructed, the samples from the proposal distribution converge to the target distribution.

The term Markov Chain refers to methods where the proposal distribution is conditioned on the previous samples. In most MCMC type samplers, the chain is first–order, which means it only depends on the last sample

$$q\left(\theta_{n+1}|\theta_1,..,\theta_n\right) = q\left(\theta_{n+1}|\theta_n\right).$$

The conditional probability in a Markov chain is also known as the *transition probability* $T_n\left(\theta_n,\theta_{n+1}\right) = q\left(\theta_{n+1}|\theta_n\right)$.

To ensure that the samples generated from the proposal distribution converge to samples from the target distribution, the Markov chain must be constructed such that the target distribution is invariant. One way to ensure that is to choose the transition probabilities such that they satisfy *detailed balance*

$$p\left(\theta_n\right)T\left(\theta_{n+1},\theta_n\right) = p\left(\theta_{n+1}\right)T\left(\theta_n,\theta_{n+1}\right).$$

In the context of Markov Chains, this is also called *reversability* of the chain. Furthermore, we wish to achieve the invariant property regardless of the initial distribution $q\left(\theta_0\right)$. This is called *ergodicity*, and if satisfied, the invariant distribution is the *equilibrium* distribution.

The different MCMC algorithms vary in the construction of the transition probabilities, such that the target distribution is the equilibrium distribution of the Markov chain. Next, we present the popular Metropolis–Hastings (MH) and two of its derivatives — Gibbs, and Hamiltonian Monte Carlo (HMC).

MH

In the MH algorithm [65], after drawing a sample θ' from the distribution $q\left(\theta|\theta_n\right)$ it is accepted with probability

$$\alpha\left(\theta', \theta_n\right) = min\left[1, \frac{p\left(\theta'|y\right) q\left(\theta_n|\theta'\right)}{p\left(\theta_n|y\right) q\left(\theta'|\theta_n\right)}\right]. \tag{2.4}$$

The Metropolis acceptance probability defines a Markov chain whose invariant distribution is $p\left(\theta|y\right)$ as it satisfied the detailed balance condition

$$\begin{aligned} p\left(\theta_n|y\right) q\left(\theta'|\theta_n\right) \alpha\left(\theta', \theta_n\right) =& \min\left[p\left(\theta_n|y\right) q\left(\theta'|\theta_n\right), p\left(\theta'|y\right) q\left(\theta_n|\theta'\right)\right] \\ =& \min\left[p\left(\theta'|y\right) q\left(\theta_n|\theta'\right), p\left(\theta_n|y\right) q\left(\theta'|\theta_n\right)\right] = \\ =& p\left(\theta'|y\right) q\left(\theta_n|\theta'\right) \alpha\left(\theta_n, \theta'\right). \end{aligned}$$

Hence, every MCMC method that can be written as a special case of the MH algorithm also has the target distribution as its invariant distribution. Nowadays, the most popular version of MH is random–walk MH. In this algorithm, the proposal distribution is a Gaussian with mean θ_n. As this is a symmetrical distribution, the acceptance probability becomes

$$\alpha\left(\theta', \theta_n\right) = min\left[1, \frac{p\left(\theta'|y\right)}{p\left(\theta_n|y\right)}\right].$$

Gibbs Sampler

Gibbs Sampler is a special case of the MH algorithm that is applicable for posterior distribution where we can sample from the conditional posterior of the parameters. We set $\theta = \{\theta_1, ..., \theta_K\}$, and assume that we can sample from each conditional $p\left(\theta_k|y, \theta_{\setminus k}\right)$. In each step of the Gibbs sampler, we iterate through all the parameters and sample a new value from the conditional posterior for each. The procedure is summarized in Algorithm 1.

Algorithm 1: Gibbs Sampler

```
Initialize $\{z_k; k = 1, ..., K\}$
for $n \leftarrow 1$ to $N$ do
    Sample $z_1^{(n)} \sim p\left(z_1|y, z_2^{(n-1)}, ..., z_K^{(n-1)}\right)$
    $\vdots$
    Sample $z_k^{(n)} \sim p\left(z_1|y, z_1^{(n)}, ..., z_{k-1}^{(n)}, z_{k+1}^{(n-1)}, ..., z_K^{n-1}\right)$
    $\vdots$
    Sample $z_K^{(n)} \sim p\left(z_K|y, z_1^{(n)}, ..., z_{k-1}^{(n)}\right)$
```

Hamiltonian Monte Carlo

So far, we have introduced two sampling schemes - MH and Gibbs sampler. Indeed, the two methods are guaranteed to converge to the correct target distribution when the number of iterations goes to infinity. In practice, we only have access to a finite number of iterations. Often, the convergence of the MH algorithm is very slow, as it tends to zigzag around the target distribution and explores it inefficiently due to the random walk on which it is based. Although the Gibbs sampler converges faster, we do not have access to the analytical form of the required conditional distributions in many cases.

Next, we present the *Hamiltonian Monte Carlo* (HMC) algorithm, which addresses these issues and can be applied to any model with a computable likelihood function. HMC was first used in 1987 [42] under the name *Hybrid Monte Carlo*. It was termed *hybrid* as it combines stochastic MCMC updates and deterministic simulation. The now accepted name *Hamiltonian Monte Carlo* was first introduced in 2003 [113], and it refers directly to the mechanical physics inspiration behind it.

A naive idea for improving the sampling path is to track the gradient of the target distribution. Implementing this idea, our samples will soon be drawn to the distribution mode, and we will fail to explore the tails of the distribution, which are necessary, especially in the case of high dimensional problems.

To counteract the attraction to the mode of the distribution in HMC, we augment the target distribution with a parameters vector ϕ called the *momentum* and the joint distribution of the problem parameters, and the momentum is

$$p\left(\theta,\phi|y\right)=p\left(\phi|y,\theta\right)p\left(\theta|y\right)=p\left(\phi|\theta\right)p\left(\theta|y\right),$$

known also as the *canonical* distribution. In doing so, we extend the parameter space of the problem to the associated *phase space* and sample in this expanded space.

As inspired by classical physics, we can investigate the augmented system in terms of the Hamiltonian associated with it

$$H\left(\theta,\phi\right)=-\log p\left(\theta,\phi\right)=-\log p\left(\phi|\theta\right)-\log p\left(\theta|y\right)\equiv K\left(\phi,\theta\right)+V\left(\theta\right).$$

V is the associated *potential energy* of the system, such that the posterior distribution can be written as $p\left(\theta|y\right)=\exp^{-V}$, and K is the associated *kinetic energy* such that $p\left(\phi|\theta\right)=\exp^{-K}$. Generally, the kinetic energy could also depend on the position variable θ. Here, we consider the separable case where the kinetic energy depends only on the momentum variable ϕ and $p\left(\phi|\theta\right)=p\left(\phi\right)$.

The definition of the Hamiltonian allows the use of the *Hamilton equations* to propagate θ and ϕ in time while preserving the system's total energy. This translates to the following steps that make up an iteration in the HMC sampler

(i) Sample momentum $\phi\sim p\left(\phi\right)$.

(ii) Propagate the system in the phase space using Hamilton equations

$$\begin{aligned}\frac{d\theta}{dt} &= \frac{\partial H}{\partial \phi} = \frac{\partial K}{\partial \phi}\\ \frac{d\phi}{dt} &= -\frac{\partial H}{\partial \theta} = -\frac{\partial K}{\partial q}.\end{aligned}$$

(iii) Use the new state and the negative new momentum as a proposed sample (θ^*, ϕ^*) and accept it with probability

$$\min\left[1, \frac{p(\phi^*)\,p(\theta^*|y)}{p(\phi)\,p(\theta|y)}\right].$$

The negation of the momentum in the last step is done to ensure the symmetry of the Metropolis proposal. Next, we set the kinetic energy so that $K(\phi) = K(-\phi)$ so the negation is not done in practice. If we step into an "illegal" value of the parameters while propagating the system, we stop the progress in the phase space and reject the new state we arrived in.

An important design choice when implementing HMC is the kinetic energy associated with the momentum parameter. The most commonly used is the *Euclidean–Gaussian Kinetic Energy* where

$$p(\phi) = \mathcal{N}(\phi|0, M) \tag{2.5}$$

$$K(\theta,\phi) = \frac{1}{2}\phi^\top M^{-1}\phi + \frac{1}{2}\log|M| + \text{const.} \tag{2.6}$$

M is a symmetric, positive–definite *mass matrix*. Often M is diagonal and set to be a multiplication of the identity matrix.

Another way to set the mass matrix is

$$M^{-1} = \frac{1}{n}\sum_{n=1}^{N}\Sigma(\theta_n)$$

where $theta_n$ is the value of θ at iteration n of the sampler. M is the inverse of the average covariance of the first N iterations of the position variable θ. In this approach, the sampler is run for N burn–in iteration with $M = I$ and then set to the expression above for the rest of the run. This approach compensates for correlations between position parameters and is often used in molecular dynamics [7].

Other methods automatically adapt the mass matrix according to the local structure at each step of the algorithm [2, 60, 97]. These methods require adaptation of the acceptance probability, and we do not use them in this work.

Even for straightforward problems, we cannot analyze the trajectories defined by the Hamilton equations but resort to numerical approximations. One of the most used methods for discretization and solving a system of differential equations is the Euler method [87]. This method, and other similar solvers, introduce a "drift" — as longer and longer trajectories are solved and the integration time increases, the error introduced by the discretization accumulates [131].

To avoid the accumulation of errors, the Hamilton trajectories are solved using *symplectic integrators* [101]. When choosing Euclidean–Gaussian kinetic energy for the momentum parameter, we can use the *leapfrog integrator* to solve the Hamilton equations, which received its name from splitting the momentum step. One step of the leapfrog integrator is described in Algorithm 2.

Algorithm 2: Leapfrog Algorithm

```
Data: Initial position $\theta_0$ and momenta $\phi_0$, step size $\epsilon$
Result: Position and momenta at time $\epsilon$
begin
    Half-step for momenta:
    $\phi_{\frac{1}{2}} \leftarrow \phi_0 - \frac{\epsilon}{2}\frac{\partial V}{\partial \theta}|_{\theta_0}$
    Full-step for positions:
    $\theta_1 \leftarrow \theta_0 + \epsilon \frac{\partial K}{\partial \phi}$
    Half-step for momenta:
    $\phi_1 \leftarrow \phi_{\frac{1}{2}} - \frac{\epsilon}{2}\frac{\partial V}{\partial \theta}|_{\theta_1}$
```

The performance of the HMC algorithm depends heavily on the choice of the parameters of the leapfrog algorithm – the step size ϵ and the number of iterations $L = \lfloor T/\epsilon \rfloor$. Finding the appropriate parameters can be done during the burn–in period. A useful heuristic is setting $\epsilon L = 1$ with $\epsilon = 0.1$ and $L = 10$ and changing the step size according to the average acceptance probability. If the acceptance probability is too high, the step size is increased, and if it is too low, the step size is reduced [54].

After the burn–in period, during the run of the HMC, the values of ϵ and L can be changed randomly, and the algorithm will still converge to the correct distribution [54]. Some more complex algorithms have been developed that change the step size and the number of steps during the algorithm's run while keeping the detailed balance and not damaging the algorithm's convergence. Such a method is the *no–U–turn sampler* [75], which will be presented later in the dissertation.

2.1.2 Augmentation

The HMC sampler described earlier incorporates the useful mechanism of *data augmentation*. We discuss it in further detail and introduce a specific augmentation scheme that appears in several of the following chapters.

Consider again the situation where the posterior distribution we are interested in $p(\theta|y)$ is difficult to compute. It is also often the case that the conditionals required for the Gibbs sampler are not available in an analytic form. In data augmentation, another set of parameters ϕ is imputed such that $p\left(\theta|y,\phi\right)$ has a convenient form. These new parameters can also be seen as *unobserved data*, hence the terminology of *data* augmentation.

Suppose $p\left(\theta|y,\phi\right)$ is available, and we can sample the augmenting parameters conditioned on the data and the model. In that case, we can implement a two steps block Gibbs sampler, where we iterate between sampling the model parameters conditioned on the unobserved data and the observed data from $p\left(\theta|y,\phi\right)$, and sampling the unobserved data conditioned on the model parameters and the observed data from $p\left(\phi|y,\theta\right)$. As we can always marginalize over the augmenting parameters, the samples θ will be sampled from $p\left(\theta|y\right)$.

Gibbs sampling with data augmentation has been used in many applications from genetics linkage data [37, 166] to mixture distributions [38, 38] and hierarchical models [139]. Although we introduce data augmentation in the context of Gibbs sampling, it can also be used with other inference schemes, such as variational inference, as demonstrated in later chapters.

Pòlya-Gamma Augmentation

The Pòlya-Gamma (PG) augmentation scheme allows for a full Bayesian inference in models with binomial likelihoods. It was first presented in [142], and its theoretical foundations were further explored in [29].

A random variable w is said to be distributed according to PG distribution if

$$w \overset{D}{=} \frac{1}{2\pi^2}\sum_{k=1}^{\infty}\frac{g_k}{\left(k-\frac{1}{2}\right)^2+\frac{c^2}{4\pi^2}}$$

with parameters $b>0$ and $c\in\mathbb{R}$ and $g_k\sim\text{Gamma}\left(b,1\right)$. The moment generating function of the PG distribution is given by

$$\mathbb{E}\left(\exp\left(w\theta\right)\right)=\frac{\cosh^b\left(\frac{c}{2}\right)}{\cosh^b\left(\sqrt{\frac{c^2/2-\theta}{2}}\right)}.$$

Taking the derivative with respect to θ and evaluating at 0 results in an analytical expression for the mean of the PG variable

$$\mathbb{E}\left(w\right)=\frac{b}{2c}\tanh\left(\frac{c}{2}\right). \tag{2.7}$$

The PG distribution was designed to represent the inverse of the hyperbolic cosine function as an infinite mixture of scaled Gaussians

$$\cosh^{-b}(\theta/2) = \int_0^\infty \exp\left(-\frac{\theta^2}{2}w\right) p(w|b,0)\,dw. \tag{2.8}$$

This is particularly useful in problems where the likelihood includes the logistic function, as it can be rewritten in terms of the hyperbolic cosine

$$\sigma(\theta) = \frac{1}{1+e^{-\theta}} = \frac{\exp\frac{\theta}{2}}{2\cosh\left(\frac{\theta}{2}\right)}. \tag{2.9}$$

Using the representation from Equation (2.8) in Equation (2.9)

$$\sigma(\theta) = \frac{1}{2}\int_0^\infty \exp\left(-\frac{\theta^2}{2}w + \frac{\theta}{2}\right) p(w|1,0)\,dw \tag{2.10}$$

We get an expression where θ appears quadratic in the exponent and is conjugated to a Gaussian prior. Furthermore, using the tilted PG distribution we see that

$$p(w|1,\theta) \propto \exp\left(-\frac{\theta^2}{2}w\right) p(w|1,0).$$

The PG augmentation scheme achieves conjugacy in models with binomial likelihoods. It was used successfully in logistic regression models [142, 105] and other models that include the logistic function [106, 41]. In the next chapters, we continue this line of research and employ the PG augmentation in point process models that include the sigmoid function.

2.2 Point process

As mentioned, our research is concerned with modeling phenomena characterized as sequences of events. The framework of point processes is often used to describe this type of data, and we next present it in general detail.

Point processes are utilized to describe the random generation of data that are concentrated in some space. These could be events in time or locations in space, for example. Generally, point processes are defined in an arbitrarily high dimensional space $\mathcal{S} \subseteq \mathbb{R}^d$ [127]. A point process X on $\mathcal{S}$ can be characterized in terms of the associated counting process with the count function

$$N(B) = n(X_B),$$

the number of points falling in $B \subseteq \mathcal{S}$.

In this work, we focus on one–dimensional point processes and model events in finite time. This allows us to use other formulations of the point process than the count function. Two of these formulations are in terms of event times and of inter–arrival times

$$N = \{\tau_1, ..., \tau_n\}$$
$$N = \{u_1, ..., u_n\} \quad u_i = \tau_i - \tau_{i-1}.$$

An illustration of the different formulations can be found in Figure 2.1. In the rest of the dissertation, we use the formulation of a temporal point process as a collection of event times.

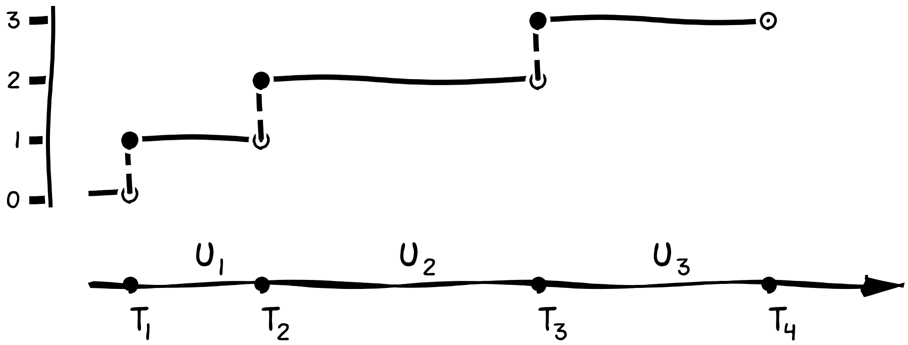


Fig. 2.1.: Above is the representation of a point process as a counting process. Below is the representation using arrival and inter–arrival times.

Two common and useful assumptions for a temporal point process are that it is *evolutionary* and *orderly*. *Evolutionary* means that the process evolves in time and that past events may influence the current event, but future events do not. *Orderly* implies that all events occur at distinct times and formally

$$\lim_{\Delta t \to 0} P\left\{N\left(t, t + \Delta t\right) > 1\right\} = 0. \tag{2.11}$$

The evolutionary assumption allows us to examine a temporal point process in terms of the conditional density

$$f\left(t|\mathcal{H}_{t_n}\right) dt = P\left\{t_{n+1} \in [t, t + dt) | \left\{t_1, ..., t_n\right\}\right\}.$$

The term $\mathcal{H}_{t_n}$ is called the *history* of the process. With $F\left(t|\mathcal{H}_{t_n}\right)$ being the corresponding cumulative distribution function, we can consider the conditional *intensity* function, also known as the *hazard* [122] function or *rate* function of the point process

$$\lambda\left(t|\mathcal{H}_{t_n}\right) = \frac{f\left(t|\mathcal{H}_{t_n}\right)}{1 - F\left(t|\mathcal{H}_{t_n}\right)}. \tag{2.12}$$

We can write the conditional density function in terms of the intensity $\lambda\left(t|\mathcal{H}_{t_n}\right)$ [35]

$$f\left(t|\mathcal{H}_{t_n}\right) = \lambda\left(t|\mathcal{H}_{t_n}\right) \exp\left(-\int_{t_n}^{t} \lambda\left(s|\mathcal{H}_{t_n}\right) ds\right) \tag{2.13}$$

and the joint likelihood function of the events in the process is

$$L\left(t_1, ...t_n\right) = \prod_{i=1}^{n} \lambda\left(t_i\right) \exp -\Lambda\left(t\right),$$

where $\Lambda\left(t\right) = \int_0^t \lambda\left(s|\mathcal{H}_{s-}\right) ds$ is called the *compensator* and $\mathcal{H}_{s-} = \{t_1, ..., t_n; t_n < s\}$.

2.2.1 Poisson Process

For completion, we present the most known type of point process called *Poisson process*. Above, we defined a temporal process where the probability of the occurrence of an event in a specific time depends on the history of the process. This dependence may take many forms. The most basic of these forms is the assumption that all event times are independent, and the process is *memoryless*. Formally it is expressed as

$$\begin{aligned} f\left(t|\mathcal{H}_{t_n}\right) &= f\left(t\right) \\ \lambda\left(t|\mathcal{H}_{t_n}\right) &= \lambda\left(t\right). \end{aligned}$$

Processes following this assumption are called Poisson Processes [84]. If the Poisson Process is constant with $\lambda\left(t\right) = \lambda$, the process is called *homogeneous*, and otherwise, it is called *nonhomogeneous*.

An important extension of the Poisson process is the *Cox Process* [32] or *doubly stochastic Poisson process*, in which the intensity function itself is a stochastic process. One example is a process where the logarithm of the intensity function is generated from a *Gaussian Process* [128].

3 Nonlinear Hawkes Process with Gaussian Process Self-Effects

We concluded the fundamental theoretical background by introducing point–processes, and the first original contribution of this dissertation is a new point–process model. This model belongs to the family of Hawkes processes, which differs from the Poisson process by being history dependent. Our model is part of the nonlinear Hawkes process family, where the causal influence between events can be either excitatory or inhibitory. We use the term *self–effects* to refer to the influence of the past on future events.

In a nutshell, the innovation of our model is that we choose a semi-parametric approach, which avoids the limiting parameterization of the memory kernel and the background rate. We assume a Gaussian process (GP) before the exogenous events' intensity and on the memory kernel, allowing an inhibitory effect between the events. To ensure that the intensity function is non–negative, we use the sigmoid link function. This descriptive modeling approach allows us to obtain a fast inference procedure. The history of self–effects defines an aggregated Gaussian process, and we perform the inference directly on this aggregation rather than obtaining a posterior over each self-effect.

We begin this chapter by introducing Hawkes processes and their main application fields. Next, we introduce some of the latest work in the field that we later use as a baseline comparison for our model. Then, we present our new model and the inference method we developed for it. After presenting the model, we include results from different fields and briefly discuss the main findings.

3.1 Hawkes Processes

Traditionally, sequences of events in continuous time are modeled by point processes, of which Cox processes [32], or doubly stochastic processes, use a stochastic process for the intensity function, which depends only on time and is not affected by the occurrences of the events. In the 1970s, Alan Hawkes developed the Hawkes process in a series of five papers [70, 68, 69, 66, 71]. The Hawkes process extends the Cox process to capture phenomena in which past events affect future arrivals by introducing a memory dependence via a memory kernel, which is also referred to as the causal influence function. When incorporating the dependence of the process on its own history, due to the superposition theorem of the point process, new events will depend on either an exogenous rate, which is independent of the history, or an endogenous rate from past arrivals.

3.1.1 Classical and Nonlinear Hawkes Process

In Section 2.2, we presented the general evolutionary point process, its rate function (Equation (2.12), and its conditional density function (Equation (2.13)). Unlike in the case of the Poisson process, the Hawkes process considers the effect of the history on the process on the intensity function. Following the notation in Hawkes [70], the intensity of the Hawkes process is defined by

$$\lambda(t|\mathcal{H}_t) = s\left(t\right) + \sum_{t_n<t} g\left(t - t_n\right), \tag{3.1}$$

where $s(t)$ is the base intensity of exogenous arrivals and $g\left(t - t_n\right)$ is the memory kernel, or causal influence function, defining the change in the intensity function following each arrival.

Originally, Hawkes processes were developed for phenomena where the history has an *excitatory* effect on the intensity function. Thus, past events may only increase the intensity function. To avoid an "explosion" of events, exponential decay is introduced, limiting each event's effect duration. Under this assumption, the most common form of the memory kernel is

$$g(t - t_n) = \beta e^{-\alpha(t-t_n)} \tag{3.2}$$

In the first 30 years after the Hawkes process was presented, its main application was earthquake modeling. Following an earthquake, the probability of aftershocks increases for a certain amount. Hawkes processes were first used to model earthquakes in Ogata [135]. They were quickly adopted in the field [136, 83, 179], and extended to model both the time and location of events [130, 137, 126].

Although the seismology community adopted it, Hawkes processes gained only moderate attention in those years. A big shift happened in the first decade of the 21st century when Hawkes processes were enthusiastically adopted in finance [19, 98, 162, 36]. Since the Hawkes process is time continuous, it proved to be especially useful for high–frequency price variations where binning time greatly hinders the prediction abilities of a model. Furthermore, these data are characterized by clusters of events, which make the Hawkes process especially suitable to model them [4, 67].

After gaining popularity in finance, the usage of Hawkes processes spread into other fields in social sciences, where data are modeled as point events in time that influence each other. Some examples are social media analysis [111, 88, 161] and crime prediction [124, 194, 138] where an event may trigger other events and cause a cascade of events.

The applications mentioned above assume only an excitatory relation between the events. This assumption does not hold for other potential applications of the Hawkes process. One application where point process models are often used, and events may have both inhibitory and excitatory influence on each other, is neuronal activity. For example, inhibitory effects between neurons [114], and even self-inhibition [160], are crucial for regulating neuronal activity.

When modeling neuronal activity with the Hawkes process, the memory kernel should also include inhibitory relations between the events, and by doing so, the intensity may become negative. To ensure that the intensity function is non–negative, a nonlinear link function is applied on the memory kernel, and the resulting model is often referred to as a *nonlinear* Hawkes process [22, 193, 175]. Theoretical results for nonlinear Hawkes processes have been developed for many years, and they include stability estimates [22] as well as convergence rates for Bayesian estimators [164].

In this chapter, we develop a new extension of the nonlinear Hawkes process and a Bayesian inference framework to estimate its parameters. To put it in the context of current developments in the field, we briefly introduce Bayesian inference for methods for the related Cox process and present several recent nonlinear Hawkes models to which we later compare our model.

3.1.2 Current Nonlinear Hawkes Process Models

Bayesian approaches to Cox processes model the intensity with a Gaussian process prior, which is then passed through a link function to ensure its positivity. A common choice of the link function is the exponential or the quadratic functions [74, 110]. Another choice, which is more relevant to our work, is the sigmoid link function, resulting in the *sigmoidal Gaussian Cox process*. Inference in this model was first made empirically in Möller, Syversveen, and Waagepetersen [128], as well as with moment-based parameters estimators [23]. Markov chain Monte Carlo methods were also developed [1], as well as variational inference [41].

In this chapter, we use a Bayesian semi-parametric inference approach. Earlier work, which introduces Bayesian nonparametric approaches to point processes, includes, among others, Ishwaran and James [77], who define kernel mixtures of Gamma measures for the intensity, Wolpert and Ickstadt [183], who define inhomogeneous Gamma random fields, and Taddy and Kottas [165], where joint nonparametric mixtures are introduced.

As for the Hawkes process, the first attempts to perform Bayesian inference relied on the definition in terms of a marked Poisson cluster process and identifying the branching structure of the self–excitation [144]. One current model that uses this approach is the mutually regressive point process (MRPP) [3]. MRPP is designed to model neuronal spike trains, and a probability term augments the classical self-excitatory Hawkes Process intensity function. This term induces inhibition when it is close to zero. In a sense, this model includes two memory kernels—one excitatory only, which appears in the intensity function, and another, which can also induce inhibition in the augmenting probability term. Unlike the MRPP, we achieve such flexibility of the self-effects in a simpler fashion by assuming the GP prior on the self–effects. As mentioned before, this also allows for the type of effect to change over time, which does not appear in the work of Apostolopoulou et al. [3].

A highly flexible approach to estimating the intensity function of the Hawkes process relies on GP priors [188, 190, 189]. A recent adaptation of this approach is the model described in Zhou et al. [191]. The authors avoid the limiting parameterization of the memory kernel by using GPs. Unlike our model, Zhou et al. [191] remain in the linear Hawkes process regime and assume that the effects of past events are only excitatory, whereas our approach allows both excitatory and inhibitory effects.

The last variation we describe is a sigmoid nonlinear multivariate Hawkes process (SNMHP) [192]. In this work, Zhou et al. describe a multivariate nonlinear HP, where, similarly to our work, the chosen link function is a sigmoid; however, they chose to model the causal influence function with a weighted mixture of basis functions from a certain family. Similar to the MRPP model, SNMHP was designed to model neuronal activity. A common assumption in this field is that each neuron is affected only by a subset of the other neurons in the network. Zhou et al. incorporate this assumption directly into their model by including a sparsity-inducing prior over the weights. In terms of inference, Zhou et al. proposed an expectation–maximization algorithm, whereas we propose a fully Bayesian approach.

3.2 Nonlinear Hawkes Process with Gaussian Process Self–Effects (NH–GPS)

In the classical Hawkes process, the memory kernel g in Equation (3.1) must be non–negative to prevent the intensity function from being negative. As a result, the model's history has only an excitatory effect on future events. We are interested in a model that includes inhibition between events, and we release the constraint over g, so it can be negative and define the following nonlinear intensity function

$$\lambda(t) = \lambda^{*} \sigma(\phi(t)) \tag{3.3}$$

$$\sigma(\phi(t)) = \frac{1}{1+\exp(-\phi(t))} \tag{3.4}$$

$$\phi(t) = s(t) + \sum_{t_n<t} g(t-t_n) \exp(-\alpha(t-t_n)) . \tag{3.5}$$

Here, we choose the sigmoid function to ensure that the intensity function $\lambda(\cdot)$ is non–negative. λ^{*} is the intensity bound and $\phi(\cdot)$ is the unmodulated intensity function, or the linear intensity function. $s(\cdot)$ and $g(\cdot)$ that appear in the equation above are the modulated versions of the background rate and self–effects functions that appear in Equation (3.1).

We explicitly add the exponential decay to enforce the forgetting constraint, which is essential for most realistic processes. Although we choose here a specific parameterization of memory decay, one can choose other forms of memory decay with minimal adaptation to the learning procedure of the model parameters.

Besides α and λ^{*}, the functions $s(\cdot)$ and $g(\cdot)$ are the model's unknown parameters we want to infer from the data. We utilize a nonparametric Bayesian inference method based on defining a prior probability measure over these functions. We use a simple but still highly flexible approach by modeling the two functions independently as realizations of Gaussian random processes (GP). For completeness, we now take a small detour to present GPs shortly.

Equation (3.2) is an example of a parametric approach to modeling. The memory kernel is a certain function we wish to estimate, and we assume it has a specific structure. This structure has hyperparameters that can be inferred from the data, but the shape of the function is fixed. A Bayesian inference approach will include a prior distribution over the hyperparameters. A nonparametric approach applies a prior directly over the function rather than assuming a certain parametric form and then a prior over the parameters.

In the parametric approach, we can use a multivariate Gaussian distribution as a prior over the model parameters. In a way, Gaussian processes are an extension of the multivariate Gaussian distribution to infinite dimensional vectors, which are analogous to functions. For a function f with input x, a Gaussian Process prior means that for any finite vector of inputs $[x_1, .., x_n]$, the marginal joint distribution over the associated function values $[f(x_1), ..., f(x_n)]$ is a multivariate Gaussian

$$p(f|X,\theta) = \mathcal{N}(f|m,K) \tag{3.6}$$

with mean m and covariance matrix K. Here, θ is the vector of hyperparameters related to the mean and covariance.

As the above hold for any input x, GPs are defined using a mean function $m(x)$ and covariance function $k(x, x')$ also called *kernel*. The covariance matrix in Equation (3.6) is constructed from the GP kernel in the following way. For a given set of inputs $\{x_1, ..., x_n\}$, $K_{i,j} = k(x_i, x_j)$.

In our model, we define the memory kernel in a semi–parametric approach. While we include a specific parametrization for the memory decay, we use the nonparametric approach for $s(\cdot)$ and $g(\cdot)$ using a Gaussian process. We write symbolically:

$$s \sim GP(0, K^s) \tag{3.7}$$

$$g \sim GP(0, K^g) \tag{3.8}$$

$$K^{s/g}(t_1, t_2) = a_{s/g} \cdot \exp\left(-\frac{\|t_1 - t_2\|^2}{\sigma_{s/g}^2}\right). \tag{3.9}$$

The corresponding Gaussian prior measures are uniquely defined by the mean functions (which we set to be equal to zero) and the second moments given by the covariance kernel functions $K^{g/s}$. The prior expectations define the latter

$$K^s(t_1, t_2) \doteq \mathbb{E}(s(t_1)\, s(t_2)) \tag{3.10}$$

$$K^g(t_1, t_2) \doteq \mathbb{E}(g(t_1)\, g(t_2)) \tag{3.11}$$

By a proper choice of kernels, we can encode further prior beliefs on typical realizations of $s(\cdot)$ and $g(\cdot)$. For this model, we work with the so-called RBF kernels from Equation (3.9). This kernel corresponds to the prior assumptions that the Gaussian processes are stationary (the kernels depend on time differences only) and that the functions $s(\cdot)$ and $g(\cdot)$ are infinitely often differentiable. The kernels depend on two hyperparameters, a, and σ, which reflect the functions' typical amplitude and length scale. The reasonable values of these hyperparameters will also be inferred from the data.

Finally, we assume a prior distribution also on the upper intensity bound

$$\lambda^* \sim \text{Gamma}(\alpha_0, \beta_0).$$

and we identify the hyperparameters of the model as $\{\sigma_g, a_g, \alpha, \sigma_s, a_s\}$.

We propose Bayesian inference for fitting the model to data. Due to the nonlinearity over $\phi(\cdot)$, we are no longer able to easily utilize the branching structure of the Hawkes process, which allowed for the estimation of $s(\cdot)$ and $g(\cdot)$ in prior work [144, 191]. A natural solution is to perform the inference directly on $\phi(\cdot)$.

We can identify the prior over the entire linear intensity $p(\phi)$. From Equation (3.5), we see that the linear intensity function ϕ is nothing but the sum of GPs, and as such, it is also a GP:

$$\phi \sim GP\left(0, \tilde{K}\right) \tag{3.12}$$

$$\tilde{K}_{lk} = K^s_{lk} + \sum_{t_i<t_l}\sum_{t_j<t_k} K^g_{t_l-t_i,t_k-t_j} \exp\left(-\alpha\left(t_l - t_i + t_k - t_j\right)\right). \tag{3.13}$$

3.2.1 The Multivariate Model

We can extend the model to multiple dimensions. This is useful in applications where different events are observed or originate from different processes that affect each other. We define an R–dimensional point process with intensity in dimension r

$$\lambda^r(t) = \lambda^*_r \sigma\left(\phi^r(t)\right)$$
$$\phi^r(t) = s^r(t) + \sum_{m=1}^{R}\sum_{t^m_n<t} g_{r,m}\left(t - t^m_n\right)\exp\left(-\alpha_{r,m}\left(t - t^m_n\right)\right)$$

where t^m_n is the time of event number n of type m. We assume that every dimension has its own intensity bound λ^*_r and background rate $s^r(\cdot)$. The different dimensions interact with each other via the self–effects term. $g_{r,m}(\cdot)$ defined the effects of the events of type m on the events of type r. As in the univariate case, this effect may change over time.

Given the observations, the different dimensions are independent of each other, and we can learn their parameters separately. Thus, in the following section, we present the inference for the univariate model, and the extension to the multivariate case is straightforward.

3.3 Inference for NH–GPS

As presented in Section 2.2, conditioned on the intensity function $\lambda(\cdot)$, the likelihood of observations $\{t_1, \ldots t_N\}$ from a Hawkes process is [35]

$$\ell\left(\{t_1, \ldots t_n\}|\lambda(\cdot)\right) = \exp\left\{-\int_0^T \lambda\left(t'\right)dt'\right\}\prod_{n=1}^{N}\lambda(t_n). \tag{3.14}$$

The inference is made in the Bayesian framework. We wish to combine the priors defined in the previous section with the likelihood above and estimate the posterior via sampling or approximate inference methods. The likelihood in Equation (3.14) includes an integral that we cannot solve analytically. One approach would be approximating the integral via discretization. This would result in a biased estimator of the likelihood and introduce errors to the inference procedure.

Instead, we implement an augmentation procedure, similar to previous work on Cox and Hawkes processes [41, 3, 191]. We do so via the introduction of auxiliary variables, which expand the model to a different likelihood form. Under the marginalization of the aforementioned auxiliary variables, the new likelihood will conserve the form of the original model likelihood. The new form of the likelihood is constructed such that the computations required for the inference procedure are either tractable, computationally fast, or both.

3.3.1 Augmenting the NH–GPS Model

The full likelihood of the nonlinear Hawkes process we propose poses two challenges. First, the intractable integral that we discussed before. Second, the sigmoid link function is not conjugate to the Gaussian prior.

We address the two challenges with two sets of augmenting variables. First, we augment the model with a set of PG variables using the augmentation scheme presented in Section 2.1.2. This step results in a Gaussian representation of the set of parameters ϕ in the product that appears in Equation (3.14).

To overcome the intractable integral, we rewrite the sigmoid function and identify the integral as the characteristic function of a Poisson process. We use Campbell's theorem and augment the model with a Poisson process and a set of realizations from it. This augmentation does not result in a conjugated representation, and we augment the model with another set of PG variables that correspond to the realizations from the augmenting Poisson process.

Pólya–Gamma Augmentation

The first step we take in treating the likelihood function is using the Pólya–Gamma (PG) augmentation scheme that was introduced in Section 2.1.2. Following Theorem 1 in Polson, Scott, and Windle [142], we can rewrite the nonlinear intensity function as

$$\sigma\left(\phi\left(t\right)\right)=\int_{0}^{\infty}e^{f\left(w,\phi\left(t\right)\right)}PG\left(w;1,0\right)dw \tag{3.15}$$

$$f\left(w,\phi\left(t\right)\right)=-\frac{\phi\left(t\right)^{2}w}{2}+\frac{\phi\left(t\right)}{2}-\ln 2. \tag{3.16}$$

As we augment each observation with a variable w_n from a PG distribution, the joint likelihood of the observed events $\{t_n\}$ and PG variables $\{w_n\}$ is

$$p\left(\{t_n\}_{n=1}^{N}, \{w_n\}_{n=1}^{N} | \phi, \lambda^*\right) = \\ \exp\left(-\int_0^T \lambda^* \sigma\left(\phi\left(t\right)\right) dt\right) \cdot \prod_{n=1}^{N} \lambda^* e^{f(w_n, t_n)} PG\left(w_n; 1, 0\right) \tag{3.17}$$

with

$$\exp\left\{-\int_0^T \lambda^* \sigma\left(\phi\left(t\right)\right) dt\right\} = \\ \exp\left(-\int_0^T \int_0^\infty \lambda^* PG\left(w; 1, 0\right)\left(1 - e^{f(w, -\phi(t))}\right) dw dt\right). \tag{3.18}$$

where we used $\sigma(t) = 1 - \sigma(-t)$.

Marked Poisson Process Augmentation

Next, we utilize the Campbell's theorem [84], which states that for a Poisson process Π with intensity φ

$$\mathbb{E}_\varphi\left(\prod_{x \in \Pi} \exp\left(h\left(x\right)\right)\right) = \\ \exp\left(-\int\left(1 - \exp\left(h\left(x\right)\right)\right)\varphi\left(x\right) dx\right).$$

Looking at Equation (3.18), we identify $x = (t, w)$ and $\varphi\left(t, w\right) = \lambda^* PG\left(w|1, 0\right)$ is the intensity of a marked Poisson process in $\mathcal{T}$ with marks $w \sim PG\left(0, 1\right)$. Furthermore, we determine $h\left(x\right) = f\left(w, -\phi\left(t\right)\right)$. We can now rewrite the exponential in Equation (3.17) as

$$\exp\left\{-\int_0^T \lambda \sigma\left(\phi\left(t\right)\right) dt\right\} = \mathbb{E}_\varphi\left(\prod_{m=1}^{M} e^{f\left(\hat{w}_m, \hat{t}_m\right)}\right) \tag{3.19}$$

for realizations $\{\hat{t}_m, \hat{w}_m\}_{m=1}^{M}$.

We substitute Equation (3.19) into Equation (3.17), which results in the full augmented likelihood. Given the prior distributions over ϕ and λ^*, we can now write the augmented model's posterior distribution as

$$p\left(\{\hat{t}_m, \hat{w}_m\}, \{w_n\}, \phi, \lambda^* | \{t_n\}\right) \propto \exp\left(-\lambda T\right) \times \\ \prod_{m=1}^{M} \lambda^* e^{f\left(\hat{w}_m, -\phi\left(\hat{t}_m\right)\right)} PG\left(\hat{w}_m; 1, 0\right) \times \\ \prod_{n=1}^{N} \lambda^* e^{f(w_n, \phi(t_n))} PG\left(w_n; 1, 0\right) \times p\left(\phi\right) p\left(\lambda^*\right). \tag{3.20}$$

The augmentation described above results in a representation of the full likelihood that does not include the intractable integral that appeared in the original likelihood function in Equation (3.14). With this representation, we could use any Bayesian sampler to infer the posterior. Here, we choose to utilize the Gibbs sampler and the mean-field variational inference previously introduced in Donner and Opper [41] and Donner and Opper [40]. Next, we outline the steps of the algorithms, and we refer the reader to the two papers mentioned above for further details.

3.3.2 Gibbs sampler for NH–GPS

We use a blocked Gibbs sampler, which groups two or more variables and samples them at once, for which we need to identify the conditional posterior distribution of all the relevant groups. We group the parameters in the following way:

1. The PG variables sampled from $p\left(\{w_n\},\{\hat{w}_m\} \mid \left\{\hat{t}_m\right\},\phi,\lambda^*,\{t_n\}\right)$
2. The upper intensity bound sampled from $p\left(\lambda^* \mid \{w_n\},\{\hat{w}_m\},\left\{\hat{t}_m\right\},\phi,\{t_n\}\right)$
3. The linear intensity function estimated at the real and augmenting events sampled from $p\left(\phi \mid \{w_n\},\{\hat{w}_m\},\left\{\hat{t}_m\right\},\lambda^*,\{t_n\}\right)$
4. The set of augmenting events sampled from $p\left(\left\{\hat{t}_m\right\} \mid \{w_n\},\{\hat{w}_m\},\phi,\lambda^*,\{t_n\}\right)$.

Next, we derive the expressions for the conditional probability densities listed above.

Conditional Density of the PG Variables

The conditional posterior distribution of the augmenting PG variables is

$$\begin{aligned}
&p\left(\{w_n\},\{\hat{w}_m\}\right)\\
&\propto \prod_{n=1}^{N} \exp\left(-\frac{\phi_n^2}{2} w_n\right)\\
&\times PG\left(w_n;1,0\right) \prod_{m=1}^{M} \exp\left(-\frac{\phi_m^2}{2}\hat{w}_m\right) PG\left(w_m;1,0\right).
\end{aligned}$$

Using the definition of the tilted PG distribution [142], the posterior distribution for these parameters is

$$\begin{aligned}
w_n &\propto PG\left(1,\phi_n\right)\\
\hat{w}_m &\propto PG\left(1,\phi_m\right).
\end{aligned}$$

Conditional Density of the Upper Intensity Bound λ^*

The conditional distribution of the upper intensity bound is

$$p\left(\lambda^*|\{\hat{t}_m, \hat{w}_m\}, \{w_n\}, \phi, \{t_n\}\right) \propto e^{-\lambda^* T} \lambda^*_{N+M} p\left(\lambda^*\right)$$

which we identify as a Gamma distribution

$$\begin{aligned} & p\left(\lambda^*|\{\hat{t}_m, \hat{w}_m\}, \{w_n\}, \phi, \{t_n\}\right) \propto \text{Gamma}\left(\alpha, \beta\right) \\ & \alpha = \alpha_0 + N + M \\ & \beta = \beta_0 + T. \end{aligned} \tag{3.21}$$

Conditional Density of the Linear Intensity Function

The conditional distribution of the linear intensity function of the observed and augmenting events is

$$\begin{aligned} & p\left(\phi_{N+M}|\{\hat{t}_m, \hat{w}_m\}, \{w_n\}, \phi, \{t_n\}, \lambda^*\right) \\ & \propto \exp\left(\sum_{n=1}^{N} f\left(w_n, \phi_n\right) + \sum_{m=1}^{M} f\left(\hat{w}_m, -\phi_m\right)\right) p\left(\phi_{N+M}\right), \end{aligned}$$

where we use the shortened notation ϕ_n instead of $\phi\left(t_n\right)$ and ϕ_{N+M} instead of $\{\{\phi\left(t_n\right)\}_{n=1}^{N}, \{\phi\left(\hat{t}_m\right)\}_{m=1}^{M}\}$. Given Equation (3.16), the likelihood term in the posterior is a GP with mean

$$\mu = \left(\frac{1}{2w_1}, \ldots, \frac{1}{2w_N}, -\frac{1}{2\hat{w}_1}, -\frac{1}{2\hat{w}_M}\right)^{\top}$$

and diagonal covariance matrix

$$\Sigma^{-1} = \text{Diag}\left(w_1, \ldots, w_N, \hat{w}_1, \hat{w}_M\right).$$

Given the GP prior over ϕ the conditional posterior is also a GP

$$\begin{aligned} & \phi_{N+M} \sim GP\left(\mu_{M+N}, \Sigma_{M+N}\right) \\ & \mu_{M+N} = \Sigma_{N+M}\Sigma^{-1}\mu \\ & \Sigma_{N+M}^{-1} = \Sigma^{-1} + \tilde{K}^{-1}. \end{aligned} \tag{3.22}$$

Conditional Density of the Augmenting Events

The conditional posterior of the augmenting events is proportional to

$$\begin{aligned} & p\left(\{\hat{t}_m\}_{m=1}^{M}|\{t_n\}_{n=1}^{N}, \{\hat{w}_m\}_{m=1}^{M}, \{w_n\}_{n=1}^{N}, \phi, \lambda^*\right) \\ & \qquad \propto \prod_{m=1}^{M} \lambda^* e^{f\left(\hat{w}_m, -\phi_m\right)} PG\left(\hat{w}_m; 1, 0\right). \end{aligned}$$

We identify the posterior as the unnormalized density of a marked inhomogeneous Poisson process with intensity

$$\Pi_0\left(\hat{w},\hat{t}\right)=\lambda^* e^{f(\hat{w}_m,-\phi_m)}PG\left(\hat{w}_m;1,0\right). \tag{3.23}$$

The underlying density of the inhomogeneous Poisson process in the realizations space $\mathcal{T}$ is given by

$$\int_{\mathcal{W}}\Pi_0\left(\hat{w},\hat{t}\right)d\hat{w}=\lambda^*\sigma\left(-\phi_m\right).$$

We use the thinning algorithm [135] to sample the augmenting realizations. First, we sample the expected number of events

$$J\sim Poisson\left(\lambda^* T\right).$$

Next, J candidates are sampled uniformly over the realizations space $\mathcal{T}$. To evaluate the intensity at the candidate points, we need to evaluate the linear intensity ϕ in these points. Given its values in the real events and the previously sampled augmenting events. This can be done using results from GP regression [143]

$$\begin{aligned}
&\phi_{J|N+M}\sim GP\left(\mu,\tilde{K}\right)\\
&\mu=\tilde{K}_{J,N+M}\tilde{K}_{N+M}^{-1}\phi_{N+M}\\
&\tilde{K}=\tilde{K}_J-\tilde{K}_{J,N+M}\tilde{K}_{N+M}^{-1}\tilde{K}_{J,N+M}^{\top}.
\end{aligned}$$

Once we have ϕ_J, we perform thinning—for each candidate $\hat{t}_j$ we generate a random number r_j between 0 and 1. If $r_j<\sigma\left(-\phi_j\right)$ we accept the candidate $\hat{t}_j$, otherwise we discard it.

The Gibbs sampler is summarized in algorithm 3.

Algorithm 3: NH–GPS Gibbs Sampler

```
Input: Observed events $\{t_n\}_{n=1}^{N}$, hyperparameters $\{\sigma_g,\alpha,a_s,\sigma_s,\alpha_0,\beta_0\}$
Output: R samples of $\{\hat{t}_m\}_{m=1}^{M_r}$, $\lambda$, $\phi_{N+M_r}$
Initialize—$M,\phi_{N+M},\{\hat{t}_m\}_{m=1}^{M_r}$ randomly.
for $r\leftarrow 0$ to $R$ do
    Sample $w\sim PG\left(1,\phi_{N+M}\right)$
    Sample $\lambda^*\sim$ Gamma $\left(\alpha,\beta\right)$ as in Equation (3.21)
    Sample $\phi_{N+M}\sim GP\left(\mu_{N+M},\Sigma_{N+M}\right)$ as in Equation (3.22)
    Sample $\{\hat{t}\}_{m=1}^{M_r}$ as in Section 3.3.2
end
```

Hyperparameters Learning for the Gibbs Sampler

The Gibbs sampler described above samples from the conditional distributions of the model's parameters, given fixed hyperparameters. Naturally, the choice of the hyperparameters effects dramatically the results of the inference process, and we next discuss how we learn the model's hyperparameters.

The augmented model is not conditionally conjugated with respect to the kernel hyperparameters, and we cannot sample them directly as part of the Gibbs sampler. This is usually solved by using MCMC within the Gibbs sampler approach [59, 117]. This method applies rejection sampling, such as Metropolis–Hastings (MH) [65] and Hamiltonian Monte Carlo (HMC) [42], to sample the hyperparameters and relies heavily on the samplers' design choices. A wrong choice of the proposal distribution (for MH) or the mass matrix (for HMC) may result in a very slow convergence or prevent the sampler from converging at all.

Rather than using Bayesian inference for the hyperparameters, we use an optimization approach and take a gradient step within the Gibbs sampler. This is implemented in the following way—after sampling all the model parameters from the conditional posterior distributions described above, we derive the negative model log posterior with respect to the hyperparameters and take a step in the direction of the negative gradient.

This approach can be developed further in the spirit of stochastic gradient descent (SGD) [149]. Rather than updating the hyperparameters after each iteration of the Gibbs sampler, we perform several sampling steps, take the gradient of the averaged posterior, and update the hyperparameters following the averaged gradient.

For the SGD, we need to derive the model's posterior with respect to the hyperparameters. Looking in Equation (3.20), we notice that all of the hyperparameters appear in the prior over the linear intensity

$$\log p\left(\phi\right) \propto -\frac{1}{2}\log\det\left(\tilde{K}\right) - \frac{1}{2}\phi^{\top}\tilde{K}^{-1}\phi$$

and specifically in the prior kernel.

The derivative of an entry in the kernel with respect to the different hyperparameters is

$$
\begin{aligned}
\frac{\partial \tilde{K_{l,k}}}{\partial a_s} &= \exp\left(-\frac{\| t_l - t_k \|^2}{\sigma_s^2}\right) \\
\frac{\partial \tilde{K_{l,k}}}{\partial \sigma_s} &= a_s \exp\left(-\frac{\| t_l - t_k \|^2}{\sigma_s^2}\right) \frac{\| t_l - t_k \|^2}{\sigma_s^3} \\
\frac{\partial \tilde{K_{l,k}}}{\partial \alpha} &= \sum_{t_i < t_l} \sum_{t_j < t_k} K^g_{t_i - t_l, t_j - t_k} \\
&\times \exp\left(-\alpha\left(t_i - t_l + t_j - t_k\right)\right)\left(t_l - t_i + t_k - t_j\right) \\
\frac{\partial \tilde{K_{l,k}}}{\partial \sigma_g} &= \sum_{t_i < t_l} \sum_{t_j < t_k} K^g_{t_i - t_l, t_j - t_k} \\
&\times \exp\left(-\alpha\left(t_i - t_l + t_j - t_k\right)\right) \frac{\| \left(t_l - t_i\right) - \left(t_k - t_j\right) \|^2}{\sigma_g^3}.
\end{aligned}
$$

We can plug these results into the chain rule, and we get

$$
\nabla \log p\left(\phi\right) = -\frac{1}{2}\text{trace}\left(\tilde{K}^{-1}\nabla\tilde{K}\right) + \frac{1}{2}\phi^\top \tilde{K}^{-1}\nabla\tilde{K}\tilde{K}^{-1}\phi.
$$

3.3.3 Variational Inference

In variational inference (VI) [81, 13], we define a tractable distribution family Q and adapt it to approximate the posterior by maximizing the lower bound $\mathcal{L}(Q)$ defined below. This procedure minimizes the Kullback–Leibler divergence between the unknown posterior and the proposed approximating distribution. For the NH–GPS model, we approximate the posterior density by

$$
\begin{aligned}
&p\left(\{\hat{t}_m, \hat{w}_m\}, \{w_n\}, \phi, \lambda^* | \{t_n\}\right) \\
&\approx Q_1\left(\phi, \lambda^*\right) Q_2\left(\{w_n\}_{n=1}^N, \{\hat{t}_m, \hat{w}_m\}_{m=1}^M\right).
\end{aligned}
$$

Subsequently, the lower bound on the evidence (ELBO) is

$$
\mathcal{L}(q) = \mathbb{E}_q\left[\log\left\{\frac{p\left(\{\hat{t}_m, \hat{w}_m\}, \{w_n\}, \phi, \lambda | \{t_n\}\right)}{q_1\left(\phi, \lambda\right) q_2\left(\{w_n\}_{n=1}^N, \{\hat{t}_m, \hat{w}_m\}_{m=1}^M\right)}\right\}\right]
$$

where $q \in Q$. Here, Q refers to the probability measure of the variational posterior. We maximize the bound by alternating the maximization over each of the factors [13]. The optimal solution for each factor is

$$\begin{aligned}
&\log q_1^* (\phi, \lambda^*) = \qquad (3.24)\\
&\mathbb{E}_{q_2\left(\{w_n\}_{n=1}^N, \{\hat{t}_m, \hat{w}_m\}_{m=1}^M\right)}[\log p(\{\hat{t}_m, \hat{w}_m\}, \{w_n\}, \phi, \lambda^*, \{t_n\})]\\
&\log q_2^* \left(\{w_n\}_{n=1}^N, \{\hat{t}_m, \hat{w}_m\}_{m=1}^M\right) = \qquad (3.25)\\
&\mathbb{E}_{q_1(\phi,\lambda^*)}[\log p(\{\hat{t}_m, \hat{w}_m\}, \{w_n\}, \phi, \lambda, \{t_n\})].
\end{aligned}$$

To obtain the optimal distribution of one of the factors, we must calculate the expectations of the logarithm of the joint distribution over the remaining factors in the approximation, resulting in an iterative algorithm.

Next, we explicitly express the functional form of the optimal distributions and obtain the corresponding expectations required for updating the factors.

Optimal Q_1

We find that the optimal $q_1 \in Q_1$ is factorized as

$$q_1 (\phi, \lambda^*) = q_1 (\lambda^*) \, q_1 (\phi)$$

The first factor is identified as a Gamma distribution

$$\begin{aligned}
q_1 (\lambda^*) &= \text{Gamma} (\alpha, \beta) \qquad (3.26)\\
\alpha &= \alpha_0 + N + \int_{\mathcal{T}x\mathcal{W}} \lambda_{q_2} (t, w) \, dtdw\\
\beta &= \beta_0 + T
\end{aligned}$$

with known expectations.

The optimal distribution for the second factor is of the form

$$\begin{aligned}
q_1^\star &\propto e^{-U(\phi) + \log p(\phi)}\\
U(\phi) &= \frac{1}{2} \int A(t) \phi^2(t) dt - \int b(t) \phi(t) dt\\
A(t) &= \sum_n \langle \omega_n \rangle_{q_2^\star} \delta(t - t_n) + \langle \omega(t) \rangle_{q_2^\star} \lambda_{q_2^\star}(t)\\
b(t) &= \sum_n \frac{1}{2} \delta(t - t_n) - \frac{1}{2} \lambda_{q_2^\star} (t) \, .
\end{aligned}$$

Generally, the integrals above cannot be evaluated analytically. We resort to another variational approximation, where we approximate the likelihood term by a distribution that depends only on a finite set of inducing point $\{c\}$

$$\tilde{q}\left(\phi_c, \phi\right) = p\left(\phi|\phi_c\right) q\left(\phi_c\right).$$

The ELBO is

$$\left\langle \log \frac{e^{-\log\langle U(\phi)\rangle_{p(\phi|\phi_c)}} p(\phi_c)}{\tilde{q}(\phi_c)} \right\rangle_{\tilde{q}},$$

and we use the notation $\langle p \rangle_q = \mathbb{E}_q\left(p\right)$. The optimal $\tilde{q}\left(\phi_c\right)$ is given by

$$\tilde{q}^{\star}(\boldsymbol{\phi}_c) \propto e^{-\log\langle U(\boldsymbol{\phi})\rangle_{p(\boldsymbol{\phi}|\boldsymbol{\phi}_c)}} p(\boldsymbol{\phi}_c).$$

Using the known results of conditional GPs and sparse variational GPs [34, 174], we have

$$\begin{aligned} \tilde{q}^{\star}(\boldsymbol{\phi}_c) &= \mathcal{N}(\boldsymbol{\phi}_c|\boldsymbol{\mu}_c, \Sigma_c) \qquad (3.27) \\ \Sigma_c &= \left[\int \boldsymbol{\kappa}(t)^{\top} A(t)\boldsymbol{\kappa}(t)dt + K_c^{-1}\right]^{-1} \\ \boldsymbol{\mu}_c &= \Sigma_c\left(\int b(t)\boldsymbol{\kappa}(t)dt\right). \end{aligned}$$

K_c is the covariance kernel between the inducing points, $\boldsymbol{\kappa}(t) = \boldsymbol{k}_c(t)^{\top} K_c^{-1}$ and $\boldsymbol{k}_c(t)$ is the kernel between the inducing points and another set of points (either the real data or the integration points), such that $\boldsymbol{k}_c\left(t\right) = \left(\tilde{K}\left(t, t_1\right), \ldots, \tilde{K}\left(t, t_L\right)\right)$ with $t_1, \ldots, t_l, \ldots, t_L$ are the inducing points. The mean and the variance of the sparse approximated GP are

$$\begin{aligned} \langle \phi(t) \rangle =& \boldsymbol{\kappa}(t)\boldsymbol{\mu}_c \qquad (3.28) \\ \sigma^2\left(\phi\left(t\right)\right) =& K(t,t) - \boldsymbol{\kappa}(t)^{\top}\boldsymbol{k}_c(t) + \boldsymbol{\kappa}(t)^{\top}\Sigma_c\boldsymbol{\kappa}(t) \qquad (3.29) \end{aligned}$$

Optimal Q_2

Similarly to the previous section, we find that the optimal $q_2 \in Q_2$ is factorized as

$$q_2\left(\{w_n\}_{n=1}^N, \Pi\right) = q_2\left(\{w_n\}_{n=1}^N\right) q_2\left(\{\hat{t}_m, \hat{w}_m\}\right)$$

Given Equation (3.20), we define the first factor as

$$q_2^{\star}(w_n) \propto \exp\left(-\frac{\langle \phi_n^2 \rangle_{q_1^{\star}}}{2} w_n\right) PG(w_n|1,0),$$

which corresponds to a tilted PG distribution

$$q_2^\star(w_n) = PG\left(w_n|1, \sqrt{\langle\phi_n^2\rangle_{q_1^\star}}\right). \tag{3.30}$$

with known expectations [142].

The second factor takes the form

$$\begin{aligned} & q_2^\star(\{\hat{t}_m, \hat{w}_m\}_{m=1}^M) \\ & \propto \prod_{m=1}^{M} \exp\left(-\frac{\langle\phi_m\rangle_{q_1^\star}}{2} - \frac{\langle\phi_m^2\rangle_{q_1^\star}}{2} w_m\right) \cdot \exp\left(\langle\ln\lambda^\star\rangle_{q_1^\star}\right). \end{aligned}$$

It can be shown that this distribution corresponds to a Poisson process with an intensity function

$$\begin{aligned} & \lambda_{q_2}\left(\hat{t}, \hat{w}\right) \\ & = \exp\left(\langle\ln\lambda^*\rangle_{q_1^\star}\right) \frac{\exp\left(-\frac{\langle\phi\rangle_{q_1^\star}}{2}\right)}{2\cosh\left(\langle\phi^2\rangle_{q_1^\star}\right)} PG\left(w_m|1, \sqrt{\langle\phi^2\rangle_{q_1^\star}}\right). \end{aligned} \tag{3.31}$$

To simplify the notation, we write ϕ instead of $\phi\left(\hat{t}\right)$.

The VI algorithm is summarized in Algorithm 4.

Algorithm 4: NH–GPS Variational Inference.

```
Input: Observed Events $\{t_n\}_{n=1}^N$, hyperparameters$\{\sigma_g, \alpha, a_s, \sigma_s, \alpha_0, \beta_0\}$
Output: Mean and variance of $\phi$
Initialize.
while $\mathcal{L}$ not converged do
    Estimate the mean and covariance of the GP over the inducing points as in
      Equation (3.27)
    Estimate the mean and variance of the GP over the observations as in
      Equations (3.28) and (3.29)
    Estimate the mean intensity bound as in Equation (3.26)
    Estimate the mean and variance of the augmenting PG variables as in
      Equation (3.30)
    Estimate the intensity function over the augmenting events as in
      Equation (3.31)
end
```

Hyperparameters Tuning

The Variational Inference does not include tuning the model's hyperparameters. To achieve a better model, we wish to tune the model's hyperparameters and improve the model's likelihood. Thus, we perform one step of gradient descent with respect to the ELBO at each iteration. The derivatives of the ELBO are given in Donner and Opper [41] (Appendix F). Notice that hyperparameter training, or tuning, is an implicit form of model selection in that we are selecting among different models by evaluating the likelihoods.

3.3.4 Identifying the Background Rate and the Self-Effects Function

In some applications, we are interested in the specific shape of the background rate function $s(\cdot)$ and the self-effects function $g(\cdot)$. Next, we describe how to recover these functions from the results of the VI algorithm.

Upon the convergence of the VI algorithm, we obtained an expression for the posterior mean and variance of the linear intensity function ϕ, as described in Equations (3.28) and (3.29). It is useful to define

$$\tilde{g}(t) = g(t)\exp(-\alpha t) \tag{3.32}$$

and we similarly define

$$\tilde{K}^g(t,t') = K^g(t,t')\exp(-\alpha(t-t')). \tag{3.33}$$

We can rewrite the posterior mean in Equation (3.28) as

$$\langle\phi(t)\rangle = \boldsymbol{k}_c^s(t)^\top \tilde{\mu}_c + \sum_l \sum_{t_i<t}\sum_{t_j<t_l} \tilde{K}^g(t-t_i, t_l-t_j)\,\tilde{\mu}_c^l \tag{3.34}$$

where we defined $\tilde{\mu_c} = K_c^{-1}\mu_c$ and used the fact that $\boldsymbol{k}_c(t)$ can be written explicitly as

$$\boldsymbol{k}_c^s(t) + \sum_{t_i<t}\sum_{t_j<t_l} \tilde{K}^g(t-t_i, t_l-t_j)$$

with $\boldsymbol{k}_c^s(t)$ being the kernel K^s between data point t and all the inducing points.

From Equation (3.34) we identify the posterior mean of s and g as

$$\langle s(t)\rangle = \boldsymbol{k}_c^s(t)^\top \tilde{\mu}_c \tag{3.35}$$
$$\langle g(t)\rangle = \sum_l \sum_{t_j<t_l} \tilde{K}^g(t, t_l-t_j)\,\tilde{\mu}_c^l. \tag{3.36}$$

To identify the covariance of s and g we start with the posterior covariance of ϕ

$$\begin{aligned}\mathrm{cov}(\phi(t),\phi(t')) = \Sigma(t,t') = \tilde{K}(t,t') - \boldsymbol{k}_c(t)^\top B\boldsymbol{k}_c(t') \\ B = K_c^{-1} - K_c^{-1}\Sigma_c K_c^{-1}.\end{aligned} \tag{3.37}$$

Using again the explicit form of $\boldsymbol{k}_c(t)$ we rewrite the equation above as

$$\Sigma\left(t,t'\right) = K_t^s + \hat{K}_t^g - K_{tc}^{s\top} B K_{tc}^s - 2K_{tc}^{s\top} B \hat{K}_{tc}^g - \hat{K}_{tc}^{g\top} B \hat{K}_{tc}^g \tag{3.38}$$

where $K_t^{s/g}$ is the kernel matrix between the data points, $K_{tc}^{s/g}$ is the kernel between the data points and the set of inducing points, and we defined

$$\hat{K}_g\left(t,t'\right) = \sum_{t_i<t} \sum_{t_j<t'} \tilde{K}_g\left(t-t_i, t'-t_j\right).$$

From the expression above, we can identify the marginal covariances of s and g separately and their joint covariance. To sample s and $\tilde{g}$ from their posterior distribution, we would need the full expression of the covariance, including both the marginal and cross covariances. For analyzing the model's ability to recover s and $\tilde{g}$, we are interested in the marginal covariances, which can be expressed as

$$\begin{aligned}
\operatorname{cov}\left(s\left(t\right), s\left(t'\right)\right) &= K_t^s - K_{tc}^{s\top} B K_{t'c}^s \qquad (3.39)\\
\operatorname{cov}\left(\tilde{g}\left(t\right), \tilde{g}\left(t'\right)\right) &= \tilde{K}_t^g - \sum_{l,m} \sum_{\substack{t_j<t_l \\ t_i<t_m}} \tilde{K}_g\left(t, t_l - t_j\right) B_{l,m} \tilde{K}_g\left(t', t_m - t_i\right).
\end{aligned}$$

3.4 Inference Analysis for NH–GPS

All the algorithms and experiments presented in this section and the next section are implemented in Python and available online under https://github.com/noashin/NHGPS. We used the JAX package [20] to parallelize the computation over the available computing resources. The PG variables were sampled using the PyPólyaGamma package [104] in the Gibbs sampler.

To assess the performance of the inference algorithms presented in the previous section, we learn the parameters of the data generated by the model, namely the intensity function and the intensity bound, and compare them to the known ground truth. To generate data, we start by sampling the memory and background GP, based on Equations (3.7) and (3.8). We generate events from the model using Poisson thinning [103]. First, we sample the number of candidates $J \sim Poisson\left(\lambda T\right)$, and sample candidate events $\{t_1, \ldots t_J\}$ uniformly. Next, we chronologically iterate through the candidates and accept them with probability $\frac{\lambda(t_j|\{t_1,\ldots t_{j-1}\})}{\lambda}$.

3.4.1 Inference for Data Generated by the Model

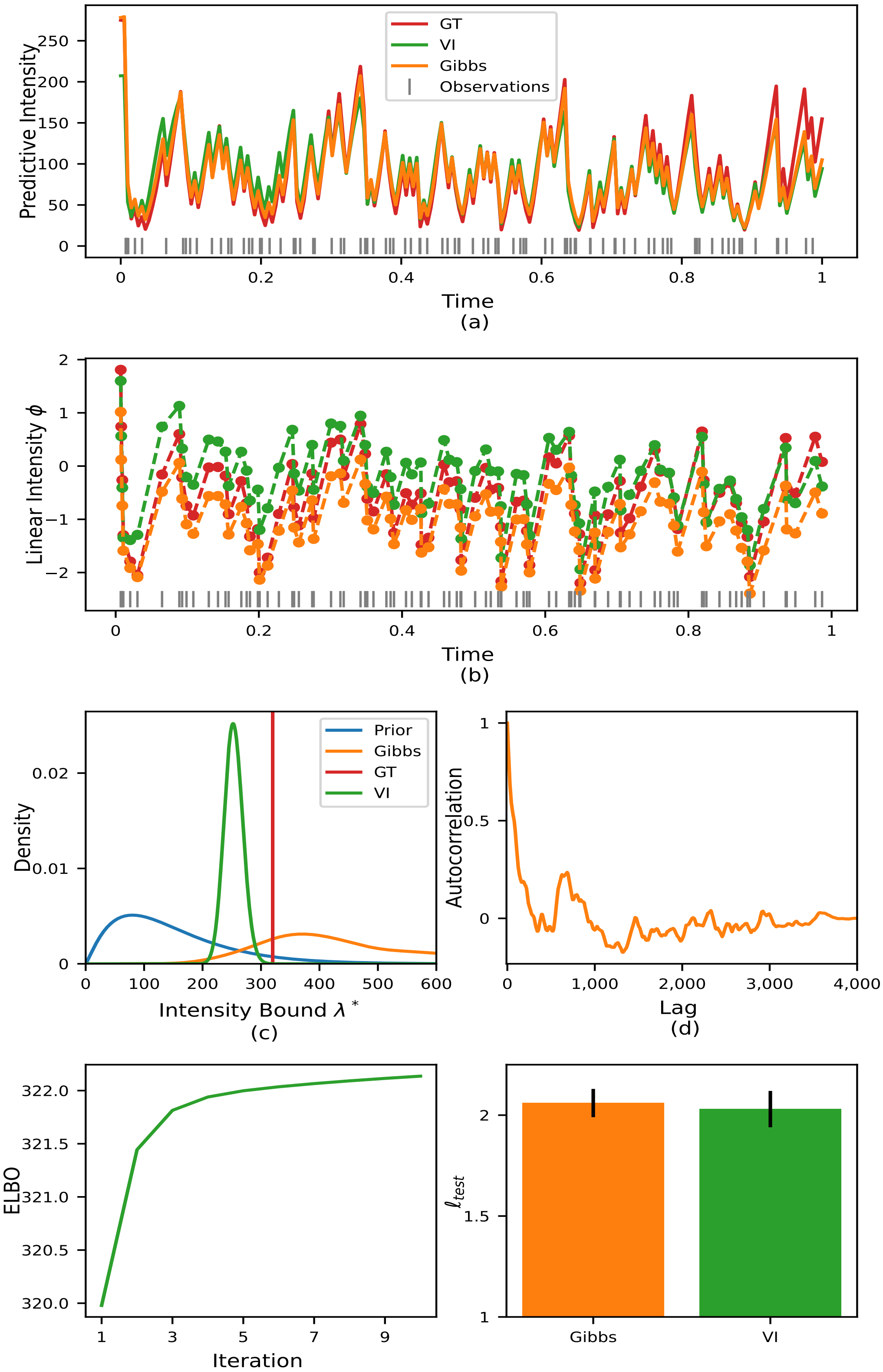
The results for the synthetic data are included in Figure 3.1. The time window was one second, and the dataset includes 91 events. In panel (a), the comparison between the ground truth predictive intensity and the one inferred by the learning algorithms demonstrates the accuracy of the inference methods. We compare the ground truth to the mean of the Gibbs samples and the mean of the approximating distribution of the VI.

Panel (b) compares the ground truth value of the linear intensity and the one inferred by the two learning algorithms. In this example, the linear intensity is negative at some of the time points. Unlike the predictive density in panel (a), the Gibbs sampler is more accurate than the VI when estimating the linear intensity. Panel (c) presents the inference results for the intensity bound. As expected, the approximated distribution by the VI is much more narrow than the distribution of the Gibbs samples. Nevertheless, the peak of the distribution of Gibbs samples is close to the ground truth.

Panels (d) and (e) show the autocorrelation of the intensity bound and the ELBO through the Gibbs samples and VI iterations. In this example, the convergence of the ELBO is very fast, and the autocorrelation of the Gibbs sample vanishes only after a few thousand iterations. We use the test log–likelihood per data point, averaged over ten datasets, to quantify the performance of the two inference algorithms. The Gibbs sampler and the VI achieve very similar results.

3.4.2 Inference for Data Generated with Different Background Rate and Self–Effects

In the experiment presented above, the data were generated from the model, where s and g were sampled from a GP. When dealing with real data, we cannot expect that the data follow the model exactly. To demonstrate that our model can fit data that were not sampled from it directly, we infer the intensity when the data is generated from a slightly different model. In this case we take $s(t) = \beta_1 \cos(\theta_1 t)$ and $g(t) = \beta_2 \cos(\theta_2 t)$. The results can be found in Figure 3.2. We perform the same analysis presented in Figure 3.1. Similarly, the Gibbs sampler and the VI algorithm accurately recover the underlying intensity. Both inference methods achieve comparable results in terms of log–likelihood averaged over 10 test datasets. As the VI algorithm achieves similar results to the Gibbs sampler in a much faster computation time, we present the results only for the VI algorithm in the next sections.



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 3.1.: Inference results for data generated from the model. (**a**) Comparison of the ground truth predictive intensity and the one sampled from the VI and Gibbs inference. (**b**) Comparison of the ground truth linear intensity $\phi(\cdot)$ and the one learned by the VI and Gibbs sampler. (**c**) Comparison of the ground truth intensity bound and the one learned by the inference and the prior distribution. (**d**) The autocorrelation of the intensity bound Gibbs samples. (**e**) The variational lower bound as a function of the algorithm iteration. (**f**) Comparison of the test log-likelihood of the Gibbs sampler and the VI.

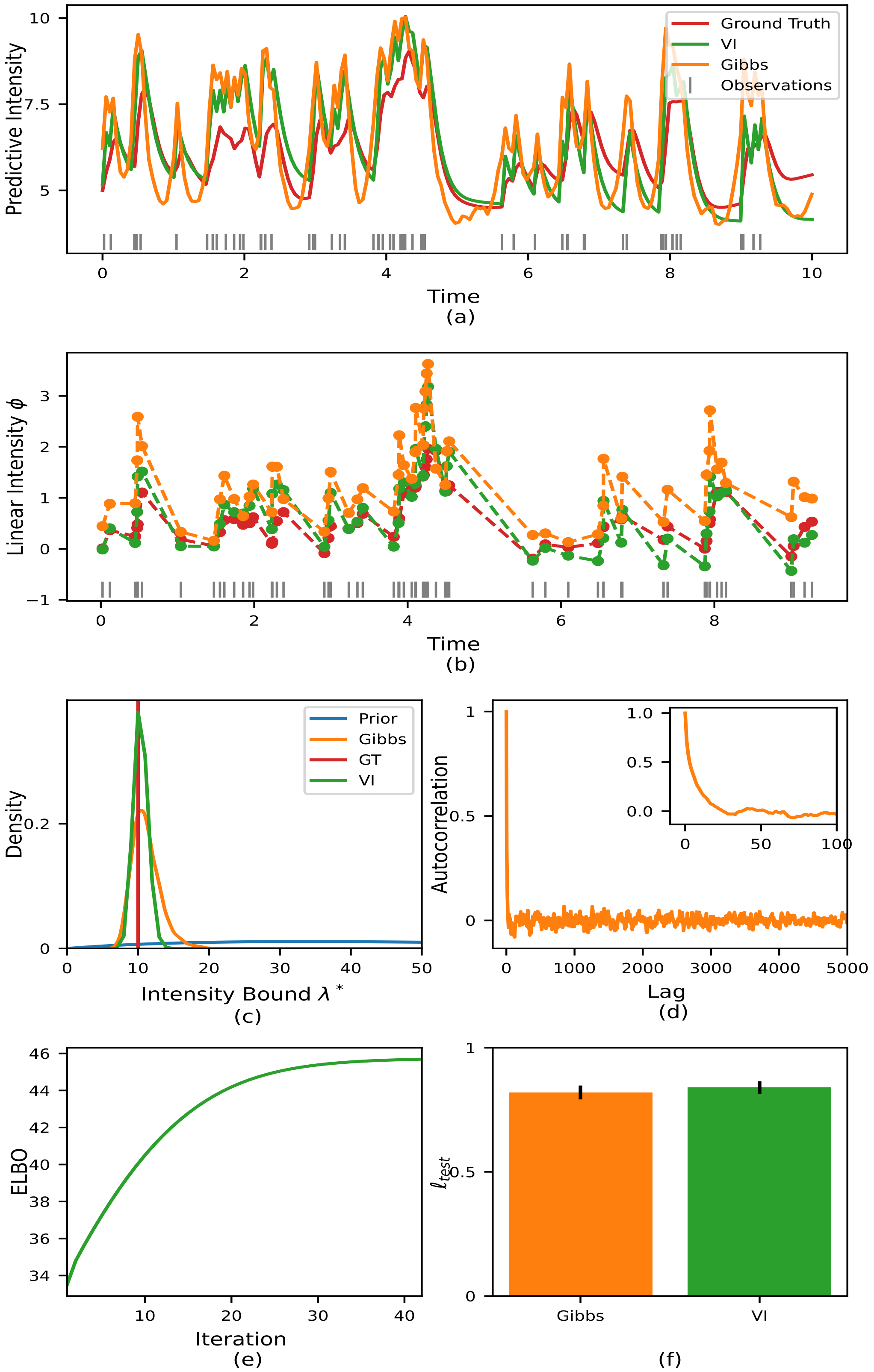


Fig. 3.2.: Inference results for data generated from a modified model. (**a**) Comparison of the ground truth predictive intensity and the one sampled from the VI and Gibbs inference. (**b**) Comparison of the ground truth linear intensity $\phi(\cdot)$ and the one learned by the VI and Gibbs sampler. (**c**) Comparison of the ground truth intensity bound and the one learned by the inference and the prior distribution. (**d**) The autocorrelation of the intensity bound Gibbs samples. (**e**) The variational lower bound as a function of the algorithm iteration. (**f**) Comparison of the test log-likelihood of the Gibbs sampler and the VI.

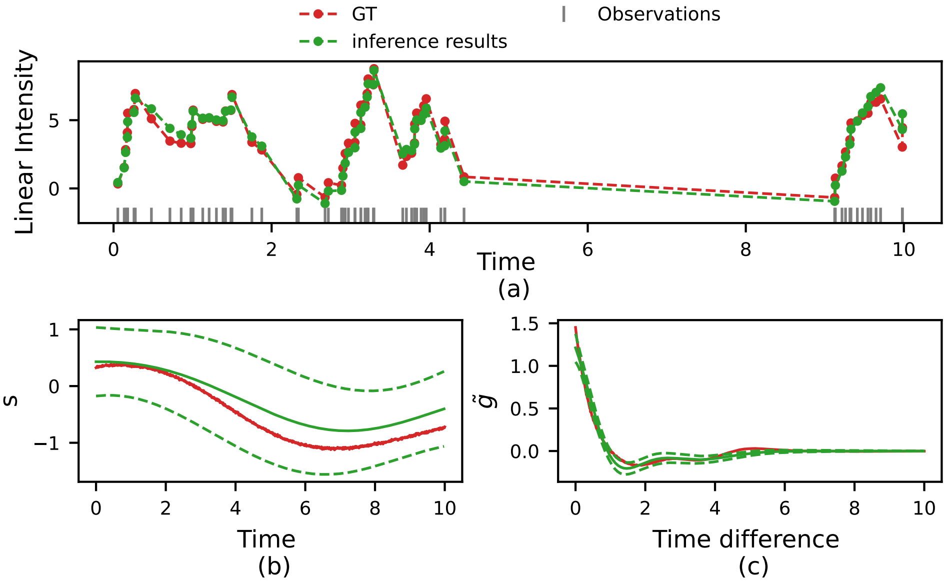


Fig. 3.3.: (**a**) Comparison of the ground truth linear intensity and the one inferred by the VI algorithm. (**b**) Comparison of the ground truth background rate function s and the one inferred by the VI algorithm. (**c**) Comparison of the ground truth self-effects function g and the one inferred by the VI algorithm.

3.4.3 Recovering the Background Rate and Self–Effects

In Section 3.3.4, we discussed the identifiability of the background rate function and the self-effects function from the inferred linear intensity. In Figure 3.3, we present the results of inferring these functions from synthetic data. Both functions are recovered from the inference results of the linear intensity. In panel c, we can see that $\tilde{g}$ goes to zero for longer time differences, as expected from the integration of the exponential decay into the self-effects kernel.

3.5 NH–GPS Model for Real World Data

Next, we apply the model and the inference algorithm to real-world datasets in the field of neuroscience and crime prediction. In these examples, the data are time series of events. We feed these events to the model and perform inference to estimate the underlying intensity function. The inferred intensity function can be used in different ways. For one, we use it to estimate the model's performance (by calculating the log-likelihood or other metrics) and compare it to competing models. We also use it to simulate data from the model, which helps us assess whether the model is a reasonable candidate to describe the data (see Section 3.5.2). In the case of multivariate data, we use the inferred intensity function to assess the interaction between the different data components (see Section 3.5.3).

3.5.1 Crime Report Data

Our model assumes both inhibitory and excitatory self–effects, but it should also be able to capture phenomena where only one of the two types of effects exists. To test this, we fit our model to crime report data, where it is assumed that past events have an excitatory effect on future events [125].

In criminology, crimes are often viewed as clustered events. For example, burglars will repeatedly attack nearby targets to take advantage of known vulnerabilities, and gang-conflict shootings may incite or encourage retaliatory violence from the opposing gang in the local territory of the rivals. The nature of such retaliatory events and the exploitation of resources by burglars are highly random, and context-based phenomena since this will depend on the criminals, location, or gang at hand. Hence, a highly flexible approach is required, and our methodology can express the inhomogeneity of each crime pattern through the unknown background rate and self–effects functions.

We use the two datasets described in Zhou, Li, Fan, Wang, Sowmya, and Chen [191] and follow their data processing procedure. Each dataset contains one type of crime, and we use the univariate version of the model. An example of the results of the inference process is presented in Figure 3.4. It includes the inferred linear intensity of the fitted model over the first 80 events in the Vancouver crime dataset.

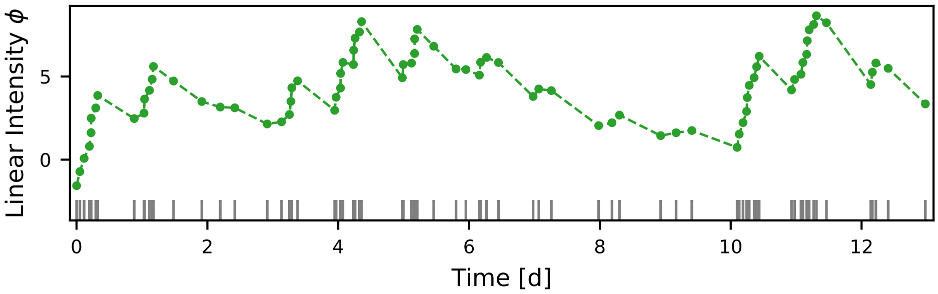


Fig. 3.4.: Inferred linear intensity ϕ for the first 80 events in the Vancouver crime dataset. The linear intensity is estimated only for the observed events, and the dashed line between the dots is merely linear interpolation.

Table 3.1 compares the test log-likelihood of our NH–GPS model to the one reported in Zhou et al. [191]. The work of Zhou et al. [191] includes several inference methods. We compare our results to the results of their reported mean-field variational inference approach as it is the closest to our inference procedure. We perform the experiment five times and report the mean and variance of the test log-likelihood. As expected, our model performs similarly to the semi-parametric Hawkes process presented by Zhou et al. [191].

Tab. 3.1.: Crime report data test log-likelihood.

Datasets	Zhou et al. (2020) [191]	NH–GPS
Vancouver	453.11 ± 8.94	453.8 ± 12.2
NYPD	-200.7 ± 3.32	-202.8 ± 7.54

3.5.2 Neuronal Activity Data

One of the motivating real-world phenomena behind our model is the spiking activity of neurons, where it is known that the process has both self–excitatory and self–inhibitory effects. A reliable model that accurately captures the data helps identify and analyze different physiological mechanisms that drive neuronal activity.

As an example of our model's ability to capture neuronal activity, we use the datasets first presented in Gerhard et al. [57] (Figure 2b,c). One dataset includes ten recordings from a single neuron in a monkey cortex, with a duration of one second each. The other consists of ten recordings from a single neuron in a human cortex for ten seconds each. This chapter used point process generalized linear models to fit the data, and the generated data yielded unrealistic spiking patterns. This hints at the need for a nonlinear model to describe the data.

The dataset described above was further analyzed in Apostolopoulou et al. [3] (Figure 5), where the mutually regressive point process (MR–PP) is introduced. The fitted MR–PP model produced realistic spike trains, which we use as a comparison for our model.

We fit the model to the multitrial single neuron datasets and infer the intensity of the assumed underlying point process. Figure 3.5 presents each dataset's results for one trial. In both cases, the inferred linear intensity obtains positive and negative values, implying excitatory and inhibitory spiking patterns.

In Figure 3.6, we assess the ability of the model to capture the data. The left column includes the raster plot of the real data, and the middle plot is the raster plot generated from the fitted model. Similarly to the real data, the generated data displays excitation as clustered events and inhibition.

To quantify how suitable the model's suitability to the data, we apply the random time change theorem [35] to the inferred intensity and the experimental data. The theorem states that realizations from a general point process can be transformed into realizations from a homogeneous Poisson process with a unit rate. Similarly to the work of Apostolopoulou et al. [3], we further transform the exponential realizations to those from a uniform distribution, following Brown et al. Brown, Barbieri, Ventura, Kass, and Frank [25]. We then use the Kolmogorov–Smirnov test to compare the quantiles of the transformed realizations' distribution to the uniform distribution quantiles. The results of this test are displayed in Figure 3.6 in the right

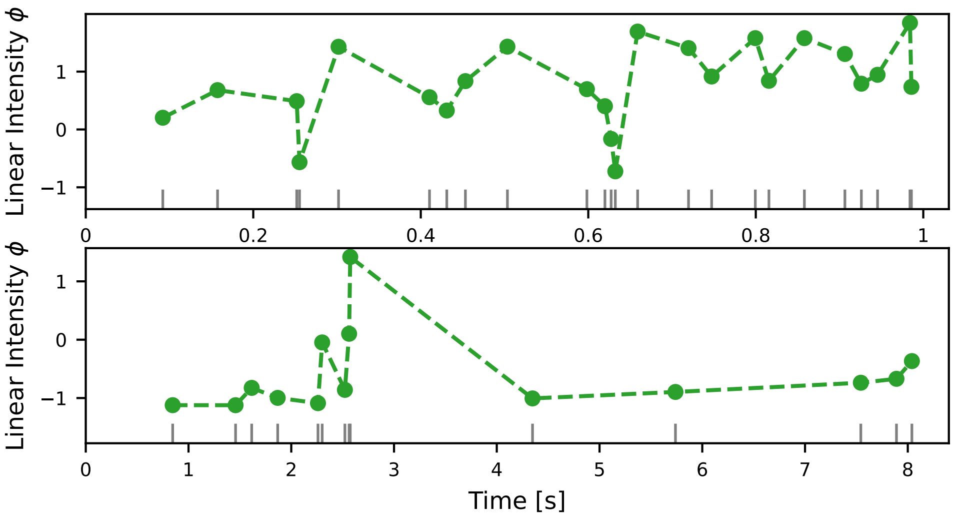


Fig. 3.5.: Inferred intensity for single neuron data. Upper panel—results for trial number 9 from the dataset recorded from monkey cortex. Lower panel—results for trial number 7 from the dataset recorded from the human cortex. The linear intensity is estimated only for the observed events, and the dashed line between the dots is merely linear interpolation.

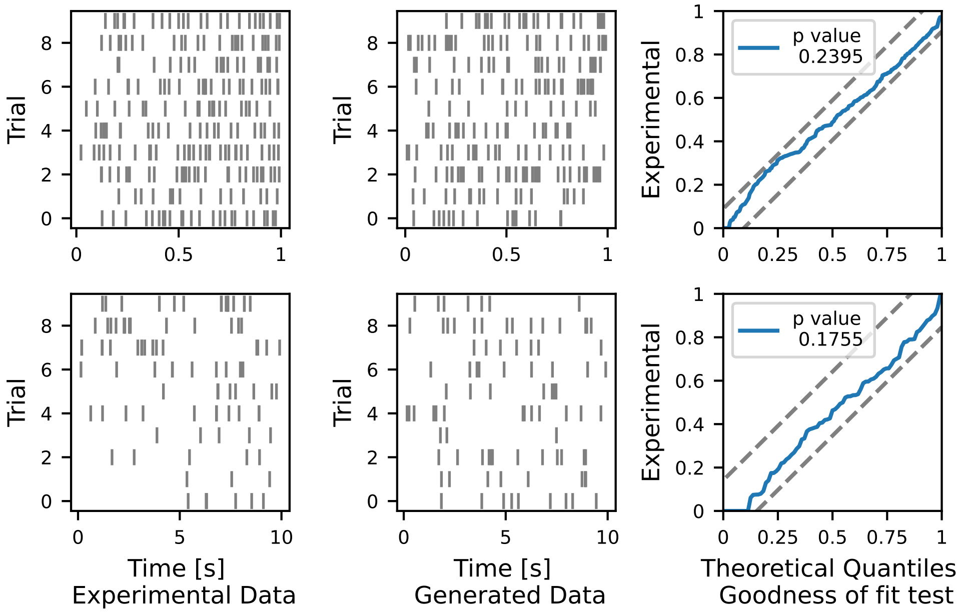


Fig. 3.6.: **Upper row—** single neuron recordings from monkey cortex. **Lower row—**single neuron recordings from the human cortex. **Left column—**recorded data. **Middle column—**data generated from the learned models. **Right column—** results of the Kolmogorov–Smirnov test. The NH–GPS generates data that resembles the real data and passes the goodness of fit test.

column. The comparison relies on 95% confidence bounds, which the dashed lines indicate. The model passes the goodness-of-fit test (p-value > 0.05). We present the reported p-value achieved by the MR–PP model for the two datasets in Table 3.2.

Tab. 3.2.: p-Value of the KS Test on neuronal activity data.

Datasets	MR–PP	NH–GPS
Monkey Cortex	0.103	0.23
Human Cortex	0.096	0.175

3.5.3 Multi-Neurons Data

Last, we demonstrate the performance of the multivariate version of the model. We use the data presented in Zhou et al. [192]. This dataset includes spike trains simultaneously recorded from 25 neurons in the primary visual cortex of an anesthetized cat.

One application of our model for the use case of multi–neuron recording use is the analysis of the interactions between the different neurons. As presented in Section 3.3.4, after fitting the model to the data, we can recover the function g, which describes the effects of past events on future events. In the case of a multivariate model, this function is defined for every pair of dimensions and the interaction between them. Figure 3.7 shows an example of this. On the left is the recorded data, and on the right are the interactions between two neurons from the dataset.

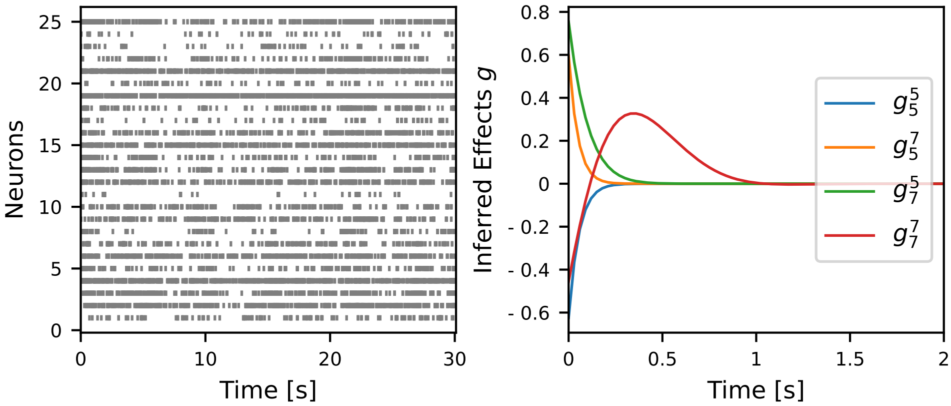


Fig. 3.7.: **Left**—the recorded activity of 25 neurons. **Right**—example of the recovery of the influence function g between neuron number 5 and neuron number 7. Both excitatory and inhibitory influence is observed.

Next, we compare the performance of our model to the model described in Zhou et al. [192] by looking at the test log-likelihood. We follow the same train–test split as Zhou et al. [192], where the first 30 seconds of the recording are used as the training set and the last 30 seconds are used as the test set.

The test log–likelihood can be found in Table 3.3. We compare our model with the one from Apostolopoulou et al. [3] (MR–PP) and the model presented in Zhou et al. [192] (SNMHP). Our model achieves a higher test log–likelihood score than the competing models. This demonstrates the power of the flexibility of our model and showcases its applicability to multivariate data.

Tab. 3.3.: Test log–likelihood for the multi-neurons datasets for different models.

SNMHP	**MR–PP**	**NH–GPS**
-6.13×10^3	-5.9×10^3	-5.29×10^3

3.6 Discussion

This chapter presents the nonlinear Hawkes model with Gaussian process self-effects (NH–GPS) and a Bayesian variational inference scheme to infer its parameters. We motivated the development of the new model with the need for a flexible model that can capture both exciting and inhibiting interactions between events while maintaining the ability to learn when data are scarce. The model includes both a univariate and a multivariate version.

Due to the structure of the model, we derive an inference algorithm without the branching structure commonly used for Bayesian inference in Hawkes processes. We propose a mean-field variational inference algorithm, which relies on an augmenting data scheme. We show that the results of the variational inference are comparable with those of a Gibbs sampler.

The augmenting data scheme used here triples the number of the model's parameters to be inferred. Another approach to the inference is discretizing the integral that appears in the likelihood and using another MH sampler, such as HMC. On the one hand, discretization results in a biased estimator of the likelihood and introduces an error to the inference. On the other hand, the VI algorithm presented in this work also includes a discretization, and it performs as well as the Gibbs sampler. Thus we believe that an HMC sampler could be successfully applied to our model and leave its implementation for future work.

In this chapter, we demonstrate the performance of our model in four different real-world applications. Due to the flexibility of our model, it achieves good results on data where events have only excitatory effects and on data where events have both excitatory and inhibitory effects.

Unlike recent work on nonlinear Hawkes process inference [192], our model presents a two-fold advantage. The introduction of the Gaussian Process allows for a broader range of applications and the Bayesian perspective, which lends itself to uncertainty quantification, model selection, and regularization. Furthermore, previous works are tailored with specific applications in mind, whereas our methodology is general.

More importantly, the introduction of priors as well as GP allows the practitioner to quickly introduce expert knowledge, as one can modify the behavior of the self–effects by introducing different kernels.

In this chapter, we did not include results regarding the prediction abilities of the model, as we leave it for future work. We believe it is relatively simple to acquire in our model, and we briefly describe how to do so. The inference over the aggregated history function ϕ, which includes both the sum of the Gaussian process associated with the exogenous arrival rate and the endogenous event rate, allows us to generate estimates for predictions. Once we sample the value of ϕ conditioned on the previously observed events, we can estimate the mean arrival time of the next event by estimating the integral $\mathbb{E}\left(t_{n+1}\right) = \int_0^\infty tP(t|\mathcal{H})$ via numerical methods, such as Monte Carlo integration.

4 Bayesian Model of Exploration and Exploitation in Natural Scene Viewing

In the previous chapter, we presented a general point process. It can be applied to data from different domains as long as they follow the assumption that events in the past affect the occurrence rate of future events. This model only includes one hypothesis regarding the nature of the data — that the effect of the past events decays with time.

In this chapter, we develop a model following a different approach. We begin with a specific domain we would like to contribute to. Then, we assemble hypotheses that we believe could explain the data in this domain and that we would like to put to test. Next, we develop a model that captures these assumptions and an inference process to fit it to real–world data. Last, we compare different versions of the model, which corresponds to the hypotheses we wish to test.

The domain we choose is natural scene viewing by human observers. Similar to the work done in the previous chapters, the datasets that characterize scene viewing are series of point events, both in time and space. Each data point is a fixated location in an image known as a *fixation*.

We develop a generative model for a series of fixations (*scan path*). The hypotheses we wish to test with this model concern the decision process underlying the generation of scan paths. In a nutshell, the innovation of our model is the assumption that the next fixation location is the result of two competing processes: 1. *exploitation* of the information in the area close to the current fixation. 2. *exploration* of other areas in the image. Furthermore, we contribute to the small family of Bayesian scan paths models and use inference methods that are not common in the field.

We begin this chapter with a short introduction to human vision and its modeling. Then, we present our new model and the innovative inference algorithm. Next, we present the results of the inference process on both generated and real data and compare different versions of the model. We conclude this chapter with a short discussion of our main findings.

4.1 Modeling Human Vision

The human visual system acquires high–acuity information from a rather small region (the fovea) surrounding the center of gaze [28]. The foveal organization of the visual system has two immediate consequences. First, visual perception of natural scenes depends critically on controlling precise and fast eye movements (saccades) that move regions of interest into the fovea for high–acuity processing. During a typical visual task (e.g., scene viewing or reading), saccades occur at a rate of 3 to 4 per second [48]. Second, the decision process for an upcoming saccade target poses a dilemma: should the observer further exploit the information near the fovea or continue exploring other patches within the given scene? The latter problem is critical for scene viewing [50, 43] and relevant to the broader field of cognitive processes in knowledge acquisition [8].

Observers select saccade targets from a priority map [14] that represents objects and regions within a given scene according to their attentional weight. Over the last decades, computational modeling of visual attention for natural scenes [78] resulted in a broad range of successful models [17] of priority maps. These models use feature maps to combine low-level saliency and top-down control. Recently, deep neural network (DNN) models achieved state–of–the–art performances in predicting saliency maps from images [91, 92]. From these advances, the problem of modeling priority maps seems basically solved [44, 93]: for an arbitrary natural image, computational models can generate a prediction of fixation density in experiments with human observers.

The next step in modeling human visual behavior is fundamentally related to eye movements introducing sequential steps in information processing. Since access to visual information is effectively limited to the fovea, the full sequence of saccadic gaze shifts, called *scan path*, needs to be modeled to understand the underlying principles. Understanding how human observers shift their attention while looking at an image requires quantifying the scan paths (Fig 4.1).

So far, few models for scan path generation and prediction have been proposed. These models can be generally classified into two groups, where one group of models is hypothesis–based and the other is hypothesis–free. The second group includes models using state of the art deep learning techniques [157, 94]. While these models capture the structure present in the data, they provide only very limited insights into the underlying principles of scan path generation. Another critical point for experimental research is that deep learning models require a lot of training data, which are typically unavailable for single observers. Thus, current deep learning approaches do not capture inter–individual differences in the statistical properties of scan paths.

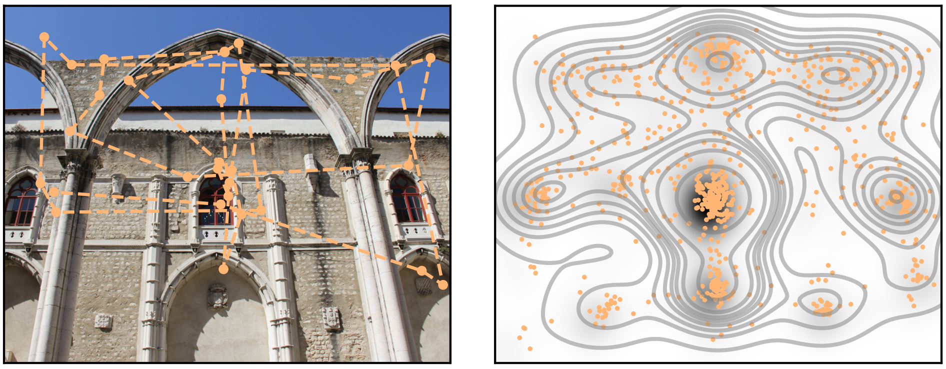


Fig. 4.1.: Left. An image and a scan path. Each dot is a fixation, and the dashed line illustrates the saccade. Right. The empirical fixation density map is generated by aggregating the fixations from all subjects for a given image.

Hypothesis–based models rely on cognitive and neural assumptions of human perception and oculomotor control that were derived from the known biological mechanisms and well-established experimental effects [170, 187, 100, 46, 168]. Thus, the key goals of the hypothesis–based models are (i) to implement these assumptions in a fully quantitative way and build a generative model, (ii) to fit the model to experimental data for hypothesis testing (statistical inference), and finally, (iii) to provide explanations for interindividual differences in experimental data sets [152].

4.2 Exploration Exploitation Model for Scene Viewing

In this chapter, we introduce a new model which belongs to the class of hypothesis–based models. As stated above, we do not model the construction of a priority map. Rather, we address the question of how saccades are generated given a specific static priority map and use the experimental priority map as input to our model. Our central hypothesis is that the generation of scan paths is based on switching between two internal states of local versus global attention. In this view, the generation of each saccade is a decision process, where the observer has to choose between following the local attention map and performing a short saccade for staying in the immediate surrounding of the current fixation and the global attention map and performing a long saccade to explore a new region of the visual environment. We assume that the decision is based on the available information to the observer. Specifically, this assumption creates a higher probability of following the local attention map if the ratio of priority values of the current and previously fixed locations is high. This hypothesis follows an assumption that the area next to a location with high priority also has high priority. This assumption is valid for natural images used in this work.

In implementing a model which changes between local and global attention mode, we continue the line of work that started already in 1976 with the work of Frost and Pöppel [49]. In Unema, Pannasch, Joos, and Velichkovsky [176] and Helmert, Joos, Pannasch, and Velichkovsky [72], it was argued that global attention mode is limited to the beginning of the scan path. Later work by Tatler and Vincent [170] showed that this is not the case. Thus our model allows the choice between a local or global attention policy throughout the entire viewing period.

Our model might also be interpreted in the context of the Exploration–Exploitation dilemma [30, 8]. In this framework, a decision for choosing an upcoming saccade target is based on two alternatives: Should the observer further exploit the information near the current gaze position or continue exploring other patches within the given scene? Hence, a saccade generated by the local attention map may correspond to an exploitation step, and a saccade generated by the global attention map may correspond to an exploration step. This approach does not consider saccades that return to a previously visited location and could be interpreted as an exploitation step, which is in line with our model of this phenomenon.

The idea of exploration and exploitation intentions in visual behavior was studied previously in Gameiro, Kaspar, König, Nordholt, and König [50]. Their work demonstrated experimentally that the tendency for exploration or exploitation, measured by saccade amplitude and fixation duration, depends on the size and spatial properties of the stimulus. The exploratory or exploitative tendencies were characterized using the statistics of the entire scan paths.

Unlike the approach taken by Gameiro, Kaspar, König, Nordholt, and König [50], we analyze the individual saccades rather than entire scan paths. Our generative model tags each saccade as either a step that follows the local attention map or a step that follows the global map.

We model natural scene viewing, which cannot be directly associated with a reward. Thus our model does not have the notion of value when choosing which policy to follow. As the terminology of Exploration and Exploitation is often associated with the notion of the value of the decision, we avoid this terminology and use the terminology of Local and Global Attention policies rather than Exploration and Exploitation policies.

Next, we describe the details of our basic model and explain the computation of the likelihood function as a fundamental tool for statistical inference. We construct the model modularly and relate each part to one of the assumptions we would like to investigate. Next, we describe the model parameters' fitting to experimental data. In the next chapter, we compare several statistics of simulated data to the statistics of the experimental data. We also analyze different variants of the basic model and quantify how well each one of them describes the data using the model's likelihood function.

4.2.1 The Local and Global Attention Model for Scan Path Generation

Our theoretical investigation of local and global attention in saccadic behavior is based on implementing a probabilistic generative model. The static viewer independent priority map for saccadic selection Bisley and Mirpour [14] is thought to be the combined result of early visual processing or *saliency* [78] and top–down cognitive control. While various models for the computation of static priority maps exist, we extend the modeling approach to generating scan paths for a given static saliency map. For simplicity, we use the time-averaged fixation density [46] as an approximation of the saliency of a given image.

The static saliency map is a function $s(z) : \mathbb{R}^2 \mapsto \mathbb{R}^+$ with $z = (x, y)$ being a location in an image and $s(z)$ being the probability of an average viewer to fixate this location (its saliency). As mentioned above, we approximate the saliency map by the experimentally–observed fixation density, and we use $s(z)$ or s_z to refer to the saliency map or the fixation density of the image at location z.

Generally, scan paths are sequences of fixation locations and fixation duration. In this chapter, we model only the spatial properties of gaze control. We account only for the temporal ordering of the fixations and do not model the fixation duration. In these settings, a scan path is written down as $Z = \{z_1, z_2, ..., z_t, ..., z_T\}$ with T being the number of fixations in the scan path and z_t being the location of the tth fixation. We discuss some aspects of spatiotemporal models for scene viewing in the concluding chapter of this dissertation in Section 6.2.2.

We begin constructing our model by assuming that the saccade generation process is a second–order Markov process. This means that the probability $p(z = z_t)$ of fixating on a specific location z_t at time step t depends only on the location of the fixation at time $t - 1$ and the fixation at time $t - 2$. The probability of a full scan path is written as

$$p(Z) = p(z_1)\, p(z_2) \prod_{t=3}^{t=T} p(z_t | z_{t-1}, z_{t-2})\, . \tag{4.1}$$

The choice of the second–order Markov process reflects our hypothesis regarding the scan path generation and will become clear in the upcoming paragraphs. In principle, it is possible to construct a simpler model corresponding to first–order Markov process. This would correspond to slightly different assumptions regarding the scan path generation, and we refer to such a model in the section discussing simplified models.

We describe the probability of the next fixation being z_t given that the previous two fixation location were z_{t-1} and z_{t-2} in terms of competing local and global attention policies:

Local Attention The next fixation location is chosen close to the current fixation location following a Gaussian distribution around the current fixation location with covariance ϵ, normalized over the entire image. This can be written as

$$p_{\text{local}}\left(z_t|z_{t-1}\right) = \frac{n\left(z_t; z_{t-1}, \epsilon\right)}{\sum_{z'} n\left(z'; z_{t-1}, \epsilon\right)} \tag{4.2}$$

where $n\left(z_t; z_{t-1}, \epsilon\right)$ is a Gaussian density with mean z_{t-1} and covariance $\epsilon = \left(\begin{smallmatrix}\epsilon_x & 0\\ 0 & \epsilon_y\end{smallmatrix}\right)$.

Global Attention A potential implementation is that the next fixation location is chosen randomly from the static saliency map of the image. This policy leads to very large saccade amplitudes, which are known to be less probable [167]. To integrate this prior regarding the saccade amplitudes knowledge into the model – instead of choosing the next fixation location from the saliency map, we modulate the saliency map by a Gaussian distribution, which gives a higher weight to areas of high saliency that are closer to the current location.

This approach results in the following expression for the global attention strategy

$$p\left(z_t|z_{t-1}\right) = \frac{s\left(z_t\right) n\left(z_t; z_{t-1}, \xi\right)}{\sum_{z'} s\left(z'\right) n\left(z'; z_{t-1}, \xi\right)} \tag{4.3}$$

with ξ a diagonal covariance matrix similarly to ϵ, $\xi_x > \epsilon_x$ and $\xi_y > \epsilon_y$.

Having Equation (4.3) as the global attention policy may result in short saccades similar to the ones generated by the local attention policy when the current fixation is in a high–priority area. A solution is to create a repulsion mechanism that forces the saccades generated by this policy to be of at least a certain length. The following expression achieves this

$$p_{\text{global}}\left(z_t|z_{t-1}\right) = \frac{\max\left(s\left(z_t\right) n\left(z_t; z_{t-1}, \xi\right) - n\left(z_t; z_{t-1}, \epsilon\right), 0\right)}{\sum_{z'} \max\left(s\left(z'\right) n\left(z'; z_{t-1}, \xi\right) - n\left(z'; z_{t-1}, \epsilon\right), 0\right)}. \tag{4.4}$$

To avoid negative values for the likelihood, we take the maximum between the subtraction and 0. Figure 4.2 visualizes the two distributions formulated in Equation (4.2) and Equation (4.3).

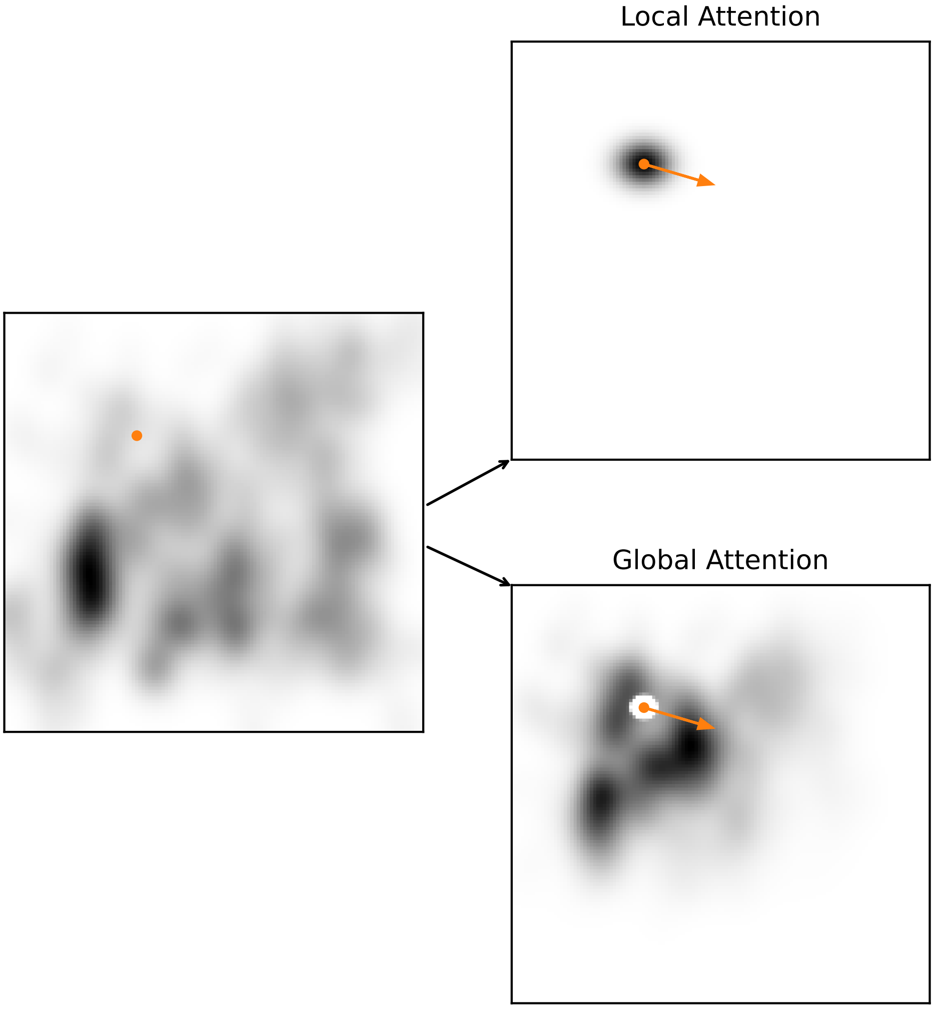


Fig. 4.2.: On the left is an example of an empirical saliency map. The dot indicates a fixation location. On the right are the probability maps generated by either the local attention (upper panel) or the global attention policy (lower panel). The arrow indicates a saccade.

Our assumption is that each fixation is chosen either from the local attention map described in Eq (4.2) or the global attention map described in Eq (4.4). This can be represented as a mixture model

$$p\left(z_t|z_{t-1},\rho\right) = \rho\, p_{\text{local}}\left(z_t|z_{t-1}\right) + (1-\rho)\, p_{\text{global}}\left(z_t|z_{t-1}\right). \tag{4.5}$$

The model parameter ρ describes the tendency to perform a step following either the local or global attention map. It can be fixed based on prior knowledge or inferred from the experimental data. If $\rho > 0.5$, then the probability for a local step is larger than for a global step for every saccade.

Next, we include in our model the assumption that ρ changes depending on the fixation location. We use the notation ρ_t to indicate that the fixation z_t was generated based on ρ_t. Importantly, this notation does not imply that ρ_t is necessarily a function of z_t.

We assume that the decision on the local or global attention maps depends on the ratio between the priority values of the current and previous fixated locations. The result is that the viewer is likelier to make a local step if the saliency value of the currently fixated location is higher than that of the previously fixated location. We include this in the model with the following expression for ρ_t

$$\rho_t = \sigma\left(f\left(s\right)\right) = \frac{1}{1 + \exp\left(-f\left(s\right)\right)} \tag{4.6}$$

with

$$f\left(s\right) = b\left(\frac{s_{t-1}}{s_{t-2}} - s^o\right) \tag{4.7}$$

with $s_{t-1} = s\left(z_{t-1}\right)$ and b and s^o being scalar variables.

Combining Equation (4.1) and Equation (4.5), the model likelihood is written as

$$p\left(Z|\Theta\right) = p\left(z_1\right)p\left(z_2\right)\prod_{t=3}^{t=T}\left(\rho_t\, p_{\text{local}}\left(z_t|z_{t-1}\right) + \left(1-\rho_t\right)p_{\text{global}}\left(z_t|z_{t-1}\right)\right) \tag{4.8}$$

with model variables $\Theta = \{\epsilon, \xi, b, s^o\}$. Here, we chose to sample the first and second fixation from the empirical static saliency map such that $p\left(z\right) = s\left(z\right)$.

Fig 4.3 presents a scan path generated by our model given a particular saliency map, alongside a scan path recorded experimentally from a viewer viewing the image corresponding to the saliency map.

4.2.2 Simplified Models

To test the different assumptions behind our full model described above, we construct three simpler models and compare their performances to the performance of the full model in the Results. We remove some of the assumptions on which the model is based to construct the simpler models. This results in the following competing models:

Local Choice Model: Equation (4.6) describes the assumption that the decision between two attention maps depends on the ratio between the priority value of the current fixation location and the priority value of the previous fixation location. A competing assumption would be that the decision depends only on the priority value of the current fixation location. In this case, we change $f\left(s\right)$

$$f\left(s\right) = b\left(s_{t-1} - s^o\right). \tag{4.9}$$

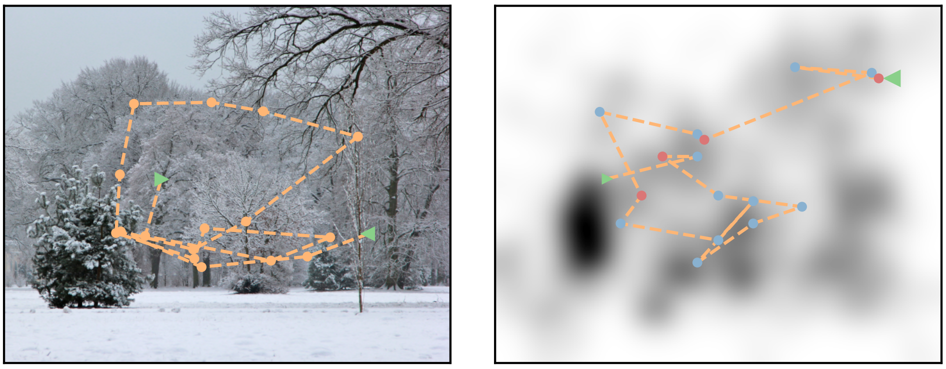


Fig. 4.3.: Left An image and a scan path recorded from a human observer. Right. The experimental static saliency map and a scan path are generated by the Local and Global Attention Model. The right green arrow represents the second randomly selected fixation location z_2. The green arrow pointing left represents the last fixation in the scan path. The blue dots are fixations that were generated from a global attention step, and the pink dots are fixations that were generated from a local attention step. The experimental data shows clearly the phenomenon of saccadic momentum, which the model does not capture. This is further discussed in the Results and Discussion.

Fixed Choice Model: We test the assumption that the decision between the modes does not depend on the saliency value of the previous fixation. In this simplification of the model, rather than having $\rho_t = f\left(z_{t-1}, z_{t-2}\right)$ we have a fixed probability to chose each policy with $\rho_t = \rho$.

Local Saliency Model: Last, we challenge the approach of two competing modes. In this variation of the model, each fixation is generated from a modulation of the empirical saliency map with a Gaussian around the current fixation location. This corresponds to the following fixation location likelihood

$$p\left(z_t|z_{t-1}\right) = \frac{s\left(z_t\right) n\left(z_t|z_{t-1}, \xi\right)}{\sum s\left(z'\right) n\left(z'|z_{t-1}, \xi\right)} \tag{4.10}$$

In this dissertation, we are concerned with Bayesian modeling in different fields. Hence, we also developed a Bayesian model and inference scheme for the use case of modeling scan paths. In the next section, we describe the Bayesian inference process of the full Model. As the three models described above are simplified versions of the full model, we do not describe their corresponding inference processes as they can be easily derived from the inference of the full model.

4.2.3 Bayesian Inference for the Exploration Exploitation Model

Our approach is based on known experimental results, and we derive the model parameters from observed data in a Bayesian framework. This approach allows us to include prior knowledge regarding the different model parameters based on known spatial features of scan paths. It also allows us to obtain distributions over the model parameters, rather than point estimates, and to compare different variations of the model via the respective test–data likelihoods.

Model Augmentation

We implement a Gibbs sampler for our model. It is not possible to apply it directly to the model as it was written in the previous section and we use two augmentation procedures as preparation for the inference. First, we use the standard approach and augment the Local and Global likelihood by

$$p(Z,\Gamma|\Theta) = p(z_1)p(z_2)\prod_{t=3}^{T} p_{\text{local}}\left(z_t|z_{t-1}\right)^{\gamma_t}\ p_{\text{global}}\left(z_t|z_{t-1}\right)^{1-\gamma_t} \tag{4.11}$$

with

$$\gamma_t \sim \text{Bern}\left(\rho_t\right) = \text{Bern}\left(\sigma\left(f\left(s\right)\right)\right) \tag{4.12}$$

and marginalizing over Γ results in Eq (4.8).

The augmentation defines a modified generative process for the model. At each time step, a variable γ_t is drawn from a Bernoulli distribution with bias ρ_t. If the result is 1, then the next saccade is generated following the local attention mode. If the result is 0, the saccade is generated from the global attention mode. This construction reflects our assumption regarding the cognitive process underlying scan path generation, where each saccade follows either local or global attention mode.

For a simple two–component mixture model with normal distribution, the augmentation described above would have been sufficient for deriving a Gibbs sampler [54]. As the model we constructed is more complex, we need to handle the sigmoid link–function in Eq (4.6) and the nontrivial form of the Global Attention distribution in Eq (4.3).

Next, we augment the model with another set of latent variables w_t following the Pólya-Gamma augmentation scheme described in Section 2.1.2. The resulting likelihood is

$$\begin{aligned} p\left(Z|W,\Gamma,\Theta\right) =& p\left(z_1\right)p\left(z_2\right)\prod_{t=3}^{T} p_{\text{local}}\left(z_t|z_{t-1}\right)^{\gamma_t} p_{\text{global}}\left(z_t|z_{t-1}\right)^{1-\gamma_t} \times \\ & \exp\left(-\frac{f\left(s\right)^2}{2}w_t + \left(\gamma_t - \frac{1}{2}\right)f\left(s\right)\right)p\left(w;1,0\right). \end{aligned} \tag{4.13}$$

After the augmentation with two sets of latent variables, we can define conjugate priors for the parameters:

$$\epsilon_{x/y} \sim \text{IG}\left(\epsilon_{x/y}; \alpha_{\epsilon_{x/y}}, \beta_{\epsilon_{x/y}}\right) \tag{4.14}$$

$$\xi_{x/y} \sim \text{IG}\left(\xi_{x/y}; \alpha_{\xi_{x/y}}, \beta_{\xi_{x/y}}\right) \tag{4.15}$$

$$b \sim \mathcal{N}\left(b; \mu_b, \sigma_b\right) \tag{4.16}$$

$$s^o \sim \mathcal{N}\left(s^o; \mu_{s^o}, \sigma_{s^o}\right) \tag{4.17}$$

where IG is the Inverse Gamma distribution, and $\mathcal{N}$ is the Gaussian distribution.

The prior distributions described above include hyperparameters. These parameters were chosen and not inferred from the data. The hyperparameters related to the prior distributions over $\epsilon_{x/y}$ and $\xi_{x/y}$ were chosen based on known characteristics of human saccades, such as that typical saccade amplitudes range from 0.5 and up to 40 visual degrees [5]. The hyperparameters related to b and s^o were chosen to be on the same scale of the average $\frac{s_{t-1}}{s_{t-2}}$ from the data. Further, all of the hyperparameters were chosen to induce wide prior distributions.

The posterior distribution over the model parameters and the latent parameters is given by

$$p\left(\Theta, \Gamma, W | Z\right) \propto p\left(Z | \Theta, \Gamma, W\right) p\left(\Gamma | \Theta\right) p\left(W | \Theta\right) p\left(\Theta\right) \tag{4.18}$$

with

$$p\left(\Theta\right) = p\left(\epsilon\right) p\left(\xi\right) p\left(b\right) p\left(s^o\right).$$

Marginal Conditional Posteriors

For the Gibbs sampler, we derive the explicit marginal conditional distributions for each model parameter given the data and the other model parameters. First, for the set of augmenting Pólya-Gamma parameters. Following the augmentation scheme, the conditional posterior of w_t is

$$p\left(\omega_t | Z, \Gamma, \Theta\right) = PG\left(w_t; 1, f\left(s\right)\right). \tag{4.19}$$

Next, the conditional posterior of γ_t is a Bernouli distribution Bern$(\gamma_t; \rho_t)$ with:

$$\rho_t = \frac{\sigma\left(f\left(s\right)\right)}{\sigma\left(f\left(s\right)\right) + \text{BF}\left(1 - \sigma\left(f\left(s\right)\right)\right)} \tag{4.20}$$

and BF defined as:

$$\text{BF} = \frac{p\left(z_t | \gamma_t = 0\right)}{p\left(z_t | \gamma_t = 1\right)} = \frac{p_{\text{global}}\left(z_t | z_{t-1}\right)}{p_{\text{local}}\left(z_t | z_{t-1}\right)} \tag{4.21}$$

Taking into account the definition of $f\left(s\right)$, we see that the parameters b and s^o appear in linear and quadratic forms in the arguments of the exponents in Equations (4.13). Given the Gaussian prior distributions we chose, the conditional distributions of these parameters are also Gaussian

$$p(b|Z, W, \Gamma, s^o, \epsilon, \xi) = n\left(b; m_b, s_b\right) \tag{4.22}$$

with mean and variance

$$m_b = \sum_{t=1}^{T} \left(\gamma_t - \frac{1}{2}\right)\left(\frac{s_{t-1}}{s_{t-2}} - s^o\right) + \frac{\mu_b}{\sigma_b} s_b \tag{4.23}$$

$$s_b = \frac{\sigma_b}{1 + \sigma_b \sum_{t=1}^{T} w_t \left(\frac{s_{t-1}}{s_{t-2}} - s^o\right)^2}. \tag{4.24}$$

Similarly

$$p(s^o|Z, W, \Gamma, b, \epsilon, \xi) = n\left(s^o; m_{s^o}, s_{s^o}\right) \tag{4.25}$$

with mean and variance

$$m_{s^o} = \left(\sum_{t=1}^{T} \left(b^2 w_t \frac{s_{t-1}}{s_{t-2}} - b\left(\gamma_t - \frac{1}{2}\right)\right) + \frac{\mu_{s^o}}{\sigma_{s^o}}\right) s_{s^o} \tag{4.26}$$

$$s_{s^o} = \frac{\sigma_{s^o}}{1 + \sigma_{s^o} b^2 \sum_{t=1}^{T} w_t}. \tag{4.27}$$

Due to the complex form of p_{global}, we do not have a closed form for the conditional distributions of ϵ and ξ from which we can sample. Thus, we resort to a technique known as MCMC within Gibbs [59, 117]. Instead of using the explicit marginal conditional, we draw a sample using HMC. The HMC algorithm requires an energy function and its derivative. In our case, we use the negative log of the model likelihood as the energy function. Rather than calculating the derivative of the log–likelihood analytically, we use automatic differentiation [63]. Specifically, we use the Python JAX package [20]. We further tuned the step size and number of steps parameters of the leapfrog algorithm to achieve an acceptance rate of 100%. Specifically, we used a step size of 0.03 for ϵ and 0.5 for ξ with eight leapfrog iterations for both.

The implementation of the models and inference can be found at
https://github.com/noashin/local_global_attention_model .

4.3 Inference Analysis for the Exploration Exploitation Model

We derived the model equations from the basic two modes approach in the previous section. We described the inference process of our model when applied to experimental data. In this Section, we present the results of the inference process. We analyzed the reliability of our procedures by fitting the model to artificial data generated from the model with known parameter values. In the next section, we fit the model to the experimental data and test the statistics of the data generated from the model against the experimental data, and we quantitatively compare different versions of the model.

The inference process includes using an MCMC approach to evaluate the posterior function over the model parameters. This approach is exact in the limit of an infinite number of samples. Still, as we can only use a finite number of samples, the result approximates the actual posterior. The distribution of the inferred parameters should concentrate on their real values.

When fitting the model to experimental data, it is impossible to know the real values of the model parameters as they do not relate directly to any measurable data features. Thus, to assess the inference's performance, we use data simulated by the model, in which case we know the exact values used to generate the data. If the inference process is correct, we expect the resulting posterior distribution to be concentrated around the ground truth values.

We generated data from our model with the parameter values that were inferred from the experimental data. In order to see whether the inference process will have reasonable results when fitting the experimental data, the size of the generated data set is comparable to the size of the experimental data for one subject.

Fig 4.4 presents the distribution over model parameters after the inference process over data generated by the model. Each of the ten colored curves represents a different inference process started at a different point. As expected, all the curves from different runs are similar in shape. The black dashed curves present the prior distribution over the parameters. As expected, the mode of the inferred parameter distribution is close to the real values used in the data generation, which are noted by the vertical solid line.

We tested the model on generated data with similar properties to the experimental data. Generated scan paths had lengths similar to the lengths of scan paths recorded experimentally. As a result, the generated data may not have sufficient information regarding the underlying model parameters, which explains the deviation of the distribution mode from the true parameter values.

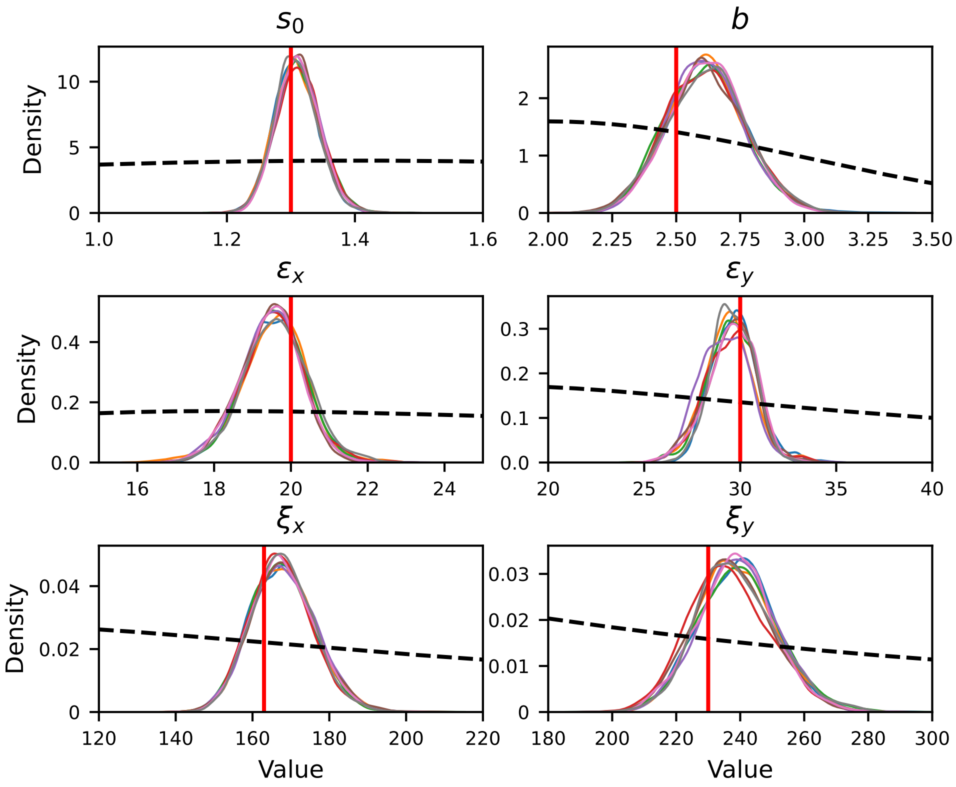


Fig. 4.4.: Model parameter recovery. To test the inference algorithm, we fit the model to simulated data with known parameter values. Each panel includes the inferred posterior distribution of each parameter after the inference process. The ten curves present 10 different inference processes starting from different values. The vertical lines are the values with which the data was generated. The black dashed curve is the prior distribution. The plotted densities are not normalized.

4.4 Model Performance on Experimental Data

Our model was derived from a set of hypotheses regarding the cognitive process of saccade generation. In order to test the validity of the model and of the corresponding hypotheses, we fit the model to the data, simulate new data using the model, and check whether the simulated data features correspond to the experimental data's features.

The data set here includes the scan paths of thirty–five human observers performing a memorization task over thirty natural images. The same data set was used before to evaluate other scan path models [46, 152]. The participants were presented with an image for 10 seconds and were instructed to explore the scene for a later memory test. The data acquisition was carried out in accordance with the Declaration of Helsinki, and informed consent was obtained for experimentation by all participants. Data from three subjects were excluded as the inference process did not converge. The data can be found at https://osf.io/me2sh/.

We fit a separate model for each subject while using the same prior hyperparameters for all models. We want to test whether the model captures subjects' tendencies that generalize over images. We use the k-fold cross-validation method with $k = 5$. All the reported quantitative results in this section are obtained from the test data averaged over the different folds.

4.4.1 Saliency Map Recovery

Since our model aims to produce a scan path for a given saliency map, the model needs to recover the empirical saliency map from experimental data. Fig 4.5 compares empirical saliency maps and the fixation locations density of data generated by the model. We used three different empirical saliency maps from the test set to simulate the full Local and Global Attention model and generated data from all the models fitted to the different subjects. The contour plot includes the density of the aggregated data, and the density is the empirical saliency map. Qualitatively, as expected, the fixation density of the data generated by the model matches the empirical saliency maps.

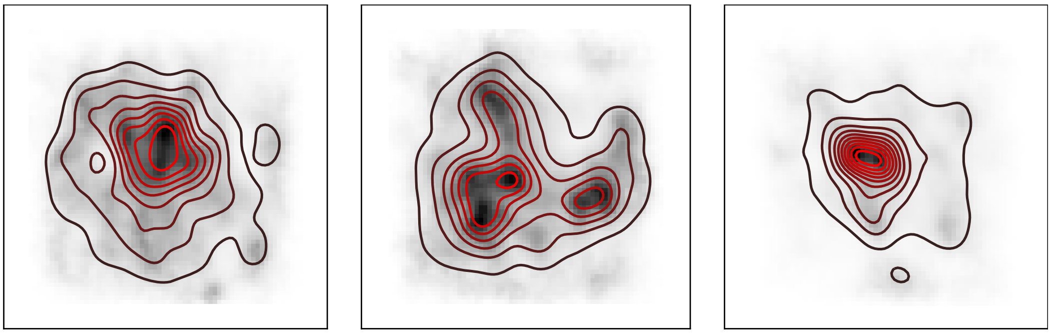


Fig. 4.5.: The empirical saliency is represented by the shading, and the contour lines represent the density of the data generated by the model. The generated data recovers the original empirical saliency map.

4.4.2 Modeling Saccade Amplitude

The Local and Global Attention Model was designed to capture the different saccade amplitudes generated by subjects while observing an image in a free viewing task. To estimate the model's performance, we compare the amplitudes of the empirical saccades with those of the saccades generated by the model. The comparison is made at a population level and for each subject separately.

Fig 4.6 compares the empirical saccade amplitude density with the saccade amplitude density of the scan paths that were generated by the full Local and Global Attention Model and the simplified versions presented previously. The density presented is over the entire population of subjects. The black curve presents the empirical data. The orange curve corresponds to data generated by the full Local and Global Attention Model, and the other curves correspond to the different simplified models.

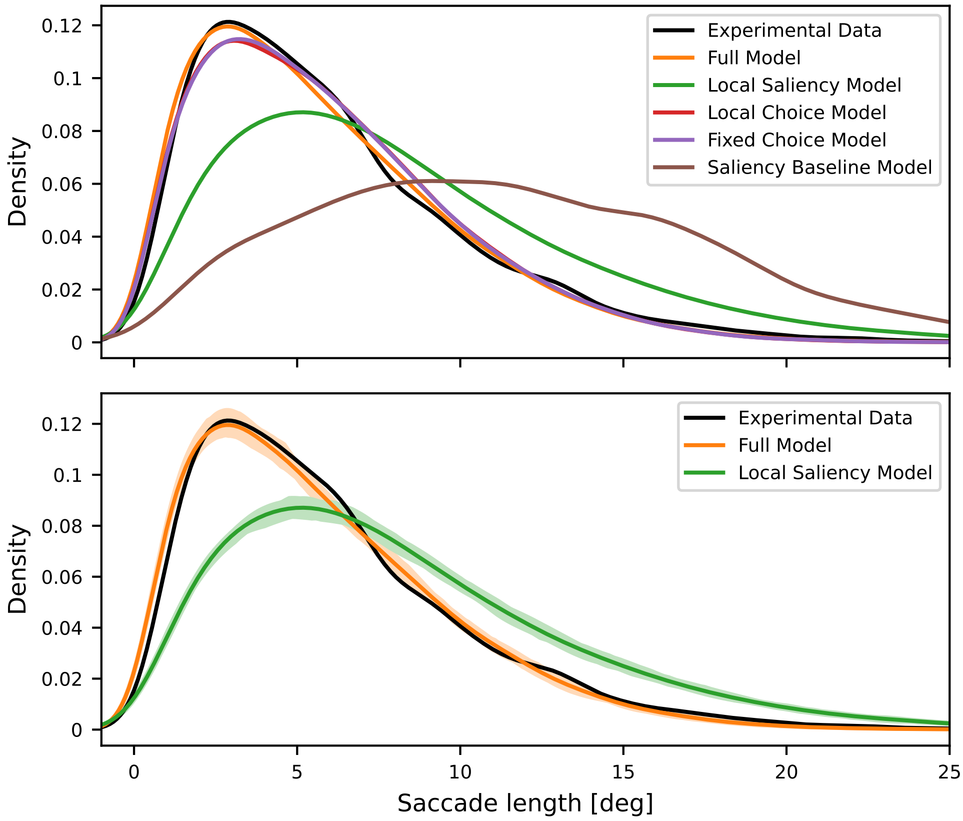


Fig. 4.6.: Saccade amplitude density, aggregated over the data from all participants, of the experimental data and data generated by the full model and the simplified competitor models. Top. Comparison of all models. Bottom. Comparison between the full model and the Local Saliency Model. The shading corresponds to confidence bounds regarding the estimate of the model parameters. The full model captures the different kinds of saccade lengths, whereas the simpler models fail to do so.

As a baseline, we include the Saliency Baseline Model, where the scan path is sampled from the saliency map. As this model does not have any constraints on the saccade amplitude other than the distance between high saliency areas in the image, the generated saccades have a much higher amplitude than the experimental data. Limiting the saccade amplitude, as in the Local Saliency model by assuming a local attentional focus, results in saccades with much more realistic amplitudes. Fig 4.6 shows that the full model performs better than the simplified models. The three simplified models tend to capture the mean saccade amplitude rather than the full variety of saccade amplitudes displayed in scan paths. This behavior is expected from the Local Saliency Model, which includes only one type of characteristic saccade amplitude, whereas the full Local and Global Attention Model has two characteristic saccade amplitudes that correspond either to the local or global attention mode.

The Bayesian inference process presented in the Methods Section results in a distribution over the possible values of the model parameters. This corresponds to uncertainty regarding the values of the model parameters. The shading around the generated data curves in Fig 4.6 corresponds to this uncertainty. We sampled 50 different values from the posterior distribution of each one of the model parameters and used this configuration to generate one data set. We split the experimental data into training and test sets three times and repeated the fitting of the models on each training set separately, resulting in 5–fold cross–validation. This process applies to all the results presented unless stated otherwise. Fig 4.6 presents the result of one such training and test split. The shading represents the 95% intervals around the mean density over the different data sets.

The confidence bounds are rather narrow, and the density distributions of the two models are highly separable. This is a good indication that the Bayesian parameter inference is reasonable—the saccade amplitude density does not change dramatically with the parameter configurations sampled from the posterior distributions. The confidence bounds for the Local and Fixed Choice Models behave in a similar way and are not included in the figure for clarity purposes.

The model presented in this work generates scan paths rather than independent saccades. Thus, we would expect to see some correlation between the generated saccades. Fig 4.7 presents the mean autocorrelations of the saccade amplitude along a scan path. The experimental data shows a clear anti-correlation between the amplitude of subsequent saccades at lag 1. Thus, a short saccade is likely to be followed by a long saccade and vice-versa. Although not as strong as in the experimental data, this effect is captured by the Local and Global Attention Model. This result is expected from our modeling assumptions since when generating fixation z_t the full model has information regarding the saliency of fixation z_{t-2}, whereas the competing models do not have access to this information. In addition to this lag–1 effect, it is important to note that our model also approximates the autocorrelation function for lags up to 20.

As described above, we fit a model for each subject individually. Thus, we can investigate how well the Local and Global Attention Model captures the difference between the subjects. In the left panel in Fig 4.8, we compare the mean saccade length of the empirical data and data generated from the full model for each subject. Each data point is one subject, and the diagonal curve is the identity line. The presented data is from one fold of the k–fold cross–validation.

Overall, the model captures the different mean saccade lengths of the different subjects. Not only does the model capture the different mean saccade amplitudes of the subjects, but it also captures the difference in the variability of saccade amplitudes of the subjects (see the right panel of Fig 4.8, where the standard deviations of the saccade amplitudes are plotted per participant.)

In Table 4.1, we report the coefficient of determination between the mean and standard deviation of the subjects' data and of the data generated by the full Local

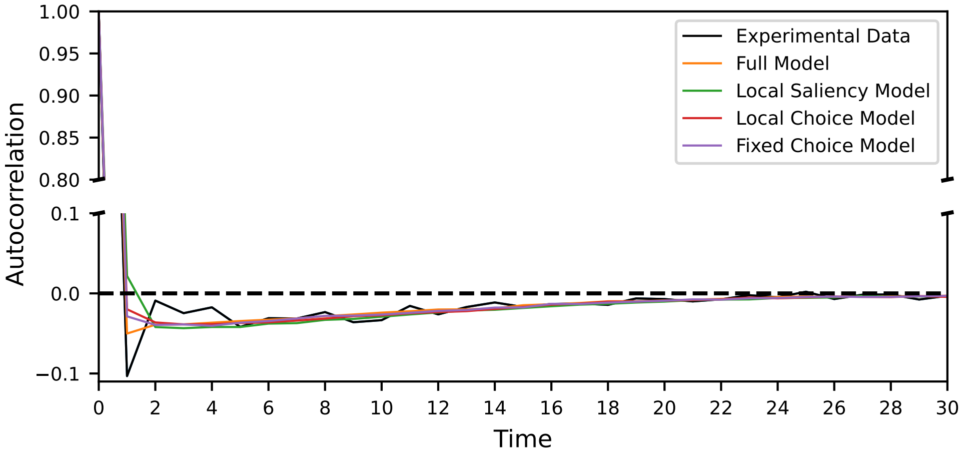


Fig. 4.7.: Saccade amplitude autocorrelation averaged over experimental data from all participants and overall simulations generated by the full model (and the various competitor models). The full Local and Global Attention Model approximates the autocorrelation in the amplitude of successive saccades, whereas the simpler models fail to reproduce the lag-1 anti-correlation.

	Local and Global Attention	Local Choice	Fixed Choice	Local Saliency
R^2 Saccade amplitude mean	0.93	0.93	0.93	0.47
R^2 Saccade amplitude mean	0.85	0.847	0.85	0.3

Tab. 4.1.: Comparison of the coefficient of determination between the mean (or std) of the subjects' saccade amplitudes and the saccade amplitudes of the data generated by the different models. Other than the local saliency model, all models capture both the mean and the standard deviation of the saccade amplitudes of the different subjects.

and Global Attention Model and the competing simplified model. The coefficient of determination was averaged across the different train-test splits in the cross–validation. Other than the Local Saliency model, all models perform similarly well and capture the different subjects' mean and standard deviation of the saccade amplitudes. This result indicates that the assumption of two length scales generated by local and global attention states represents a major improvement in the model fit.

4.4.3 Modeling Saccade Direction

Saccades represent the eye movement from one fixated location to another. Hence, they are characterized not only by amplitude but also by direction. After analyzing the model performance with regard to the saccade amplitude, we turn to analyze the model performance with respect to the saccade direction.

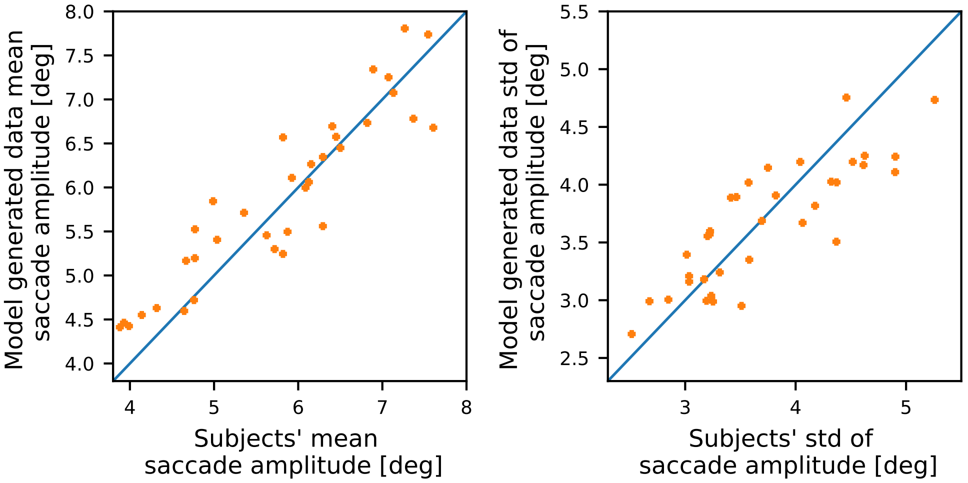


Fig. 4.8.: Left. Participants' mean saccade amplitudes compared with the mean saccade lengths of data generated by the Local and Global Attention model. Right. The standard deviation of the subjects' saccade amplitude compared to the standard deviation of the data generated by the Local and Global Attention model. Overall the model captures both the mean and the standard deviation of the saccade amplitude of the different subjects.

There are two important aspects regarding saccade direction, i.e., absolute saccade direction and the direction relative to the previous saccade. In the left panel in Fig 4.9, we compare the saccade direction density, over the entire population of subjects, of the empirical data and of data generated by the fitted full model and its variations. The empirical data demonstrate a clear preference for horizontal saccades and a weaker tendency towards vertical upward saccades. The different variations of the model generate similar distributions of saccade directions. The data generated by the models correspond to a tendency to perform horizontal saccades, but this tendency is not as strong as in the empirical data. The generated data also shows a tendency towards vertical saccades.

Interestingly, the subjects seem to perform upwards saccades much more than downward saccades. In contrast, the models do not differentiate between "up" and "down," and the two vertical directions are equally likely. This could imply a cognitive bias of human observers resulting from their prior knowledge of the world, which our model does not capture.

The right panel in Fig 4.9 presents the frequency of the values of the change in the saccade direction. The experimental data is characterized by a large peak around 0, which indicates the persistence of the current saccade direction [148, 21, 182], also known as saccadic momentum. Additionally, a weaker peak around 180 and -180 degrees indicates a tendency to return to the previously fixated location. All models discussed here fail to reproduce this effect. The peak in the saccade direction change is only around 180 and -180 degrees which are due to the hard constraints given by the image boundaries.

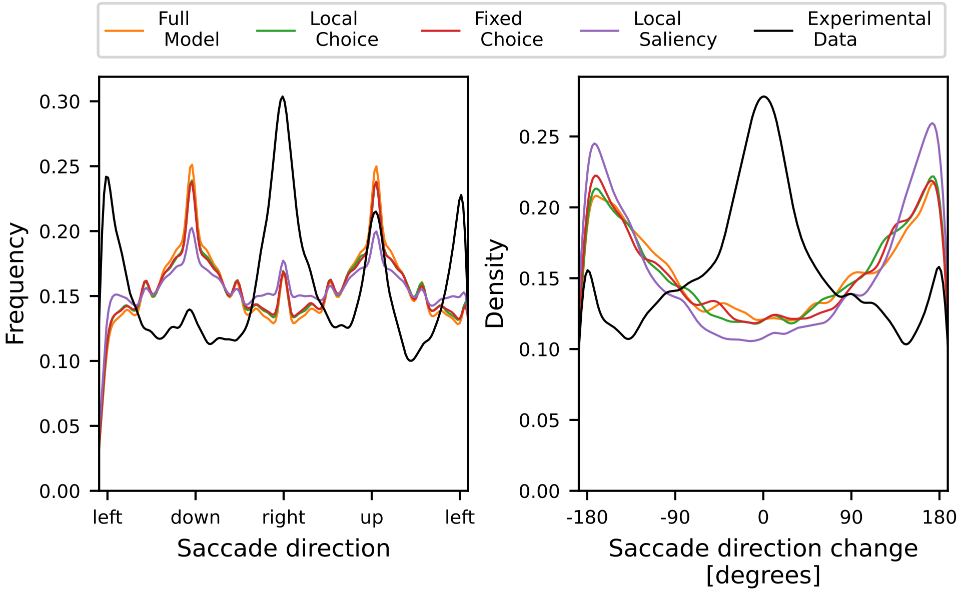


Fig. 4.9.: Left. Absolute saccade direction. The empirical data demonstrate a strong tendency to saccades towards the left and right directions and a weaker tendency to perform saccades directed upwards. The generated data captures the tendency to perform horizontal and vertical saccades. Right. Change in saccade direction. The generated data demonstrates the tendency to persist in the same direction. The models fail to capture this persistence.

4.4.4 Model Comparison

Last, we would like to compare the performance of the full Local and Global Attention Model to the simplified variants of the model presented in Section 4.2.2. As measurements of the model performance, we use the Receiver Operating Characteristic (ROC), the respective Area Under the Curve (AUC), the Normalized Scan path Saliency (NSS), and the Information Gain (IG). The first two methods are widely used in the field of attention modeling, and have been successfully adapted to scan path modeling [141, 181, 18, 147, 93, 94]. The IG for saliency and scan path modeling analysis was first suggested by Kümmerer et al. in [95] and is becoming more common ever since [152, 155].

The AUC measure is a very common tool to analyze the performance of a probabilistic classifier by looking at the trade-off between True Positive and False Positive Rates (TPR and FPR) when using different thresholds for classification. In the context of attention modeling, the fixation locations are used as samples with positive labels. As samples with negative labels, we used image coordinates that were sampled uniformly in the image space. This method is denoted as AUC-Borji [27]. In our analysis, the likelihood of the model at each time step is used as the probabilistic classifier. Fig 4.10 presents the ROC curves for the different models.

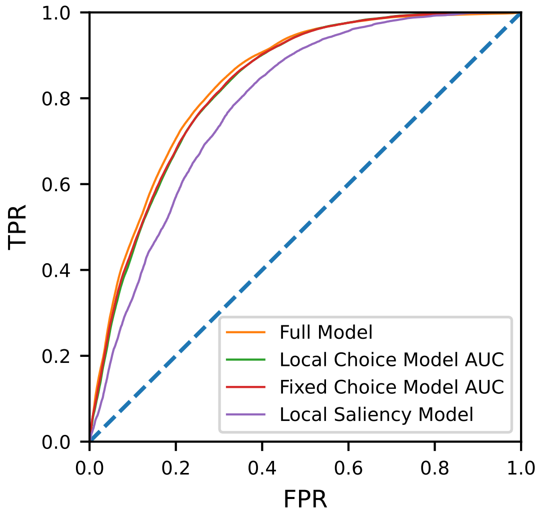


Fig. 4.10.: The full model performs slightly better than the Local and Fixed Choice models. The Local Saliency Model performs significantly worse than the other model variants.

The NSS is the average normalized saliency (with mean zero and standard deviation of one over the image) along the scan path of fixated locations. In this work, we used the likelihood value of each fixation in the scan path and normalized the likelihood over the entire image, as one would do with a saliency map.

Here, we use the same definition of IG as in [95] - as the average difference of the log-likelihood between a model and some baseline model. As a baseline, we take a uniform distribution over the image, where the log-likelihood for each fixation is constant and equal to $\log_2(\text{number of pixels})$. The unit of this measure is bit per fixation (bit/fix) and is interpreted as the amount of information gained compared to the baseline model per fixation.

As done in the previous analysis, the scores are reported on the test data set, which was not used in fitting data, and averaged over the different samples from the posterior distributions and the folds of the cross–validation process. Due to the data split into training and test sets, the discussed measures are sensitive to the model complexity. If a model is too complex (usually manifested by having a lot of parameters), the model will achieve high AUC, NSS, or IG, over the training set but may suffer from overfitting and perform poorly on the test set.

The results in Table 4.2 show that the full model performs better than the other variants and achieves higher scores. Although the scores of the full model are the highest, they are only slightly better than the ones of the Local and Fixed Choice models. The Local Saliency model performs poorly, emphasizing the importance of the two characteristic length scales present in all of the model variants other than the Local Saliency Model.

	Local and Global Attention	Local Choice	Fixed Choice	Local Saliency
AUC	0.843	0.835	0.838	0.804
NSS	1.48	1.44	1.41	1.05
IG [bit/fix]	2.1	1.99	1.97	1.92

Tab. 4.2.: The full model performs better than the Local and Fixed Choice models. The Local Saliency Model performs significantly worse than the other model variants.

4.5 Discussion

We proposed and analyzed a mathematical model of fixation selection, motivated by the Local and Global Attention modes suggested previously as a mechanism driving eye movements in natural scene-viewing tasks Frost and Pöppel [49], Unema, Pannasch, Joos, and Velichkovsky [176], Helmert, Joos, Pannasch, and Velichkovsky [72], and Tatler and Vincent [170]. We constructed a generative scan path model based on a small set of assumptions. Using Bayesian inference, we fit the model to experimental data. By doing so, we continue the line of work of using generative likelihood–based models for scan path generation [152, 154]. Importantly, we use recent developments in Bayesian statistics to construct more efficient parameter inference algorithms.

A different approach uses deep neural networks for scan path modeling [157, 94]. One of the downsides of this approach is its reliance on large amounts of data, which precludes the study of interindividual differences. Thus, by using a hypothesis–based model, which requires only a relatively small number of parameters, we can fit individual models for each experimental subject and capture inter–subject variability.

We demonstrate how our model captures the saccade amplitude at the population and individual levels. Whereas two of the competing models perform equally well in terms of the coefficient of determination fitted to the mean and standard deviation of the individual subjects' saccade lengths, the advantage of the full model is demonstrated when looking at the autocorrelation of the saccade length. This analysis considers the individual saccades and the dynamics of the entire scan path. These results emphasize the importance of the information about the saliency of the previously fixated location when deciding on the following fixation location, as described in Equation (4.7). Our model generates the typical behavior of a short saccade after a long one, or vice versa, which results in the observed anti-correlation of the length of subsequent saccades.

To further quantify the model performance, we calculated the AUC, NSS, and IG scores of the different variants of the model. The complete model and the Local and Fixed Choice variants performed similarly well, and the full model achieved slightly higher scores than the other two. This result indicates that these quantitative scores may not be enough to evaluate how well a scan path model fits the data. Rather than relying only on likelihood–based measures such as AUC, NSS, and IG, a more careful investigation of the whole body of results suggests that the saccade behavior is characterized explicitly by its autocorrelation function of the saccade amplitudes.

The Local and Global Attention Model captures the experimental saccade amplitudes at the population and subject levels. Another spatial aspect of saccades is saccade direction. Our model captures only the tendency to perform horizontal saccades but not the tendency to perform vertical saccades. This is expected from the construction of the model.

As the full model has information regarding the saliency of the previous fixation location but not of the location itself, in its current form, the model does not capture the change in the saccade direction (i.e., the saccade direction relative to the previous saccade). The relative saccade direction is essential for modeling known phenomena such as visual persistence or saccadic momentum [148, 21, 182, 112]. The model's inability to capture the relative saccade direction stems from the choice of Gaussian functions in the local and global attentional states. Our model could be extended to account for these tendencies by a Gaussian mixture. Each Gaussian component would be designed to capture different directional tendencies rather than capturing only one tendency, as in the current version of the model. For example, one Gaussian can be aligned in the direction of the previous saccade, accounting for visual persistence.

Other limitations of the model stem from the choice of a second–order Markov process. Due to this choice, the model is almost memory-less and cannot capture known phenomena in scene viewing, which span multiple saccades. Incorporating more extended history is not straightforward in our model. A heuristic approach could be including dynamics in the saliency map.

Finally, our mathematical model does not account for fixation durations in scene viewing, which play an essential role in eye-movement control [73, 134, 99, 168]. So far, most of the modeling attempts of scene viewing addressed either the spatial or the temporal aspects of scene viewing. Indeed, some models use temporal dynamics but do not attempt to learn them from the data and use a heuristic–based approach. We propose new approaches to spatiotemporal modeling of scan paths later in this dissertation in Section 6.2.2.

5 Evaluating Sampling Algorithms for Bayesian Models in Human Scene Viewing

5.1 Introduction

In the previous chapters, we presented models with a tailored inference algorithm. Our innovation and contribution were not only the models themselves but also the algorithm used to infer the parameters of the models from real data. This approach poses limitations on the models. We cannot develop a Gibbs sampler or a variational inference scheme for most models. In such cases, we can resort to using inference methods that can be applied to many models. Specifically to methods that only require a model with a computable likelihood function and do not impose any restrictions on the shape of the likelihood.

This raises a new question - which method should we use? In this chapter, we explore this question. We chose the word "explore" very carefully, as we do not believe this question can have a definite answer. That said, we are interested in shedding light on using different sampling algorithms and their performances.

The question presented in the previous paragraph is very general, and for the sake of remaining in the scope of this dissertation, we must narrow it. First, as our work's main interest in this dissertation, we will examine Bayesian inference methods. Hence, methods to estimate the posterior of the model. Second, we will limit the field of applications. As a use case, we stay in the same field as the previous chapter - Bayesian models of natural scene viewing.

We chose this field as it seems to lack "best practices" when inferring model parameters. Earlier statistical models put very little emphasis, and often non, on describing rigorously the model parameters and how they were estimated. For example, the seminal work of Itti and Koch [80] includes the values of the model parameters but does not describe how these values were found. Similarly, the CLE model [16], which models saccades with Levy flights [102], reports the values with which the experiments were conducted but not how these values were chosen.

In other models, some parameters can be calculated explicitly from the experimental data. For example, in Tavakoli et al. [171], the parameter P_0 describes the "fixation probability." It is estimated by dividing the number of data points containing fixations by the total number of data points. This value is generated from a specific data–set, which can be thought of as training data, whereas other data–sets are used to evaluate the model performance.

In other models, the parameters are estimated by optimizing the model performance concerning some metric. For example, Zanca et al. present a model which relies on mechanical physics and includes four parameters. To estimate these parameters, they maximize in [186] the AUC, which we described in 4.4.4. Another variation of this model is described in [185], where the authors maximize the NSS, which we also described in 4.4.4, to estimate the model parameters.

The above models do not define an explicit likelihood function over the data. One of the first models to use an explicit likelihood function is the model of Liu et al. described in [108]. This model includes modules describing low–level feature saliency, semantic content, and spatial position. The second module is modeled as a hidden Markov model (HMM), and its parameters are estimated using a maximum likelihood scheme.

Another model that defined an explicit likelihood function is the one from Penttinen et al. [140]. In this work, the authors define an inhomogeneous self-interacting random walk that accounts for the different saliency levels in the image, saccade lengths, and history–dependent effects. An explicit likelihood function is derived, and the model parameters are estimated by optimizing the likelihood.

The models described above are defined with a likelihood function but do not include a Bayesian approach. No prior probability is associated with the model parameters, and they are estimated based on the likelihood alone. This approach may result in undesirable artifacts that can be avoided using a Bayesian approach.

An early example of using Bayesian inference for inference of saccadic eye movement is the work of Welke et al. from 2009. In this work, the authors describe a Bayesian network to model saccadic eye movement and use particle filter [129] to infer the hidden states of the system. A further example of the usage of Bayesian inference for natural scene viewing models is the work of Courtrot et al. [31], where the number of states in their hidden Markov model is determined with variational inference [118]. In this case, the authors chose a Bayesian approach, as maximum likelihood estimation tends to favor complicated models which include many hidden states. Here, the prior is used as a regularizer, balancing the tendency of the likelihood of overly complicated models.

Another model that uses a Bayesian approach is the Scenewalk model. The earlier version of the model used an MH sampler to estimate the model's parameters [153]. In contrast, the updated version of the model [155] uses the more sophisticated DREAM [96] algorithm. In this chapter, we test the different sampling methods on the exploration–exploitation model presented in the previous chapter and on the earlier version of the Scenewalk model described in [153].

We will next introduce in general detail the Scenewalk model. Then, we will present the different inference algorithms whose performances we want to analyze, followed by the two metrics we will use to compare the inference algorithms. After that, we will demonstrate the performance of the inference algorithms on the two models and conclude with a discussion of their performances.

5.2 Scenewalk Model

The second model for which we report the results of the different samplers is the Scenewalk model. It was first introduced in [46]. A Bayesian inference framework was integrated into it in [153]. The Scenewalk model was further extended in [155], but here we use the simpler version of the model that was included in [153].

Similarly to the saccade generation mechanism presented in 4, the Scenewalk model takes as an input a static saliency map S and returns a scan path - a series of fixations location and fixations duration. For a given time t, the model estimates a dynamic saliency map from which the next fixation location is selected. The dynamic saliency map is created by combining two separate processing streams that evolve in parallel. These processing streams represent the inhibitory and excitatory mechanisms identified as the driving forces underlying saccadic eye movement. The excitatory process is associated with attention [80, 79] and is balanced by the competing process, which models the concept of inhibition of return [86, 85].

As mentioned, the Scenewalk model generates fixation locations and fixation durations, but it treats the spatial and temporal aspects separately. First, a series of fixation durations $\Delta t_1, \Delta t_2, ..., \Delta t_n$ are drawn i.i.d. from a Gamma distribution. Then, the fixation locations are generated sequentially for each fixation duration. The fixation locations are drawn from a dynamic saliency map that is the product of the inhibitory and excitatory processing streams. We next describe the calculation of the dynamic saliency map.

The excitatory process A is represented by an *attention map* and the inhibitory process F by an *inhibition map*. These maps are defined for each location z in the image space and are calculated given a fixation duration Δt_f and fixation location z_f

$$A_{\Delta t_f, z_f}\left(z\right) = \frac{G_A\left(z\right)S}{\sum G_A\left(z\right)S} + e^{-\omega_A \Delta t_f}\left(A_0\left(z\right) - \frac{G_A\left(z\right)S}{\sum G_A\left(z\right)S}\right) \tag{5.1}$$

$$F_{\Delta t_f, z_f}\left(z\right) = \frac{G_F\left(z\right)}{\sum G_F\left(z\right)} + e^{-\omega_F \Delta t_f}\left(F_0\left(z\right) - \frac{G_F\left(z\right)}{\sum G_F\left(z\right)}\right). \tag{5.2}$$

G_A and G_F are Gaussian functions centered around the current fixation location $z_f = (x_f, y_f)$, with a diagonal covariance matrix $\sigma_{A/F}I$. A_0 and F_0 are the previously calculated attention and inhibition maps. The summing in the denominator is done over the discretized image space. Last, the Gaussian attention map G_A is multiplied by the static saliency map of the image S.

In the next steps of the Scenewalk model, Equations (5.1) and (5.2) are combined and normalized to produce a valid distribution function over the image space from which the next fixation location is selected. First, the inhibitory map is subtracted from the excitatory map, and the result is passed through a ReLu function where the negative values are zeroed

$$u(z) = \frac{A(z)^{\gamma}}{\sum A(z)^{\gamma}} - C_F \frac{F(z)^{\gamma}}{\sum F(z)^{\gamma}} \tag{5.3}$$

$$u^*(z) = ReLu(u(z)) \tag{5.4}$$

and we omitted the notation A_{Δ_f, z_f}. Finally, the priority map u^* is normalized and noise is added to produce the final probability map over the image space

$$p(z) = (1 - \zeta) \frac{u^*(z)}{\sum_z u^*(z)} + \zeta \frac{1}{\sum_z 1}. \tag{5.5}$$

The model parameters to be infered are $\{\sigma_A, \sigma_F, \omega_A, \omega_F, \gamma, C_F, \zeta\}$. In [153], an MH algorithm was used to infer those parameters. When developing the extended version of the Scenewalk model [155], the authors chose to use the DREAM [96] algorithm to infer the model parameters. We next present the inference algorithms that we compare in this work.

5.3 Inference algorithms

In this chapter, we chose to focus on sampling algorithms. As discussed in Chapter 2, sampling is fundamental for Bayesian inference, and sampling algorithms have been developed and used successfully for many years. Furthermore, as we showed in Section 5.1, samplers have been used successfully for parameter inference in scan path models.

In developing a novel model for scan path generations, see Chapter 4, we developed a Gibbs sampler to infer the model parameters. This sampler was tailored to our model and cannot be used "out of the box" for any likelihood–based model.

This chapter focuses only on samplers that can be applied to any model with a computational likelihood function. One is the HMC sampler, whose details were presented in Chapter 2. This section presents the other samplers that have not yet been presented. The first is an auto–tuned variation of HMC.

5.3.1 No–U–Turns Sampler

The HMC sampler used in different parts of this dissertation relies only on a computable likelihood function, but it contains several tuning parameters. A wrong choice of these parameters may lead to very slow convergence, and sometimes the sampler may not converge at all. The no–U–turns (NUTS) sampler [75] is an extension of HMC, which automatically tunes the sampler parameters.

The HMC sampler requires setting the covariance matrix of the prior over the augmenting parameter ϕ and choosing the number of steps L and steps size ϵ for the leapfrog integration algorithm. The NUTS algorithm heuristically sets the mass matrix of the momentum parameter and automatically tunes L and ϵ.

Mass Matrix

NUTS algorithm uses the common Euclidean–Gaussian kinetic energy and sets the mass matrix to be the identity matrix, which results in the following Hamiltonian

$$H\left(\theta,\phi\right) = -\log\left(p\left(\theta|y\right)\right) + \frac{1}{2}\phi^{\top}\phi.$$

Number of Leapfrog Steps L

The number of leapfrog steps taken by the integrator is an important parameter of HMC. If L is too small, the proposed new position will be close to the initial position, and the sampler will explore the target distribution slowly. If L is too large, the path may begin to retrace itself, performing a U–turn, and the resulting proposed position will be very close to the initial position.

NUTS aims to dynamically set L at each step of the algorithm. In a nutshell, the main differences between NUTS and HMC are:

1. The leapfrog integrator simulates the path in the Hamiltonian space both forward and backward in time until a U–turn is detected.

2. Rather than having an accept–reject step on the last state of the leapfrog integrator, NUTS samples a point from the simulated states.

The first point includes two important details of NUTS: the extension of the path in both directions and the termination condition. The dynamic path extension is performed in the following way. At step j of the leapfrog integrator, the direction (forward or backward) is chosen randomly. Then, 2^{j-1} leapfrog steps are performed in this direction. The concurrent extension forward and backward is essential for the process reversibility.

To determine whether the simulated path performs a U–turn, NUTS uses as a criterion the derivative in time of the size of the change in position

$$\frac{d}{dt}\frac{\left(\tilde{\theta}-\theta\right)^{\top}\left(\tilde{\theta}-\theta\right)}{2}=\left(\tilde{\theta}-\theta\right)^{\top}\frac{d}{dt}\left(\tilde{\theta}-\theta\right)=\left(\tilde{\theta}-\theta\right)^{\top}\tilde{\phi} \tag{5.6}$$

where the tilde sign indicates the current position and momentum. When the quantity above becomes negative indicates a U–turn. The doubling process of the leapfrog steps is then stopped when a sub–trajectory fulfills the U–turn condition.

As NUTS does not include an accept–reject step, the selection process from the simulated states needs to be constructed such that it preserves detailed balance. We next describe this construction in general detail and refer the reader to [75] for an in–depth description of the algorithm.

To preserve the detailed balance, NUTS uses slice sampling [132]. At each step of NUTS, after sampling the momentum variable, the model is further augmented with a *slice variable* u which is sampled from

$$p\left(u|\theta,\phi\right)=\text{Uniform}\left(u;\left[0,\exp\left(-H\left(\theta,\phi\right)\right)\right]\right).$$

Then, after the doubling process of the leapfrog trajectory terminates, we have the set $\mathcal{B}$ of all visited states. A sub–set of states $\mathcal{C}\in\mathcal{B}$ is deterministically selected such that $(\theta',\phi')\in\mathcal{C}\Rightarrow\exp\left(-H\left(\theta',\phi'\right)\right)>u$.

The slice sampling is equivalent to the accept–reject step in HMC. In HMC the proposal (θ',ϕ') is accepted if

$$\begin{aligned}v&\sim\text{Uniform}\left(0,1\right)\\ \log v&<H\left(\theta,\phi\right)-H\left(\theta',\phi'\right).\end{aligned}$$

Taking the exponent and rearranging, we get the condition

$$v\exp\left(-H\left(\theta,\phi\right)\right)<\exp\left(-H\left(\theta',\phi'\right)\right)$$

and the left hand side is equivalent to u. This means that (θ',ϕ') will be accepted when

$$u<\exp\left(-H\left(\theta',\phi'\right)\right).$$

In NUTS, using slice sampling, rather than checking if the final state passes the acceptance threshold in HMC, all the states along the trajectory are checked, and the new state is chosen randomly from the ones that pass the threshold.

Step Size ϵ

To determine ϵ, NUTS implements a version of stochastic optimization via vanishing adaptation [149], which is an adaptation of the primal–dual algorithm [133]. The optimization process is done during the burn–in period, and the resulting ϵ is fixed for the rest of the sampling process.

The optimization is done with respect to the desired acceptance rate δ, and we define $H_t = \delta - \alpha_t$, where α_t is the acceptance probability at iteration t. The goal of the optimization is to set the sampler parameters, in our case, the step size of the leapfrog, such that $\mathbb{E}_t \left(H_t|x\right)$ converges to 0, where x is the tunable parameter. The update step of the dual–averaging algorithm is

$$x_{t+1} \leftarrow \mu - \frac{\sqrt{t}}{\gamma} \frac{1}{t+t_0} \sum_{i=1}^{t} H_i \tag{5.7}$$

$$\bar{x}_{t+1} \leftarrow t^{-\kappa} x_{t+1} + \left(1 - t^{-\kappa}\right) x_{t+1}. \tag{5.8}$$

μ, γ, t_0 and κ are freely chosen parameter, and the authors of [75] recommend the following default values $\gamma = 0.05$, $t_0 = 10$ and $\kappa = 0.75$.

Unlike HMC, NUTS iteration does not include an accept/reject step, and we need to define an alternative to the Metropolis acceptance probability α_t. NUTS uses the following statistics

$$H_t^{NUTS} = \frac{1}{|\mathcal{B}_t|} \sum_{\theta', \phi' \in \mathcal{B}_t} min \left\{1, \frac{p\left(\theta', \phi'\right)}{p\left(\theta^{t-1}, \phi^{t,0}\right)}\right\},$$

where $\mathcal{B}_t$ is the set of all the states visited in the simulated trajectory. H_t^{NUTS} is comparable to the acceptance rate in HMC, averaged over all the states encountered during the leapfrog integration, rather than only evaluated in the last state in the simulation. Following this definition, the desired quantity to be optimized is set to be $H_t \equiv \delta - H_T^{NUTS}$, and $x \equiv \log \epsilon$.

5.3.2 Affine Invariant Interacting Langevin Dynamics Sampler

Another sampling method we put to the test is the novel ALDI algorithm [51]. Similarly to HMC, this sampling method only requires the model to have a differentiable likelihood function. The sampling is based on first–order Langevign dynamics and it is invariant under an affine change of coordinates (affine invariance [61]). We describe the sampler only in the general terms that are relevant to the application of the scene–viewing models considered here. Further details can be found in [51].

The ALDI sampler consists of a stochastic process of M interacting particles. The stochastic process is characterized by a gradient-based evolution equation of the following form

$$d\theta_t^i = -C\left(\Theta_t\right)\nabla_{\theta^i}\Phi\left(\theta_t^i\right)dt + \frac{D+1}{M}\left(\theta_t^i - m\left(\Theta_t\right)\right)dt + \sqrt{2}C^{\frac{1}{2}}\left(\Theta_t\right)dW_t^i \tag{5.9}$$

with

- θ being the vector of the model parameters
- θ_t^i referring to the value of particle i at time t
- Θ_t being a matrix containing all particles at time t
- $C\left(\Theta_t\right)$ the particles' empirical covariance matrix
- $m\left(\Theta_t\right)$ the particles' empirical mean
- $\Phi\left(\theta_t^i\right)$ the potential function which is associated with the model
- D the dimension of the particles which corresponds to the number of model parameters we infer
- M number of particles
- W_t^i denoting a standard M–dimensional Brownian motion.

We solve the SDE in Equation (5.9) with the following update rule, according to the Euler–Maruyama method [87]

$$\theta_{n+1}^i = \theta_n + \left(-C\left(\Theta_n\right)\nabla_{\theta^i}\Phi\left(\theta_n^i\right) + \frac{D+1}{M}\left(\theta_n^i - m\left(\Theta_n\right)\right)\right)\Delta t + \sqrt{2}C^{\frac{1}{2}}\left(\Theta_t\right)\Delta W_n^i$$
$$\Delta W^i \sim \mathcal{N}\left(0, \Delta t\right).$$

In the experiments in this section, we take $M = D + 1$ as it is demonstrated to be sufficient in [51].

5.3.3 Multi–Trial Differential Evolution Adaptive Metropolis Sampler

The fourth algorithm we consider in this chapter is a multi–chain MCMC–type sampler that requires only a computable target distribution, the posterior distribution of the model parameters in our case. This algorithm is the multi–trial differential evolution adaptive Metropolis (MT-DREAM) algorithm [96]. We will first present the different ideas the sampler builds on and then describe it in detail.

The first building point behind the MT-DREAM algorithm is the *Adaptive Metropolis*. One of the main challenges in MCMC algorithms is the choice of an appropriate proposal. A well–chosen proposal will result in fast convergence to the target distribution. In contrast, a badly chosen one may require a very long simulation and have a very low acceptance rate. Overcoming this is could be done by tuning the proposal based on the history of the samples [47, 52, 24]. Probably the most well-known implementation of this idea is the Adaptive Metropolis (AM) algorithm from Haario et al. [64]. In this AM algorithm, the proposal is still a Gaussian distribution with mean in the current point θ_{t-1}, like in MH, but the covariance of the proposal is taken to be the covariance of the past accepted samples.

Another approach used in MT–DREAM is the *multiple trial Metropolis* (MT–MC) [109]. Like the AM sampler, it should result in faster convergence and a higher acceptance rate than the regular MH sampler. In multiple trial Metropolis, at each step k candidates $\{z_1, ..., z_k\}$ are sampled from the proposal distribution $q\left(\theta_{t-1}\right)$. Then, one of the candidates is chosen with probability $p\left(z_i\right)$, with $p\left(\cdot\right)$ being the target distribution. The selected candidate is then accepted with a probability

$$\alpha = \min\left\{\frac{p\left(z_1\right) + ... + p\left(z_k\right)}{p\left(\theta_1^*\right) + ... + p\left(\theta_k^*\right)}, 1\right\} \tag{5.10}$$

where $\theta_1^*, ..., \theta_{k-1}^*$ are reference points drawn from the proposal distribution $q\left(z_i, \cdot\right)$ and $\theta_{j-1}^* = \theta_{t-1}$.

The last tool MT– DREAM uses is *differential evolution* (DE). This algorithm was first introduced in [163] and is part of the family of genetic or evolutionary algorithms used in optimization problems. Ter Braak first used it in the context of MCMC in [172] in an algorithm he named *differential evolution Markov chain* (DE–MC). In this algorithm, M chains are evolved in parallel and are viewed as the *population* in evolutionary algorithms terms. At each time step, a proposal is generated for chain i from the difference between the current state of two randomly selected chains

$$z^i = \theta_{t-1}^i + \gamma\left(\theta_{t-1}^{r_1} - \theta_{t-1}^{r_2}\right) + e \qquad r_1 \neq r_2 \neq i$$

where $r_1, r_2 \in \{1, ..., M\}$ randomly selected indices of two other chains. The proposal is accepted with probability $\frac{p\left(z^i\right)}{p\left(\theta_{t-1}^i\right)}$.

DE–MC applies the *crossover* mechanism [163] to improve the sampling process further, and each iteration updates only some of the dimensions in θ. Before deciding whether to accept or reject the proposal z, it is adjusted by crossover. In DE–MC, the crossover is implemented with a binomial scheme. Each element in z^i is replaced with the corresponding element in θ_{t-1}^i, with probability $1 - CR$, where CR is the *crossover probability*.

The MT–DREAM algorithm combines and extends the three approaches presented above. Similarly to MT–MC, it generates at each iteration, and for each chain, k candidates, and similarly to DE–MC, it uses the differences between previously accepted samples as a proposal. At each time step and for each chain, the k proposal is generated in the following way

$$z^{l,i} = \theta_{t-1}^{i} + (1+e)\,\gamma\left(\delta, d'\right)\left(\sum_{j=1}^{\delta}\Theta^{r_1(j)} - \sum_{n=1}^{\delta}\Theta^{r_2(n)}\right) + \epsilon$$

where $l \in \{1, ..., k\}$ is the proposal index and $i \in \{1, ..., M\}$ is the chain index. Θ is the archive matrix containing all accepted samples from the previous $t-1$ iterations. $r_1(j), r_2(n) \in \{1, ..., t_1\}$ and $\Theta^{r_1(j)}$ and $\Theta^{r_2(n)}$ are two accepted samples. e is sampled from $U(-b, b)$, ϵ is sampled from $\mathcal{N}(0, b^*)$ and they are both vectors with the same dimensions as θ_{t-1}. δ is the number of pairs of chains used to generate the proposal. [180] states that a good choice for γ is $\gamma(\delta, d') = 2.38/\sqrt{2\delta d'}$. Differently to DE–MC, MT–DREAM uses multiple pairs of samples from the archive to generate the proposal.

Next, MT–DREAM applies the same crossover scheme as DE–MC, and sets d' to be the number of dimensions that are actually updated in $z^{l,i}$. Namely, in the crossover process, the number of dimensions that were not replaced by an element in θ_{t-1}^{i}. Lastly, as in MT–MC, for each chain, one of the k candidates is selected with probability $p\left(z^{l,i}\right)$, and is accepted with the probability that is described in Equation (5.10).

We presented the MT–DREAM algorithm in general terms. Further details, such as the values of the crossover probability and other model parameters, can be found in [96]. In all the experiments described in the following sections of this chapter, we used the algorithm parameters' default values suggested in [96].

5.3.4 Performance metrics

Once the sampling process is done, we are interested in the quality of the samples. As we will see later, a poorly tuned sampler may produce samples that are not usable, as they were not sampled from the target distribution. One way to analyze the quality of the sampler is through visual inspection. We can plot the samples and inspect whether they have converged and if the different chains have mixed, as in Figure 5.3, for example. This is a reasonable step, but it is not feasible for high–dimensional models. Hence, we are interested in a quantitative metric of the sampler's performance.

This chapter will use the following two standard quantitative metrics, following the Gelman-Rubin (GR) framework [55]. This framework is designed to analyze the performance of a sampler with M parallel MCMC chains and N samples in each chain. In the context of ALDI, M corresponds to the number of particles being propagated in parallel. The same goes for DREAM where we have M interacting chains, which is set to $D+1$, with D being the number of model parameters. For HMC and NUTS, we run each chain separately and use M chains to keep compatibility with ALDI and DREAM.

The first metric is used for convergence diagnostic and is known as *potential scale reduction factor* (PSRF) and is denoted as $\hat{R}$. This diagnostic uses the variability between multiple chains to determine whether the sampler has converged. It consists of the ratio between two quantities that estimate the marginal posterior variance of the parameter, but one overestimates it and the other underestimates it. Each estimate should converge to the actual variance of the target distribution, as the number of samples n increases $\hat{R}$ approaches 1.

For a scalar estimated model parameter θ, and its corresponding samples $\theta_{i,j}$, $i \in \{1, ..., n\}$ and $j \in \{1, ..., m\}$ the within chain variance is defined as

$$\begin{aligned} W &= \frac{1}{m} \sum_{j=1}^{m} s_j^2 \\ s_j^2 &= \frac{1}{n-1} \sum_{i=1}^{n} \left(\theta_{i,j} - \bar{\theta}_{\cdot,j} \right)^2 \\ \bar{\theta}_{\cdot,j} &= \frac{1}{n} \sum_{i=1}^{n} \theta_{i,j}. \end{aligned} \tag{5.11}$$

Similarly, the between chains variance is defined as

$$\begin{aligned} B &= \frac{n}{m-1} \sum_{j=1}^{m} \left(\bar{\theta}_{\cdot,j} - \bar{\theta}_{\cdot,\cdot} \right)^2 \\ \bar{\theta}_{\cdot,\cdot} &= \frac{1}{m} \sum_{j=1}^{m} \theta_{\cdot,j}. \end{aligned} \tag{5.12}$$

The variance of the target distribution is estimated as a weighted average of B and W

$$V = \frac{n-1}{n} W + \frac{1}{n} B. \tag{5.13}$$

This quantity overestimates the marginal posterior variance, whereas the within–chain variance W underestimates it. After a finite number of time steps, the chains have not explored the entire target distribution and will have less variability.

The PSRF is defined as

$$\hat{R} = \sqrt{\frac{V}{W}}. \tag{5.14}$$

This convergence diagnostic is performed on split chains. Each chain is split into two halves and treated as two independent chains. This approach checks mixing and stationarity. If the chains are well mixed, then the two parts of the chains should also mix. If the chains have reached stationarity, their two halves should have reached stationarity.

Usually, the metric described above is used to determine when to stop the sampling process. A threshold $\delta > 1$ is chosen, and the sampling is stopped when $\hat{R} < \delta$. In [54], Gelman et al. recommend setting $\delta = 1.1$, but in practice, a wide range of values is used from 1.003 to 1.3.

The second metric we use to measure the performance of the different sampling algorithms is the *effective sample size* (ESS), which quantifies the number of independent draws generated in the sampling process. If the draws in each chain were independent, we would have had mn independent samples at the end of the sampling process, but in practice, the chains are autocorrelated, and the samples are not independent. Whereas for $\hat{R}$, the lower the better; in the case of the ESS, a higher value indicates that the sampler performs better.

The ESS can be estimated in different ways, and here we follow the one described in [54]. Asymptotically, the ESS is defined as

$$n_{eff} = \frac{mn}{1 + 2\sum_{t=1}^{\infty} \rho_t} \tag{5.15}$$

where ρ_t is the autocorrelation of the samples in a chain with lag t. In practice, we have a finite number of samples and must estimate the infinite sum of the autocorrelations.

To estimate ρ_t, we first define the variogram at lag t

$$W_t = \frac{1}{m\,(n-t)} \sum_{j=1}^{m} \sum_{i=1}^{n} \left(\theta_{i,j} - \theta_{i-t,j}\right)^2 \tag{5.16}$$

and the autocorrelations are estimated with

$$\hat{\rho}_t = 1 - \frac{W_t}{2V}. \tag{5.17}$$

For large t, the sample correlations become too noisy. Thus, we need to use a partial sum and find a T until we perform the summing. T is defined as the time at which the sum of two successive autocorrelations $\hat{\rho}_{2t} + \hat{\rho}_{2t+1}$ is negative. Putting it back in Equation (5.15), we have the definition for the ESS

$$\hat{n}_{eff} = \frac{mn}{1 + 2\sum_{t=1}^{T} \hat{\rho}_t}. \tag{5.18}$$

In our analysis in this chapter, we use a recommended modification to calculate the ESS and $\hat{R}$ described in [178]. The metrics are not directly computed on the parameters' values but rather on their rank normalization. This is defined as

$$z_{i,j} = \Phi^{-1}\left(\frac{r_{i,j} - 3/8}{mn + 1/4}\right) \tag{5.19}$$

with Φ being the cdf of the normal distribution and $r_{i,j}$ the rank of $\theta_{i,j}$.

We use the code implementation from the Python package ArviZ [89].

5.4 Hyperparameters Tuning

We compare the performances of the sampling algorithms in a setting where we know the ground–truth values of the model parameters. Thus, we generate data from the two models and perform the inference on the generated data sets.

As with most inference algorithms, all four samplers compared in this chapter include tunable hyperparameters. Two samplers, NUTS and DREAM, are often used *out of the box* with default hyperparameters, and we follow this recommendation and use these default values. Next, we describe the effect of the hyperparameters on the remaining two algorithms, ALDI and HMC. We demonstrate it on the inference in the EE model.

5.4.1 ALDI hyperparameters

ALDI does not require many design choices and includes only two hyperparameters. These are the step size and the number of particles in the ensemble. Another hyperparameter that is referenced in [51] is the time interval for the Euler–Maruyama algorithm. Still, it is equivalent to the number of samples and can be determined using the metrics described in the previous section.

In [51], the authors suggest setting $N = D + 1$, which we follow in our experiments. Regarding the step size, the authors report setting $\Delta t = 0.01$. For the EE model, the recommended step size led to instabilities in the inference process as the parameters stepped out of range which caused an error. Thus, we had to set the step size much smaller to $\Delta t = 0.0003$. Figure 5.1 presents the ensemble path with this step size. The chains mix well around the ground truth value, but the samples are correlated.

Our solution is adapting the step size during the sampling iterations. We set the initial step size to 0.5 and followed the subsequent heuristic. After performing an iteration, we check whether all the particles are within reasonable values, and if not, we decrease the step size. After 10 such iterations, we increase the step size again. The results of this approach are shown in Figure 5.2. Compared to Figure 5.1, the mixing is much better, and the samples are less correlated. Thus, when we compare the different samplers, we use the version of ALDI with the adaptive step size.

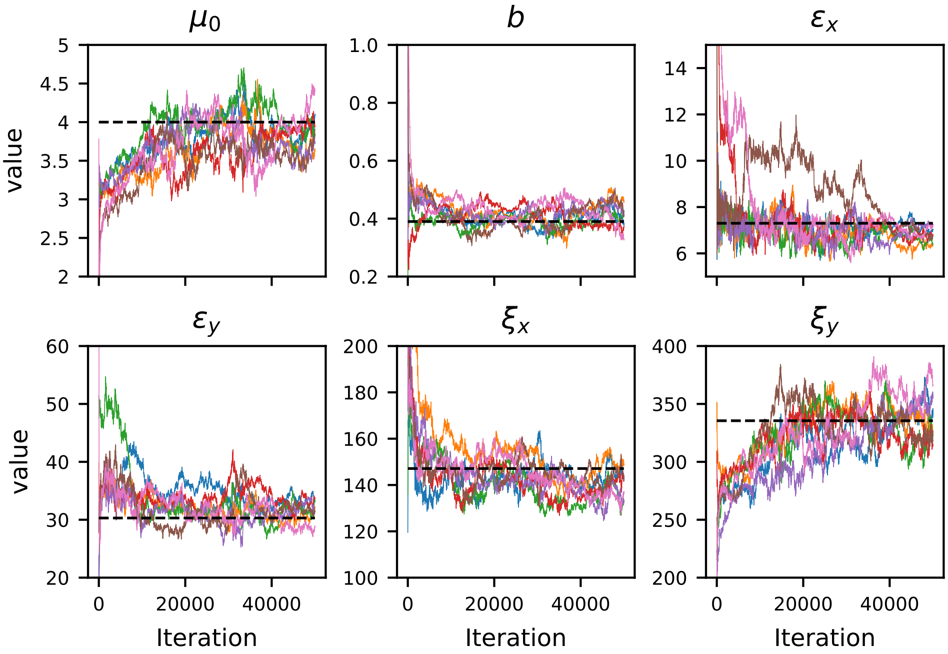


Fig. 5.1.: Samples path of the ALDI sampler with $\Delta t = 0.0003$. The horizontal line indicates the parameter ground–truth value. The chains mix well but contain noticeable autocorrelations.

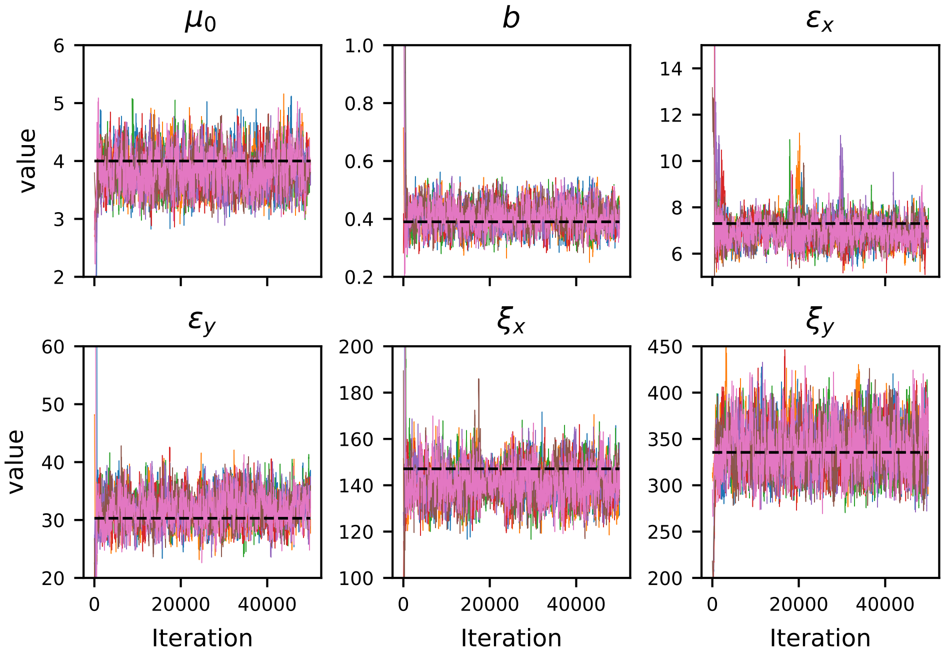


Fig. 5.2.: Samples path of the ALDI sampler with an adaptive step size. The horizontal line indicates the parameter ground–truth value. The chains mix well and contain fewer autocorrelations than with the constant step size.

5.4.2 HMC hyperparameters

HMC includes several hyperparameters. The most important one, which is more of a design choice, is the kinetic energy. Other hyperparameters are the step size ϵ and the number of leapfrog iterations L. We follow the heuristic presented in Chapter 2 and set $\epsilon L = 1$.

Our experiments use the standard Gaussian kinetic energy described in 2.5. We chose the simple case where the mass matrix M is diagonal. A naive choice is the identity matrix. We compare the inference results with the identity and tuned mass matrix.

The results of the HMC sampler with an identity mass matrix are presented in Figure 5.3. It includes 7 chains and the first $10,000$ samples, including the burn–in period and without thinning. We use $\epsilon = 0.01$ and $L = 100$.

Qualitatively, the chains seem to be mixing very well. The chains for all parameters achieve stationarity around the ground–truth value of the model parameters. The samples for xi_x and xi_y are still correlated after $10,000$ iterations. This could be solved by thinning, which will result in fewer samples.

The results of the HMC sampler with a non–identity diagonal mass matrix are presented in Figure 5.4. It includes 7 chains and the first $10,000$ samples, including the burn–in period and without thinning. We use $\epsilon = 0.1$ and $L = 10$.

Qualitatively, the chains mix well in this case and achieve stationarity around the ground–truth value of the model parameters. Unlike in the case of the identity mass matrix, the samples for all parameters are uniformly correlated and converge faster.

Using the customized mass matrix results in a much faster algorithm. When setting $\epsilon = 0.1$ and $L = 10$ with the identity mass matrix, the parameters wander off regimes where they are poorly defined, and the inference fails. To customize the mass matrix, we used our prior knowledge of the model parameters, which are expected to have quite different scales.

The tuning process includes running the HMC sampler a few times, but it is done only once and is then used for the model, assuming all data arrives from similar sources. As the return on investment, we have a sampler that requires 10 times less estimation of the energy and its gradient, with the highest computational cost when generating a sample. Thus, in the following sections, we use HMC with a custom diagonal mass matrix for each model.

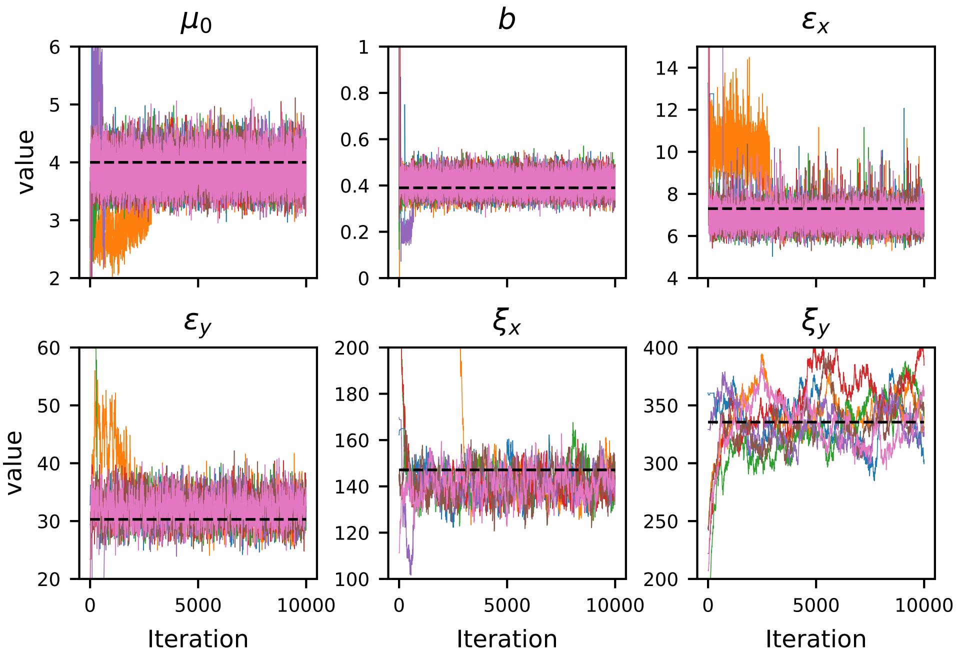


Fig. 5.3.: Samples paths for seven chains using HMC with identity mass matrix and 100 leapfrog steps. The horizontal line indicates the parameter ground–truth value. Overall the chains mix well, but the samples of two of the parameters, ξ_x and xi_y, are heavily autocorrelated.

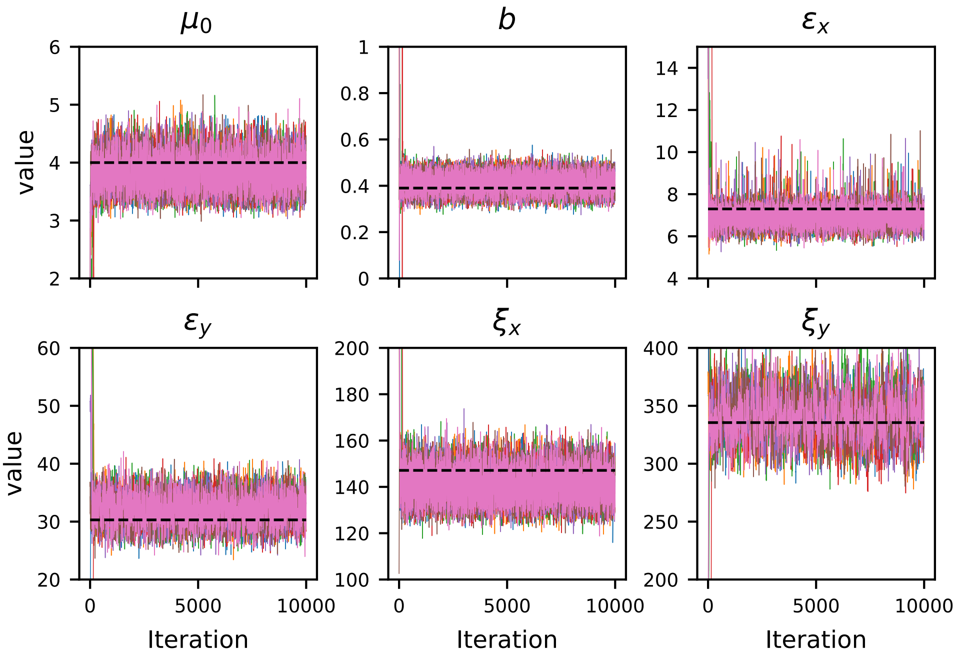


Fig. 5.4.: Samples paths for seven chains using HMC with a customized mass matrix and 10 leapfrog steps. The horizontal line indicates the parameter ground–truth value. The chains are uniformly correlated and converge faster in comparison to using the identity matrix.

5.5 Results

All the results were generated using a Python implementation of the inference algorithms. For the DREAM algorithm, we used the PyDREAM package available under `https://github.com/LoLab-VU/PyDREAM` [158]. For NUTS, we relied on the Python package available under `https://github.com/mfouesneau/NUTS`. HMC and ALDI were implemented by the authors of this work using Python. The implementation for HMC is available under `https://github.com/noashin/pyhmc_minimal`. The implementation for ALDI is available under `https://github.com/noashin/PyALDI`.

All the samplers were run with the same number of M chains. M was set to be $D + 1$ with D the number of model parameters, as required by ALDI. As mentioned before, ALDI and DREAM include interactions between the different chains, whereas, in the case of NUTS and HMC, the chains are independent of each other.

Figures of the sample paths for all the samplers can be found in Appendix A.1. We inspect the performances of the different sampling algorithms in terms of the metrics defined in Section 5.3.4. We start with the potential scale reduction factor $\hat{R}$.

5.5.1 Potential Scale Reduction Factor

As an example, figure 5.6 illustrates the evolution of $\hat{R}$ as a function of the total number of draws for the ALDI sampler and the EE model. The corresponding figures for the rest of the samplers can be found in Appendix A.5. In the case of the ALDI sampler, the convergence of $\hat{R}$ is relatively uniform across the parameters. The initial values of $\hat{R}$ are similar for all the parameters, and it converges to 1 after a similar number of draws.

As mentioned in Section 5.3.4, $\hat{R}$ is used as a stopping criterion. Once it reaches 1.1, the sampling process is stopped. In Table 5.1, we compare the different samples based on the number of draws required to achieve $\hat{R} < 1.1$.

Tab. 5.1.: Number of draws from each sampler that are needed to achieve $\hat{R} < 1.1$ for each parameter of the EE model.

Parameter	ALDI	DREAM	HMC	NUTS
μ_0	13.63×10^3	3.72×10^3	0.3×10^3	6.8×10^3
b	13.63×10^3	3.3×10^3	0.27×10^3	3.75×10^3
ϵ_x	23.57×10^3	3.3×10^3	1.4×10^3	6.87×10^3
ϵ_y	23.57×10^3	2.45×10^3	1.05×10^3	55.26×10^3
ξ_x	23.57×10^3	1.61×10^3	0.21×10^3	161.1×10^3
ξ_y	16.94×10^3	2×10^3	3.61×10^3	866×10^3

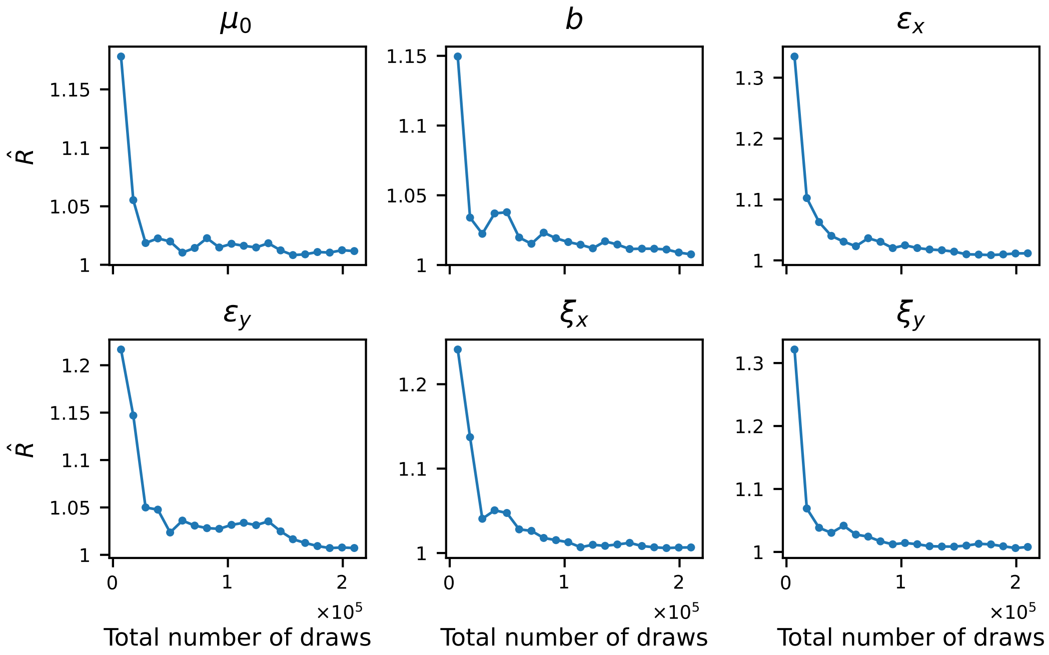


Fig. 5.5.: PSRF evolution for the ALDI sampler for the EE model. All the parameters require a similar amount of draws for the $\hat{R}$ to converge to 1.

The samplers present pretty different performances. Overall, the DREAM parameter requires the least amount of draws to reach the goal of $\hat{R} < 1.1$. Interestingly, in the case of HMC and NUTS, some parameters reach $\hat{R} < 1.1$ much faster than others. In both cases, the parameter that converges the slowest requires more than a hundred times more draws than the parameter that converges the fastest.

In contrast, in the case of ALDI and DREAM, all parameters require similar draws to reach the goal. Comparing the different algorithms, ALDI and DREAM exchange more information between the parameters than HMC and NUTS. The update step of ALDI includes the empirical covariance matrix of the samples, such that the different dimensions influence each other. In the case of HMC and NUTS, the information between the various parameters is exchanged only via the likelihood calculation. In both cases, the mass matrix was set to be diagonal. We expect that if the mass matrix were non–diagonal, the $\hat{R}$ would have converged equally fast along the different dimensions.

We repeat the same analysis for the Scenewalk model. Table 5.2 presents the number of draws required to reach $\hat{R} < 1.1$ for each parameter. The results show the same trends as the analysis of the inference of the EE model. Whereas ALDI and DREAM require similar draws for each parameter, in the case of HMC and NUTS, some parameters converge much slower than others. Furthermore, DREAM requires the least amount of draws on average across parameters.

Tab. 5.2.: Number of draws from each sampler that are needed to achieve $\hat{R} < 1.1$ for each parameter of the SW model.

Parameter	**ALDI**	**DREAM**	**HMC**	**NUTS**
ω_A	10.84×10^3	6.2×10^3	144.76×10^3	$1,125.6 \times 10^3$
ω_A/ω_F	9.68×10^3	7.36×10^3	167.62×10^3	$2,462.4 \times 10^3$
σ_A	9.68×10^3	7.36×10^3	152.38×10^3	$1,130.4 \times 10^3$
σ_F	6.2×10^3	9.68×10^3	144.76×10^3	967.2×10^3
γ	8.52×10^3	5.04×10^3	53.6×10^3	696×10^3
C_F	7.36×10^3	5.04×10^3	6.56×10^3	9.6×10^3
ζ	5.04×10^3	6.2×10^3	0.25×10^3	0.25×10^3

5.5.2 Effective Sample Size

The second metric we look at is the Effective Sample Size. As an example, figure 5.6 illustrates the evolution of the ESS as the number of samples increases for the ALDI sampler. All parameters reach the 400 ESS limit around $100,000$ samples. The evolution of the ESS for the other samplers can be found in Appendix A.3.

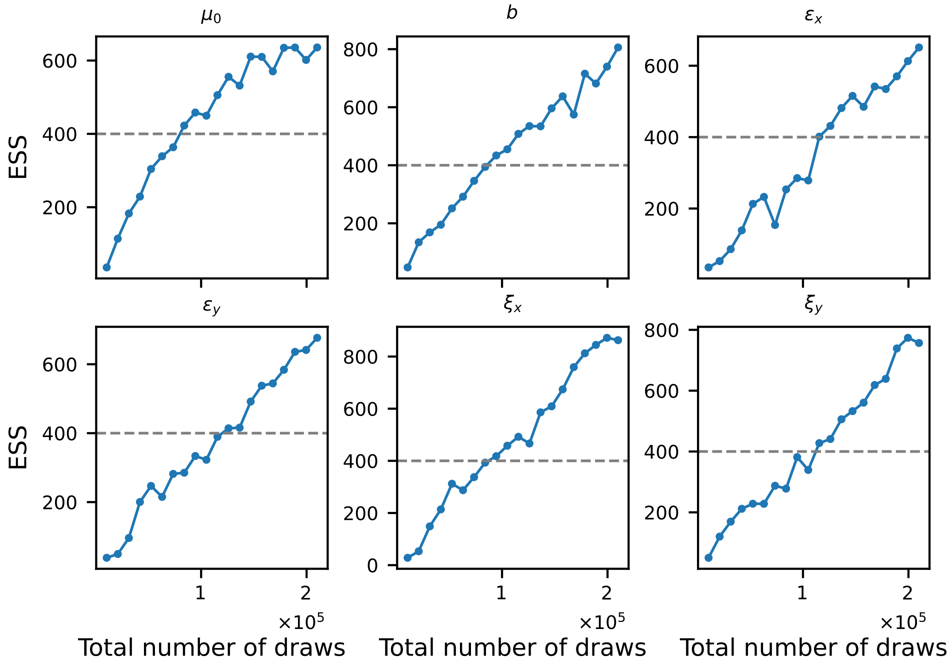


Fig. 5.6.: ESS as a function of the number of samples drawn by the ALDI algorithm for the EE model. The horizontal line indicates 400 ESS.

In Section 5.3.4, we mentioned that a threshold for ESS is 400. Table 5.3 presents the number of draws required to reach an ESS of 400 for each sampler for each parameter. Similarly to the case of $\hat{R}$, ALDI and DREAM require a similar amount of draws to reach an ESS of 400, whereas HMC and NUTS display a large variety between the parameters.

Tab. 5.3.: Number of draws from each sampler needed to achieve 400 ESS for each parameter of the EE model.

Parameter	ALDI	DREAM	HMC	NUTS
μ_0	81.4×10^3	14.25×10^3	2.8×10^3	62.8×10^3
b	85.7×10^3	14.7×10^3	1.96×10^3	49.6×10^3
ϵ_x	114.28×10^3	14.4×10^3	0.4×10^3	21.18×10^3
ϵ_y	124.28×10^3	11.3×10^3	4.48×10^3	280×10^3
ξ_x	90×10^3	8.7×10^3	1.8×10^3	752×10^3
ξ_y	112.85×10^3	9.6×10^3	22.3×10^3	5.6×10^6

For practical uses, we are interested in a sampler that achieves the ESS quickly with fewer draws and in a sampler that generates a draw fast. Next, we look into the computations required to generate a draw from each sampler.

For most models, the most demanding computation is evaluating the likelihood. For ALDI and DREAM, the number of draws is identical to the number of likelihood evaluations. In contrast, NUTS and HMC require multiples of the likelihood per draw.

Here, HMC includes 10 leapfrog steps, each estimating both the likelihood and its derivative. The number of leapfrog iterations in the NUTS sampler is not constant. In our experiments, NUTS included 4 leapfrog steps on average. Each leapfrog step involves evaluating both the likelihood and its derivative, which can be assumed to be similar.

Tab. 5.4.: Number of likelihood evaluations from each sampler needed to achieve 400 ESS for each parameter of the EE model.

Parameter	ALDI	DREAM	HMC	NUTS
μ_0	81.4×10^3	14.25×10^3	56×10^3	351.44×10^3
b	85.7×10^3	14.7×10^3	39.2×10^3	90.72×10^3
ϵ_x	114.28×10^3	14.4×10^3	8×10^3	125.52×10^3
ϵ_y	124.28×10^3	11.3×10^3	89.6×10^3	1.326×10^6
ξ_x	90×10^3	8.7×10^3	36×10^3	4.22×10^6
ξ_y	112.85×10^3	9.6×10^3	446×10^3	31.2×10^6

Table 5.4 presents the likelihood estimations required to reach the ESS goal. The DREAM algorithm outperforms the other samplers. We repeat the same analysis for the SW model. Table 5.5 presents the likelihood estimations required for each parameter to reach ESS of 400. The results show the same trends as the analysis of the inference of the EE model. Whereas ALDI and DREAM require similar estimations for each parameter, in the case of HMC and NUTS, some parameters converge much slower than others. Furthermore, DREAM requires the least amount of estimations on average across parameters.

Tab. 5.5.: Number of likelihood evaluations from each sampler needed to achieve 400 ESS for each parameter of the SW model.

Parameter	ALDI	DREAM	HMC	NUTS
ω_A	39.6×10^3	28.8×10^3	$81,600 \times 10^3$	$17,628 \times 10^3$
ω_A/ω_F	48.8×10^3	37.6×10^3	$9,600 \times 10^3$	$24,549 \times 10^3$
σ_A	46.4×10^3	39.2×10^3	$7,360 \times 10^3$	$18,480 \times 10^3$
σ_F	53.6×10^3	38.4×10^3	$7,040 \times 10^3$	$16,968 \times 10^3$
γ	38.8×10^3	33.6×10^3	$4,800 \times 10^3$	$16,506 \times 10^3$
C_F	36.8×10^3	23.2×10^3	448×10^3	378×10^3
ζ	39.6×10^3	23.2×10^3	8×10^3	2.62×10^3

5.5.3 Discussion

In this chapter, we compared different samplers for posterior inference in human scene–viewing models. We chose four samplers – ALDI, DREAM, HMC, and NUTS. All the samplers require a model with a computable likelihood function and prior functions over the model's parameters. Other than DREAM, the samplers also require a computable derivative of the likelihood. Other than that, there is no restriction over the likelihood structure.

We chose the field of modeling human scene–viewing as Bayesian modeling gains popularity in the area, but there are no best practices for performing inference. We completed the analysis on two contemporary models, the Scenewalk model and the Exploration–Exploitation model. Both models suit the chosen samplers as their likelihood functions and their derivatives are computable. Both models also have a similar number of parameters. The SW model has seven parameters, and the EE model has six.

Using data sets generated from the models, we first showed that all samplers concentrate around the ground–truth values of the parameters after sufficient samples and that multiple chains mix well. Then we compared the performance of the samplers using the Gelman–Rubin framework. This framework estimates the potential scale reduction factor and the effective sample size and is considered the standard in the field.

The samplers performed similarly on both models. The ALDI and DREAM demonstrated consistent results over the different model parameters, whereas HMC and NUTS had mixed results. The algorithms themselves explain this. In each DREAM iteration, only some dimensions are updated using the crossover scheme. In ALDI, the update step of the sampler includes the empirical covariance matrix of the past samples.

HMC can be adapted to encourage uniformity over the different parameters. In our experiments, the mass matrix was set to be diagonal. Using a well–designed non–diagonal mass matrix could result in more consistent results over the different parameters. Creating a non–diagonal mass matrix is much more complex than a diagonal matrix. One approach could be using the empirical covariance matrix of a different sampler. In the case of the NUTS algorithm, the mass matrix is set to be the identity. This sampler is encouraged to be used as an *out of the box* solution; hence, we did not temper it.

Overall, the NUTS algorithm performed the worst. This shows the importance of a well–designed mass matrix for efficient inference in the HMC framework. In our case, it was beneficial to have several iterations of adapting the mass matrix of HMC over using the NUTS algorithm, which does not require such adaptation. Here, we used the prior knowledge that the parameters of the models had different scales and set the mass matrix accordingly. Without this intimate knowledge of the model, designing the mass matrix would have been much more complicated. If the model has many more parameters with unknown scales using NUTS may be easier and faster than attempting to fiddle with the mass matrix of HMC.

In the case of the EE model, looking just at the number of draws necessary to reach the desired $\hat{R}$ or ESS, HMC performs the best. As mentioned in the previous chapter, we are interested in a computationally efficient sampler. HMC involves estimating the likelihood and its derivative ten times. Although it is a constant factor, it is noteworthy, which makes HMC significantly slower than DREAM. In the SW model, the difference in the number of draws between the parameters is higher than in the EE model. This makes the DREAM sampler the most efficient regarding the number of draws and not only the number of likelihood estimations.

Overall, all the inference algorithms resulted in a very low ESS compared to the number of draws, even in the case of DREAM. This could indicate several problems in the data and the inference process. One issue could be correlations between the model parameters, which could affect the posterior inference even when the inference is made with data that the model generated.

Another issue could be the small sample size. It could be that more data is required for efficient inference of the model parameters for the discussed models. The sample size used here was not random. We used generated data sets with a similar size to experimental data sets. Hence, although many draws are required to achieve sufficient ESS, it is possible under the constraints of experimental settings.

Conclusions and Outlook 6

6.1 Summary and Discussion

In this dissertation, we contributed to the line of work on Bayesian point processes modeling. We introduced new models and inference algorithms. We also discussed the advantages and disadvantages of different inference algorithms in the context of computational cognitive neuroscience, specifically human gaze modeling.

6.1.1 Models and Inference Methods

In Chapter 3, we presented a new nonlinear Hawkes model. We chose to work on the nonlinear Hawkes process since it allows us to model phenomena where the interactions between events are excitatory and inhibitory. The nonlinearity is achieved by applying the sigmoid function to the linear intensity function. The innovation of our model is modeling the background rate and inter–events effects with Gaussian Processes and explicitly assuming exponential decay of the self–effects. This flexible framework allows the functional form of the effects to change over time. The effect of an event may start as excitatory but become inhibitory after some time. Last, including the GP priors, results in a simple structure of the linear intensity function, characterized by a kernel function.

In Chapter 4, we contributed to the field of human gaze modeling. We presented a model for exploration and exploitation in natural scene viewing. Our model hypothesizes that the saccades generated during scene viewing can be characterized as either "exploitative" or "exploratory." The saccade is relatively short in the first case, and the viewer accumulates more information in the same region of interest. In the latter, the saccade length is longer, and the viewer shifts their attention to another area in the image.

We used the PG augmentation scheme for both models to achieve a model for which we can derive a Gibbs sampler. For the case of the EE model, it was enough to augment the model with the set of PG variables. The NH–GPS model needed two sets of augmenting variables. First, we augmented the observations with a set of PG variables. Then, we identified the auxiliary marked Poisson process and its realizations and associated marks.

For The EE model, we could not sample from all the conditional posteriors of the model's parameters. For these parameters, we used an HMC step within the Gibbs sampler. In the case of the NH–GPS model, we could derive an efficient VI algorithm.

We showed that despite the inherent approximations, the VI algorithm converged fast and achieved similar results to the Gibbs sampler.

In Chapter 5, we attempted to provide practitioners with tools to choose the appropriate inference method for their problem. We chose human scan path modeling as a use case, specifically our EE model and the known Sceneviewing model. In Chapters 3 and 4, we derived custom inference algorithms; in this chapter, we compared four samplers that can be applied to any model given a computable likelihood function. These samplers were HMC, NUTS, ALDI, and DREAM.

6.1.2 Evaluation and Applications

In Chapters 3 and 5, we first established the correctness of the inference algorithms. We demonstrated this by performing inference on data generated from the model with known ground truth. For both algorithms, the inference recovered the values of the model's parameters.

In Chapter 3, we applied the NH–GPS model to data from different domains. We chose domains where prior work existed to compare our model to others. First, we showed that our model performs better than existing models in the field of modeling crime report data. Then we used three datasets of neuronal activity. We showed that our model achieved comparable or better results than existing models, also in the case of multivariate data. The performance of the models was measured using the test log–likelihood and the p–value of the KS test.

In Chapter 4, we used the test log–likelihood to measure the model's performance. In this case, we did not compare our model to existing models but compared different versions of it. These versions represented different hypotheses regarding the mechanism underlying scan path generation. Using this metric, we showed that the full model we suggested is the most likely out of the tested versions.

In this chapter, we also used another approach to measure the model's performance. We expected that data generated from the fitted resembled the observed data. First, we showed that the data generated by the model recovered the empirical saliency map. Then, we showed that the model captured the distribution of saccade lengths of the observed data. Furthermore, as we fitted our model to data from individual subjects, we could capture inter–individual differences. Last, we showed that the model could capture the distribution of saccade directions but failed to model the saccade direction change. The latter was expected due to the structure of the model, which did not consider the direction of the previous saccade.

To compare the different sampling algorithms in Chapter 5, we used the standard metrics of effective sample size (ESS) and the potential scale reduction factor ($\hat{R}$). As some of the algorithms need multiple estimations of the likelihood function or its gradient to generate one sample, we compared the different samplers in terms of the number of likelihood estimations required to reach the target ESS and $\hat{R}$. For both models considered, the DREAM samplers achieved the best results. Furthermore, we could easily apply the algorithm's existing code implementation to our model.

6.2 Outlook

The scientific process involves developing questions and figuring out how to answer them. As the doctorate research period is limited, not all of these questions can be answered in the given time frame. This means that the research presented in this dissertation is only a subset of many ideas conceptualized during the last five years. We hope that our work is exciting and relevant enough to be continued, and we next present some of the ideas we had to leave behind to complete this dissertation on schedule.

6.2.1 Efficient inference for the NH–GPS model

Whereas theoretical mathematics is constrained only by the researcher's imagination and maybe some axioms, numerical analysis suffers from much more prosaic limitations. All the models and algorithms presented in the previous chapters were eventually translated to machine–readable code, and the experiments were executed on computers. Thus, we were limited by space and time—the computation resources that were available to us.

The most demanding computation we executed was calculating the memory kernel of the NH–GPS model, presented in Equation (3.13). This involves a double summation of all the data and is quadratic in the number of events. Furthermore, this computation is also done with respect to the set of induced points and the integration points. In most cases, the number of integration points is larger than the number of events in the data, and it becomes the limiting factor.

In the research presented here, we only included temporal data, which is one–dimensional. Theoretically, the NH–GPS model can be easily extended to higher dimensional data. This includes, of course, spatiotemporal data, which are three-dimensional. Hawkes processes were already successfully applied to spatiotemporal data in different fields from earthquakes [137, 126] to social media and crime prediction [184, 194].

When applying our model to spatiotemporal data, we encountered difficulties computing the memory kernels. In cases where the memory kernel was computed successfully, its inversion proved too demanding for the computational resources available to us. We hope that this can be solved with future research.

One direction is developing a more efficient calculation of the memory kernel. A possible modification is considering the decay of the process when computing the memory kernel. Due to the explicit decay of the memory kernel, if two events are far enough from each other, effectively, their effect on each other is 0. It may be beneficial to incorporate domain–specific knowledge and explicitly assume that if $t - t_n > c$, then $g\left(t - t_n\right) \exp\left(-\alpha\left(t - t_n\right)\right) = 0$. This will reduce the number of computations needed for the memory kernel.

The process described above results in a sparse covariance kernel, which is helpful for GP estimation. For example, one could use sparse Cholesky factor methods [177, 39]. The main challenge of this approach is constructing a valid compactly supported covariance and building a positive semidefinite matrix. Potential solutions can be found in Buhmann [26] and Melkumyan and Ramos [119]. A thorough review of approximate methods for efficient GP computation can be found in Liu, Ong, Shen, and Cai [107].

6.2.2 Spatiotemporal Hawkes process Models for Scene Viewing

In Chapter 4, we presented a new model for scene viewing. This model assumes that a fixation depends only on the previous two fixations. In Sections 4.4 and 4.5, we discussed the limitations of this assumption. We believe that the experimental effects that our model did not capture may be better modeled with a spatiotemporal Hawkes process.

Furthermore, most approaches designed for scan path modeling currently focus on the spatial features of the fixations and ignore the temporal aspect. Almost all the models covered in the thorough comparison of Kümmerer and Bethge [90] model only the location of the fixations. Two notable exceptions are the LATEST model [169] and the version of the Scenewalk model described in Schwetlick, Backhaus, and Engbert [154].

We suggest that a spatiotemporal implementation of the NH–GPS model described in Chapter 3 could join the slowly growing family of models for both saccade locations and durations. We next present some ideas and thoughts regarding applying the NH–GPS model to scene viewing.

In Chapter 3 we defined the linear intensity function

$$\phi\left(t\right) = s\left(t\right) + \sum_{t_n < t} g\left(t - t_n\right) \exp\left(-\alpha\left(t - t_n\right)\right).$$

A spatiotemporal variation of it could take the following form

$$\phi\left(t,z\right)=s\left(t,z\right)+\sum_{t_n<t,z_n<z}g\left(t-t_n,z-z_n\right)\exp\left(-\alpha\left(t-t_n\right)-\beta\left(z-z_n\right)\right),$$

and it is left to design the background rate and self–effects functions.

For the spatiotemporal Hawkes process, it is common to separate the effects of the past events in time and space [137, 145, 184, 82]. This means that we can write the self–effects function as

$$g\left(t-t_n,z-z_n\right)=f\left(t-t_n\right)\exp\left(-\alpha\left(t-t_n\right)\right)h\left(z-z_n\right)\exp\left(-\beta\left(z-z_n\right)\right).$$

The NH–GPS model takes as input only the events. This is insufficient to model scan paths, as they are coupled to a specific image. Thus, if we wish to model scan paths using a Hawkes process, we must integrate the image into the intensity function. Similarly to what we did in Chapter 4, we suggest including information about the image via the static saliency function $sal\left(z\right)$.

First, the saliency could be integrated into the background rate function. One approach is to assume that the spatial aspect of the background rate is proportional to the saliency

$$s\left(t,z\right)=s\left(t\right)\mu^{*}sal\left(z\right).$$

Next, the saliency should be integrated into the self–effects function. One can assume that the self–effects depend only on the saliency value at location z and write the self–effects as $sal\left(z\right)g\left(t-t_n,z-z_n\right)$. Another assumption could be that the impact of the previous event depends on the difference between the saliency at the past fixation and the current fixation z and write the self–effects as $\left(sal\left(z\right)-sal\left(z_n\right)\right))\,g\left(t-t_n,z-z_n\right)$. Instead of the difference between the saliencies, one can also use the ratio.

The usage of the Hawkes process model for scene viewing has excellent potential. As mentioned in the previous section extending the model to three dimensions comes at a high computational cost. We hope the suggestions in the last section prove helpful and enable the implementation of our model's spatiotemporal extension and applying it to scan path modeling.

Bibliography

[1] Ryan Prescott Adams, Iain Murray, and David J C MacKay. „Tractable Nonparametric Bayesian Inference in Poisson Processes With Gaussian Process Intensities“. In: *Proceedings of the 26th Annual International Conference on Machine Learning*. Ed. by Andrea Pohoreckyj Danyluk, Léon Bottou, and Michael L. Littman. Vol. 382. ACM International Conference Proceeding Series. Montreal, Canada: ACM, 2009, pp. 9–16.

[2] Shun-ichi Amari and Hiroshi Nagaoka. *Methods of Information Geometry*. Vol. 191. American Mathematical Soc., 2000.

[3] Ifigeneia Apostolopoulou, Scott Linderman, Kyle Miller, and Artur Dubrawski. „Mutually Regressive Point Processes“. In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Vancouver, Canada: Curran Associates, Inc., 2019.

[4] Emmanuel Bacry, Iacopo Mastromatteo, and Jean-François Muzy. „Hawkes Processes in Finance“. In: *Market Microstructure and Liquidity* 1.01 (2015), p. 1550005.

[5] A Terry Bahill. „Most Naturally Occurring Human Saccades Have Magnitudes of 15 Deg or Less“. In: *Invest. Ophthalmol* 14 (1975), pp. 468–469.

[6] Thomas Bayes. „LII. An Essay Towards Solving a Problem in the Doctrine of Chances. By the Late Rev. Mr. Bayes, FRS Communicated by Mr. Price, in a Letter to John Canton, AMFR S“. In: *Philosophical Transactions of the Royal Society of London* 53 (1763), pp. 370–418.

[7] Charles H Bennett. „Mass Tensor Molecular Dynamics“. In: *Journal of Computational Physics* 19.3 (1975), pp. 267–279.

[8] Oded Berger-Tal, Jonathan Nathan, Ehud Meron, and David Saltz. „The Exploration-Exploitation Dilemma: A Multidisciplinary Framework“. In: *PloS One* 9.4 (2014), e95693.

[9] Julian Besag. „Digital Image Processing: Towards Bayesian Image Analysis“. In: *Journal of Applied Statistics* 16.3 (1989), pp. 395–407.

[10] Julian Besag and Peter Clifford. „Generalized Monte Carlo Significance Tests“. In: *Biometrika* 76.4 (1989), pp. 633–642.

[11] Julian Besag and Peter Clifford. „Sequential Monte Carlo P-Values“. In: *Biometrika* 78.2 (1991), pp. 301–304.

[12] Julian Besag, Jeremy York, and Annie Mollié. „Bayesian Image Restoration, With Two Applications in Spatial Statistics“. In: *Annals of the Institute of Statistical Mathematics* 43 (1991), pp. 1–20.

[13] Christopher M Bishop. *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[14] James W Bisley and Koorosh Mirpour. „The Neural Instantiation of a Priority Map“. In: *Current Opinion in Psychology* (2019), pp. 108–112.

[15] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. „Variational Inference: A Review for Statisticians“. In: *Journal of the American Statistical Association* 112.518 (2017), pp. 859–877.

[16] Giuseppe Boccignone and Mario Ferraro. „Modelling Gaze Shift as a Constrained Random Walk“. In: *Physica A: Statistical Mechanics and Its Applications* 331.1-2 (2004), pp. 207–218.

[17] Ali Borji and Laurent Itti. „State-of-the-Art in Visual Attention Modeling“. In: *IEEE Transactions On Pattern Analysis And Machine Intelligence* 35.1 (2012), pp. 185–207.

[18] Ali Borji and Laurent Itti. „State-of-the-Art in Visual Attention Modeling“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.1 (2012), pp. 185–207.

[19] Clive G Bowsher. „Modelling Security Market Events in Continuous Time: Intensity Based, Multivariate Point Process Models“. In: *Journal of Econometrics* 141.2 (2007), pp. 876–912.

[20] James Bradbury, Roy Frostig, Peter Hawkins, Matthew J Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. „JAX: Composable Transformations of Python+NumPy Programs“. In: 2018. URL: `http://github.com/google/jax`.

[21] Bruno G Breitmeyer, Walter Kropfl, and Bela Julesz. „The Existence and Role of Retinotopic and Spatiotopic Forms of Visual Persistence“. In: *Acta Psychologica* 52.3 (1982), pp. 175–196.

[22] Pierre Brémaud and Laurent Massoulié. „Stability of Nonlinear Hawkes Processes“. In: *The Annals of Probability* (1996), pp. 1563–1588.

[23] Anders Brix and Peter J Diggle. „Spatiotemporal Prediction for Log-Gaussian Cox Processes“. In: *Journal of the Royal Statistical Society: Series B* 63.4 (2001), pp. 823–841.

[24] Anthony E Brockwell and Joseph B Kadane. „Identification of Regeneration Times in McMc Simulation, With Application to Adaptive Schemes“. In: *Journal of Computational and Graphical Statistics* 14.2 (2005), pp. 436–458.

[25] Emery N Brown, Riccardo Barbieri, Valérie Ventura, Robert E Kass, and Loren M Frank. „The Time-Rescaling Theorem and Its Application to Neural Spike Train Data Analysis“. In: *Neural Computation* 14.2 (2002), pp. 325–346.

[26] Martin Buhmann. „A New Class of Radial Basis Functions With Compact Support“. In: *Mathematics of Computation* 70.233 (2001), pp. 307–318.

[27] Zoya Bylinskii, Tilke Judd, Aude Oliva, Antonio Torralba, and Frédo Durand. „What Do Different Evaluation Metrics Tell Us About Saliency Models?“ In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.3 (2018), pp. 740–757.

[28] Leo M Chalupa and John S Werner. *The Visual Neurosciences, Vols. 1 & 2*. MIT Press, 2004.

[29] Hee Min Choi and James P Hobert. „The Pólya-Gamma Gibbs Sampler for Bayesian Logistic Regression Is Uniformly Ergodic“. In: *Electronic Journal of Statistics* 7 (2013), pp. 2054–2064.

[30] Jonathan D Cohen, Samuel M McClure, and Angela J Yu. „Should I Stay or Should I Go? How the Human Brain Manages the Trade-Off Between Exploitation and Exploration“. In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 362.1481 (2007), pp. 933–942.

[31] Antoine Coutrot, Janet H Hsiao, and Antoni B Chan. „Scanpath Modeling and Classification With Hidden Markov Models“. In: *Behavior Research Methods* 50.1 (2018), pp. 362–379.

[32] David R Cox. „Some Statistical Methods Connected With Series of Events“. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 17.2 (1955), pp. 129–157.

[33] Kenneth James William Craik. *The Nature of Explanation*. Cambridge University Press, 1943.

[34] Lehel Csató, Manfred Opper, and Ole Winther. „TAP Gibbs Free Energy, Belief Propagation and Sparsity.“ In: *NIPS*. 2001, pp. 657–663.

[35] Daryl J Daley and David Vere-Jones. *An Introduction to the Theory of Point Processes Vol. I*. Second. Probability and its Applications (New York). New York: Springer-Verlag, 2003.

[36] Angelos Dassios and Hongbiao Zhao. „A Dynamic Contagion Process“. In: *Advances in Applied Probability* 43.3 (2011), pp. 814–846.

[37] Arthur P Dempster, Nan M Laird, and Donald B Rubin. „Maximum Likelihood From Incomplete Data via the EM Algorithm“. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1 (1977), pp. 1–22.

[38] Jean Diebolt and Christian P Robert. „Estimation of Finite Mixture Distributions Through Bayesian Sampling“. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 56.2 (1994), pp. 363–375.

[39] Yi Ding, Risi Kondor, and Jonathan Eskreis-Winkler. „Multiresolution Kernel Approximation for Gaussian Process Regression“. In: *Advances in Neural Information Processing Systems* 30 (2017).

[40] Christian Donner and Manfred Opper. „Efficient Bayesian Inference for a Gaussian Process Density Model“. In: *Conference on Uncertainty in Artificial Intelligence*. Ed. by Amir Globerson and Ricardo Silva. PMLR. Monterey, CA, USA: AUAI Press, 2018.

[41] Christian Donner and Manfred Opper. „Efficient Bayesian Inference of Sigmoidal Gaussian Cox Processes“. In: *The Journal of Machine Learning Research* 19.1 (2018), pp. 2710–2743.

[42] Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. „Hybrid Monte Carlo“. In: *Physics Letters B* 195.2 (1987), pp. 216–222.

[43] Benedikt V Ehinger, Lilli Kaufhold, and Peter König. „Probing the Temporal Dynamics of the Exploration–Exploitation Dilemma of Eye Movements“. In: *Journal of Vision* 18.3 (2018), pp. 6–6.

[44] Wolfgang Einhäuser and Peter König. „Getting Real – Sensory Processing of Natural Stimuli“. In: *Current Opinion in Neurobiology* 20.3 (2010), pp. 389–395.

[45] Ahmed Elgammal, Ramani Duraiswami, David Harwood, and Larry S Davis. „Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance“. In: *Proceedings of the IEEE* 90.7 (2002), pp. 1151–1163.

[46] Ralf Engbert, Hans A Trukenbrod, Simon Barthelmé, and Felix A Wichmann. „Spatial Statistics and Attentional Dynamics in Scene Viewing“. In: *Journal of Vision* 15.1 (2015), pp. 14–14.

[47] Michael Evans. „Chaining via Annealing“. In: *The Annals of Statistics* (1991), pp. 382–393.

[48] John M Findlay and Iain D Gilchrist. *Active Vision: The Psychology of Looking and Seeing*. Oxford University Press, 2003.

[49] Douglas Frost and Ernst Pöppel. „Different Programming Modes of Human Saccadic Eye Movements as a Function of Stimulus Eccentricity: Indications of a Functional Subdivision of the Visual Field“. In: *Biological Cybernetics* 23.1 (1976), pp. 39–48.

[50] Ricardo Ramos Gameiro, Kai Kaspar, Sabine König, Sontje Nordholt, and Peter König. „Exploration and Exploitation in Natural Viewing Behavior“. In: *Scientific Reports* 7.1 (2017), pp. 1–23.

[51] Alfredo Garbuno-Inigo, Nikolas Nüsken, and Sebastian Reich. „Affine Invariant Interacting Langevin Dynamics for Bayesian Inference“. In: *SIAM Journal on Applied Dynamical Systems* 19.3 (2020), pp. 1633–1658.

[52] Alan E Gelfand and Sujit K Sahu. „On Markov Chain Monte Carlo Acceleration“. In: *Journal of Computational and Graphical Statistics* 3.3 (1994), pp. 261–276.

[53] Alan E Gelfand and Adrian FM Smith. „Sampling-Based Approaches to Calculating Marginal Densities“. In: *Journal of the American Statistical Association* 85.410 (1990), pp. 398–409.

[54] Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Akti Vehtari, and Donald B. Rubin. *Bayesian Data Analysis*. Third. Chapman & Hall/CRC Texts in Statistical Science Series. Boca Raton, Florida, 2013. DOI: 10.1201/b16018.

[55] Andrew Gelman and Donald B Rubin. „Inference From Iterative Simulation Using Multiple Sequences“. In: *Statistical Science* (1992), pp. 457–472.

[56] Stuart Geman and Donald Geman. „Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6 (1984), pp. 721–741.

[57] Felipe Gerhard, Moritz Deger, and Wilson Truccolo. „On the Stability and Dynamics of Stochastic Spiking Neuron Models: Nonlinear Hawkes Process and Point Process GLMs“. In: *PLOS Computational Biology* 13.2 (Feb. 2017), pp. 1–31.

[58] Walter R Gilks, David G Clayton, David J Spiegelhalter, Nicky G Best, Alexander J McNeil, Linda D Sharples, and AJ Kirby. „Modelling Complexity: Applications of Gibbs Sampling in Medicine“. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 55.1 (1993), pp. 39–52.

[59] Walter R Gilks and Pascal Wild. „Adaptive Rejection Sampling for Gibbs Sampling“. In: *Journal of the Royal Statistical Society: Series C* 41.2 (1992), pp. 337–348.

[60] Mark Girolami and Ben Calderhead. „Riemann Manifold Langevin and Hamiltonian Monte Carlo Methods“. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 73.2 (2011), pp. 123–214.

[61] Jonathan Goodman and Jonathan Weare. „Ensemble Samplers With Affine Invariance“. In: *Communications in Applied Mathematics and Computational Science* 5.1 (2010), pp. 65–80.

[62] Peter J Green. „Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination“. In: *Biometrika* 82.4 (1995), pp. 711–732.

[63] Andreas Griewank. „On Automatic Differentiation“. In: *Mathematical Programming: Recent Developments and Applications* 6.6 (1989), pp. 83–107.

[64] Heikki Haario, Eero Saksman, and Johanna Tamminen. „An Adaptive Metropolis Algorithm“. In: *Bernoulli* (2001), pp. 223–242.

[65] W Keith Hastings. „Monte Carlo Sampling Methods Using Markov Chains and Their Applications“. In: (1970).

[66] Alan G Hawkes. „Cluster Models for Earthquakes-Regional Comparisons“. In: *Bull. Int. Stat. Inst.* 45.3 (1973), pp. 454–461.

[67] Alan G Hawkes. „Hawkes Processes and Their Applications to Finance: A Review“. In: *Quantitative Finance* 18.2 (2018), pp. 193–198.

[68] Alan G Hawkes. „Point Spectra of Some Mutually Exciting Point Processes“. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 33.3 (1971), pp. 438–443.

[69] Alan G Hawkes. „Spectra of Some Mutually Exciting Point Processes With Associated Variables“. In: *Stochastic Point Processes* (1972), pp. 261–271.

[70] Alan G Hawkes. „Spectra of Some Self-Exciting and Mutually Exciting Point Processes“. In: *Biometrika* 58.1 (1971), pp. 83–90.

[71] Alan G Hawkes and David Oakes. „A Cluster Process Representation of a Self-Exciting Process“. In: *Journal of Applied Probability* 11.3 (1974), pp. 493–503.

[72] Jens R Helmert, Markus Joos, Sebastian Pannasch, and Boris M Velichkovsky. „Two Visual Systems and Their Eye Movements: Evidence From Static and Dynamic Scene Perception“. In: *Proceedings of the Annual Meeting of the Cognitive Science Society*. 2005, pp. 2283–2288.

[73] John M Henderson. „Human Gaze Control During Real-World Scene Perception“. In: *Trends in Cognitive Sciences* 7.11 (2003), pp. 498–504.

[74] James Hensman, Alexander G Matthews, Maurizio Filippone, and Zoubin Ghahramani. „McMc for Variationally Sparse Gaussian Processes“. In: *arXiv Preprint arXiv:1506.04000* (2015).

[75] Matthew D Hoffman and Andrew Gelman. „The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo“. In: *J. Mach. Learn. Res.* 15.1 (2014), pp. 1593–1623.

[76] Jenq-Neng Hwang, Shyh-Rong Lay, and Alan Lippman. „Nonparametric Multivariate Density Estimation: A Comparative Study“. In: *IEEE Transactions on Signal Processing* 42.10 (1994), pp. 2795–2810.

[77] Hemant Ishwaran and Lancelot F James. „Computational Methods for Multiplicative Intensity Models Using Weighted Gamma Processes: Proportional Hazards, Marked Point Processes, and Panel Count Data“. In: *Journal of the American Statistical Association* 99.465 (2004), pp. 175–190.

[78] Laurent Itti and Christof Koch. „A Saliency-Based Search Mechanism for Overt and Covert Shifts of Visual Attention“. In: *Vision Research* 40.10-12 (2000), pp. 1489–1506.

[79] Laurent Itti and Christof Koch. „Computational Modelling of Visual Attention“. In: *Nature Reviews Neuroscience* 2.3 (2001), pp. 194–203.

[80] Laurent Itti, Christof Koch, and Ernst Niebur. „A Model of Saliency-Based Visual Attention for Rapid Scene Analysis“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.11 (1998), pp. 1254–1259.

[81] 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.

[82] Mikyoung Jun and Scott Cook. „Flexible Multivariate Spatio-Temporal Hawkes Process Models of Terrorism“. In: *arXiv Preprint arXiv:2202.12346* (2022).

[83] Yan Y Kagan. „Likelihood Analysis of Earthquake Catalogues“. In: *Geophysical Journal International* 106.1 (1991), pp. 135–148.

[84] John O F Kingman. *Poisson Processes*. Oxford Studies in Probability. Clarendon Press, 1992. ISBN: 9780191591242.

[85] Raymond M Klein. „Inhibition of Return“. In: *Trends in Cognitive Sciences* 4.4 (2000), pp. 138–147.

[86] Raymond M Klein and William J MacInnes. „Inhibition of Return Is a Foraging Facilitator in Visual Search“. In: *Psychological Science* 10.4 (1999), pp. 346–352.

[87] Peter E Kloeden and Eckhard Platen. „Stochastic Differential Equations“. In: *Numerical solution of stochastic differential equations*. Springer, 1992, pp. 103–160.

[88] Ryota Kobayashi and Renaud Lambiotte. „Tideh: Time-Dependent Hawkes Process for Predicting Retweet Dynamics“. In: *Tenth International AAAI Conference on Web and Social Media*. 2016.

[89] Ravin Kumar, Colin Carroll, Ari Hartikainen, and Osvaldo Antonio Martin. „ArviZ a Unified Library for Exploratory Analysis of Bayesian Models in Python“. In: (2019).

[90] Matthias Kümmerer and Matthias Bethge. „State-of-the-Art in Human Scanpath Prediction“. In: *arXiv Preprint arXiv:2102.12239* (2021).

[91] Matthias Kümmerer, Theis Wallis, and Matthias Bethge. „Deep Gaze I: Boosting Saliency Prediction With Feature Maps Trained on Imagenet“. In: *Preprint arXiv:1411.1045* (2014).

[92] Matthias Kümmerer, Theis Wallis, and Matthias Bethge. „Deep Gaze II: Reading Fixations From Deep Features Trained on Object Recognition“. In: *Preprint arXiv:1610.01563* (2016).

[93] Matthias Kümmerer, Thomas S. A. Wallis, and Matthias Bethge. „Saliency Benchmarking Made Easy: Separating Models, Maps and Metrics“. In: *Computer Vision – ECCV 2018*. Ed. by Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss. Lecture Notes in Computer Science. Springer International Publishing, 2018, pp. 798–814.

[94] Matthias Kümmerer, Thomas SA Wallis, and Matthias Bethge. „DeepGaze III: Using Deep Learning to Probe Interactions Between Scene Content and Scanpath History in Fixation Selection“. In: *2019 Conference on Cognitive Computational Neuroscience, 13-16 September 2019, Berlin, Germany*. 2019.

[95] Matthias Kümmerer, Thomas SA Wallis, and Matthias Bethge. „Information-Theoretic Model Comparison Unifies Saliency Metrics“. In: *Proceedings of the National Academy of Sciences* 112.52 (2015), pp. 16054–16059.

[96] Eric Laloy and Jasper A Vrugt. „High-Dimensional Posterior Exploration of Hydrologic Models Using Multiple-Try DREAM (ZS) and High-Performance Computing“. In: *Water Resources Research* 48.1 (2012).

[97] Shiwei Lan, Vasileios Stathopoulos, Babak Shahbaba, and Mark Girolami. „Markov Chain Monte Carlo From Lagrangian Dynamics“. In: *Journal of Computational and Graphical Statistics* 24.2 (2015), pp. 357–378.

[98] Jeremy Large. „Measuring the Resiliency of an Electronic Limit Order Book“. In: *Journal of Financial Markets* 10.1 (2007), pp. 1–25.

[99] Jochen Laubrock, Anke Cajar, and Ralf Engbert. „Control of Fixation Duration During Scene Viewing by Interaction of Foveal and Peripheral Processing“. In: *Journal of Vision* 13.12 (2013), pp. 11–11.

[100] Olivier Le Meur and Zhi Liu. „Saccadic Model of Eye Movements for Free-Viewing Condition“. In: *Vision Research* 116 (2015), pp. 152–164.

[101] Benedict Leimkuhler and Sebastian Reich. *Simulating Hamiltonian Dynamics*. 14. Cambridge university press, 2004.

[102] Paul Lévy. „Theory of a Brownian Motion With a Drift“. In: *Comptes Rendus De L'Académie Des Sciences* 238.8 (1954), pp. 1313–1316.

[103] Peter A W Lewis and Gerald S Shedler. „Simulation of Nonhomogeneous Poisson Processes by Thinning“. In: *Naval Research Logistics Quarterly* 26.3 (1979), pp. 403–413.

[104] Scott Linderman. „PyPólyaGamma“. In: GitHub, 2017.

[105] Scott Linderman, Ryan P Adams, and Jonathan W Pillow. „Bayesian Latent Structure Discovery From Multi-Neuron Recordings“. In: *Advances in Neural Information Processing Systems* 29 (2016).

[106] Scott Linderman, Matthew J Johnson, and Ryan P Adams. „Dependent Multinomial Models Made Easy: Stick-Breaking With the Pólya-Gamma Augmentation“. In: *Advances in Neural Information Processing Systems* 28 (2015).

[107] Haitao Liu, Yew-Soon Ong, Xiaobo Shen, and Jianfei Cai. „When Gaussian Process Meets Big Data: A Review of Scalable GPs“. In: *IEEE Transactions on Neural Networks and Learning Systems* 31.11 (2020), pp. 4405–4423.

[108] Huiying Liu, Dong Xu, Qingming Huang, Wen Li, Min Xu, and Stephen Lin. „Semantically-Based Human Scanpath Estimation With HMMs“. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2013, pp. 3232–3239.

[109] Jun S Liu, Faming Liang, and Wing Hung Wong. „The Multiple-Try Method and Local Optimization in Metropolis Sampling“. In: *Journal of the American Statistical Association* 95.449 (2000), pp. 121–134.

[110] Chris M. Lloyd, Tom Gunter, Michael A. Osborne, and Stephen J. Roberts. „Variational Inference for Gaussian Process Modulated Poisson Processes.“ In: *International Conference on Machine Learning*. Ed. by Francis R Bach and David M Blei. Vol. 37. JMLR Workshop and Conference Proceedings. Lille, France: JMLR.org, 2015, pp. 1814–1822.

[111] Michal Lukasik, PK Srijith, Duy Vu, Kalina Bontcheva, Arkaitz Zubiaga, and Trevor Cohn. „Hawkes Processes for Continuous Time Sequence Classification: An Application to Rumour Stance Classification in Twitter“. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 2016, pp. 393–398.

[112] Steven G Luke, Tim J Smith, Joseph Schmidt, and John M Henderson. „Dissociating Temporal Inhibition of Return and Saccadic Momentum Across Multiple Eye-Movement Tasks“. In: *Journal of Vision* 14.14 (2014), pp. 9–9.

[113] David JC MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge university press, 2003.

[114] Arianna Maffei, Sacha B Nelson, and Gina G Turrigiano. „Selective Reconfiguration of Layer 4 Visual Cortical Circuitry by Visual Deprivation“. In: *Nature Neuroscience* 7.12 (2004), pp. 1353–1359.

[115] Noa Malem-Shinitski, César Ojeda, and Manfred Opper. „Variational Bayesian Inference for Nonlinear Hawkes Process With Gaussian Process Self-Effects“. In: *Entropy* 24.3 (2022), p. 356.

[116] Noa Malem-Shinitski, Manfred Opper, Sebastian Reich, Lisa Schwetlick, Stefan A Seelig, and Ralf Engbert. „A Mathematical Model of Local and Global Attention in Natural Scene Viewing“. In: *PLOS Computational Biology* 16.12 (2020), e1007880.

[117] Luca Martino, Hanxue Yang, David Luengo, Juho Kanniainen, and Jukka Corander. „A Fast Universal Self-Tuned Sampler Within Gibbs Sampling“. In: *Digital Signal Processing* 47 (2015), pp. 68–83.

[118] Clare A McGrory and Donald M Titterington. „Variational Bayesian Analysis for Hidden Markov Models“. In: *Australian & New Zealand Journal of Statistics* 51.2 (2009), pp. 227–244.

[119] Arman Melkumyan and Fabio Tozeto Ramos. „A Sparse Covariance Function for Exact Gaussian Process Inference in Large Datasets“. In: *Twenty-first international joint conference on artificial intelligence*. 2009.

[120] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. „Equation of State Calculations by Fast Computing Machines“. In: *The Journal of Chemical Physics* 21.6 (1953), pp. 1087–1092.

[121] Nicholas Metropolis and Stanislaw Ulam. „The Monte Carlo Method“. In: *Journal of the American Statistical Association* 44.247 (1949), pp. 335–341.

[122] Rupert G Miller Jr. *Survival Analysis*. John Wiley & Sons, 2011.

[123] Thomas P Minka. „Expectation Propagation for Approximate Bayesian Inference“. In: *arXiv Preprint arXiv:1301.2294* (2013).

[124] George Mohler. „Modeling and Estimation of Multi-Source Clustering in Crime and Security Data“. In: *The Annals of Applied Statistics* (2013), pp. 1525–1539.

[125] George O Mohler, Martin B Short, P Jeffrey Brantingham, Frederic Paik Schoenberg, and George E Tita. „Self-Exciting Point Process Modeling of Crime“. In: *Journal of the American Statistical Association* 106.493 (2011), pp. 100–108.

[126] Christian Molkenthin, Christian Donner, Sebastian Reich, Gert Zöller, Sebastian Hainzl, Matthias Holschneider, and Manfred Opper. „GP-ETAS: Semiparametric Bayesian Inference for the Spatio-Temporal Epidemic Type Aftershock Sequence Model“. In: *Statistics and Computing* 32.2 (2022), pp. 1–25.

[127] Jesper Moller and Rasmus Plenge Waagepetersen. *Statistical Inference and Simulation for Spatial Point Processes*. CRC press, 2003.

[128] Jesper Möller, Anne Randi Syversveen, and Rasmus Plenge Waagepetersen. „Log Gaussian Cox Processes“. In: *Scandinavian Journal of Statistics* 25.3 (1998), pp. 451–482.

[129] Kevin Murphy and Stuart Russell. „Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks“. In: *Sequential Monte Carlo methods in practice*. Springer, 2001, pp. 499–515.

[130] Fabio Musmeci and David Vere-Jones. „A Space-Time Clustering Model for Historical Earthquakes“. In: *Annals of the Institute of Statistical Mathematics* 44.1 (1992), pp. 1–11.

[131] Radford M Neal. „McMc Using Hamiltonian Dynamics“. In: *Handbook of Markov Chain Monte Carlo* 2.11 (2011), p. 2.

[132] Radford M Neal. „Slice Sampling“. In: *The Annals of Statistics* 31.3 (2003), pp. 705–767.

[133] Yurii Nesterov. „Primal-Dual Subgradient Methods for Convex Problems“. In: *Mathematical Programming* 120.1 (2009), pp. 221–259.

[134] Antje Nuthmann, Tim J Smith, Ralf Engbert, and John M Henderson. „CRISP: A Computational Model of Fixation Durations in Scene Viewing.“ In: *Psychological Review* 117.2 (2010), p. 382.

[135] Yosihiko Ogata. „On Lewis’ Simulation Method for Point Processes“. In: *IEEE Transactions on Information Theory* 27.1 (1981), pp. 23–31.

[136] Yosihiko Ogata. „Statistical Models for Earthquake Occurrences and Residual Analysis for Point Processes“. In: *Journal of the American Statistical Association* 83.401 (1988), pp. 9–27.

[137] Yosihiko Ogata and Jiancang Zhuang. „Space–Time ETAS Models and an Improved Extension“. In: *Tectonophysics* 413.1-2 (2006), pp. 13–23.

[138] Jack Olinde and Martin B Short. „A Self-Limiting Hawkes Process: Interpretation, Estimation, and Use in Crime Modeling“. In: *2020 IEEE International Conference on Big Data (Big Data)*. IEEE. 2020, pp. 3212–3219.

[139] Omiros Papaspiliopoulos and Gareth Roberts. „Non-Centered Parameterisations for Hierarchical Models and Data Augmentation“. In: *Bayesian Statistics* 7 (Jan. 2003), pp. 307–326.

[140] Antti Penttinen and Anna-Kaisa Ylitalo. „Deducing Self-Interaction in Eye Movement Data Using Sequential Spatial Point Processes“. In: *Spatial Statistics* 17 (2016), pp. 1–21.

[141] Robert J Peters, Asha Iyer, Laurent Itti, and Christof Koch. „Components of Bottom-Up Gaze Allocation in Natural Images“. In: *Vision Research* 45.18 (2005), pp. 2397–2416.

[142] Nicholas G Polson, James G Scott, and Jesse Windle. „Bayesian Inference for Logistic Models Using Pólya–Gamma Latent Variables“. In: *Journal of the American Statistical Association* 108.504 (2013), pp. 1339–1349.

[143] Carl Edward Rasmussen and Christopher K I Williams. *Gaussian Processes for Machine Learning*. Adaptive computation and machine learning. MIT Press, 2006.

[144] Jakob G Rasmussen. „Bayesian Inference for Hawkes Processes“. In: *Methodology and Computing in Applied Probability* 15.3 (2013), pp. 623–642.

[145] Alex Reinhart. „A Review of Self-Exciting Spatio-Temporal Point Processes and Their Applications“. In: *Statistical Science* 33.3 (2018), pp. 299–318.

[146] Lewis F Richardson. „Atmospheric Diffusion Shown on a Distance-Neighbour Graph“. In: *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 110.756 (1926), pp. 709–737.

[147] Nicolas Riche, Matthieu Duvinage, Matei Mancas, Bernard Gosselin, and Thierry Dutoit. „Saliency and Human Fixations: State-of-the-Art and Study of Comparison Metrics“. In: *Proceedings of the IEEE international conference on computer vision*. 2013, pp. 1153–1160.

[148] Manfred Ritter. „Evidence for Visual Persistence During Saccadic Eye Movements“. In: *Psychological Research* 39.1 (1976), pp. 67–85.

[149] Herbert Robbins and Sutton Monro. „A Stochastic Approximation Method“. In: *The Annals of Mathematical Statistics* (1951), pp. 400–407.

[150] Murray Rosenblatt. „Remarks on Some Nonparametric Estimates of a Density Function“. In: *The Annals of Mathematical Statistics* (1956), pp. 832–837.

[151] Jürgen Schmidhuber. „Deep Learning in Neural Networks: An Overview“. In: *Neural Networks* 61 (2015), pp. 85–117.

[152] Heiko H Schütt, Lars OM Rothkegel, Hans A Trukenbrod, Sebastian Reich, Felix A Wichmann, and Ralf Engbert. „Likelihood-Based Parameter Estimation and Comparison of Dynamical Cognitive Models.“ In: *Psychological Review* 124.4 (2017), p. 505.

[153] Heiko H Schütt, Lars OM Rothkegel, Hans A Trukenbrod, Sebastian Reich, Felix A Wichmann, and Ralf Engbert. „Likelihood-Based Parameter Estimation and Comparison of Dynamical Cognitive Models.“ In: *Psychological Review* 124.4 (2017), p. 505.

[154] Lisa Schwetlick, Daniel Backhaus, and Ralf Engbert. „A Dynamical Scan-Path Model for Task-Dependence During Scene Viewing.“ In: *Psychological Review* (2022).

[155] Lisa Schwetlick, Lars Oliver Martin Rothkegel, Hans Arne Trukenbrod, and Ralf Engbert. „Modeling the Effects of Perisaccadic Attention on Gaze Statistics During Scene Viewing“. In: *Communications Biology* 3.1 (2020), pp. 1–11.

[156] Emilio Segrè. „From X-Rays to Quarks: Modern Physicists and Their Discoveries“. In: Courier Corporation, 2012. Chap. 10.

[157] Xuan Shao, Ye Luo, Dandan Zhu, Shuqin Li, Laurent Itti, and Jianwei Lu. „Scanpath Prediction Based on High-Level Features and Memory Bias“. In: *International Conference on Neural Information Processing*. Springer. 2017, pp. 3–13.

[158] Erin M Shockley, Jasper A Vrugt, and Carlos F Lopez. „PyDREAM: High-Dimensional Parameter Inference for Biological Models in Python“. In: *Bioinformatics* 34.4 (2018), pp. 695–697.

[159] Adrian FM Smith and Gareth O Roberts. „Bayesian Computation via the Gibbs Sampler and Related Markov Chain Monte Carlo Methods“. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 55.1 (1993), pp. 3–23.

[160] T Caitlin Smith and Craig E Jahr. „Self-Inhibition of Olfactory Bulb Neurons“. In: *Nature Neuroscience* 5.8 (2002), pp. 760–766.

[161] PK Srijith, Michal Lukasik, Kalina Bontcheva, and Trevor Cohn. „Longitudinal Modeling of Social Media With Hawkes Process Based on Users and Networks“. In: *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*. 2017, pp. 195–202.

[162] Gabriele Stabile and Giovanni Luca Torrisi. „Risk Processes With Non-Stationary Hawkes Claims Arrivals“. In: *Methodology and Computing in Applied Probability* 12.3 (2010), pp. 415–429.

[163] Rainer Storn and Kenneth Price. „Differential Evolution–a Simple and Efficient Heuristic for Global Optimization Over Continuous Spaces“. In: *Journal of Global Optimization* 11.4 (1997), pp. 341–359.

[164] Deborah Sulem, Vincent Rivoirard, and Judith Rousseau. „Bayesian Estimation of Nonlinear Hawkes Process“. In: *arXiv Preprint arXiv:2103.17164* (2021).

[165] Matthew A Taddy and Athanasios Kottas. „Mixture Modeling for Marked Poisson Processes“. In: *Bayesian Analysis* 7.2 (2012), pp. 335–362.

[166] Martin A Tanner and Wing Hung Wong. „The Calculation of Posterior Distributions by Data Augmentation“. In: *Journal of the American Statistical Association* 82.398 (1987), pp. 528–540.

[167] Benjamin W Tatler, Roland J Baddeley, and Benjamin T Vincent. „The Long and the Short of It: Spatial Statistics at Fixation Vary With Saccade Amplitude and Task“. In: *Vision Research* 46.12 (2006), pp. 1857–1862.

[168] Benjamin W Tatler, James R Brockmole, and Roger HS Carpenter. „LATEST: A Model of Saccadic Decisions in Space and Time.“ In: *Psychological Review* 124.3 (2017), 267–?

[169] Benjamin W Tatler, James R Brockmole, and Roger HS Carpenter. „LATEST: A Model of Saccadic Decisions in Space and Time.“ In: *Psychological Review* 124.3 (2017), p. 267.

[170] Benjamin W Tatler and Benjamin T Vincent. „Systematic Tendencies in Scene Viewing“. In: *Journal of Eye Movement Research* 13.12 (2008), pp. 1–18.

[171] Hamed Rezazadegan Tavakoli, Esa Rahtu, and Janne Heikkilä. „Stochastic Bottom–Up Fixation Prediction and Saccade Generation“. In: *Image and Vision Computing* 31.9 (2013), pp. 686–693.

[172] Cajo JF Ter Braak. „A Markov Chain Monte Carlo Version of the Genetic Algorithm Differential Evolution: Easy Bayesian Computing for Real Parameter Spaces“. In: *Statistics and Computing* 16.3 (2006), pp. 239–249.

[173] Luke Tierney. „Markov Chains for Exploring Posterior Distributions“. In: *The Annals of Statistics* (1994), pp. 1701–1728.

[174] Michalis K. Titsias. „Variational Learning of Inducing Variables in Sparse Gaussian Processes.“ In: *Artificial Intelligence and Statistics*. Ed. by David A. Van Dyk and Max Welling. Vol. 5. JMLR Proceedings. Clearwater Beach, FL, USA: JMLR.org, 2009, pp. 567–574.

[175] Wilson Truccolo. „From Point Process Observations to Collective Neural Dynamics: Nonlinear Hawkes Process GLMs, Low-Dimensional Dynamics and Coarse Graining“. In: *Journal of Physiology-Paris* 110.4 (2016), pp. 336–347.

[176] Pieter JA Unema, Sebastian Pannasch, Markus Joos, and Boris M Velichkovsky. „Time Course of Information Processing During Scene Perception: The Relationship Between Saccade Amplitude and Fixation Duration“. In: *Visual Cognition* 12.3 (2005), pp. 473–494.

[177] Aldo V Vecchia. „Estimation and Model Identification for Continuous Spatial Processes“. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 50.2 (1988), pp. 297–312.

[178] Aki Vehtari, Andrew Gelman, Daniel Simpson, Bob Carpenter, and Paul-Christian Bürkner. „Rank-Normalization, Folding, and Localization: An Improved $\hat{R}$ for Assessing Convergence of MCMC (with Discussion)“. In: *Bayesian analysis* 16.2 (2021), pp. 667–718.

[179] David Vere-Jones and Tomoaki Ozaki. „Some Examples of Statistical Estimation Applied to Earthquake Data“. In: *Annals of the Institute of Statistical Mathematics* 34.1 (1982), pp. 189–207.

[180] Jasper A Vrugt, CJF Ter Braak, CGH Diks, Bruce A Robinson, James M Hyman, and Dave Higdon. „Accelerating Markov Chain Monte Carlo Simulation by Differential Evolution With Self-Adaptive Randomized Subspace Sampling“. In: *International Journal of Nonlinear Sciences and Numerical Simulation* 10.3 (2009), pp. 273–290.

[181] Wei Wang, Cheng Chen, Yizhou Wang, Tingting Jiang, Fang Fang, and Yuan Yao. „Simulating Human Saccadic Scanpaths on Natural Images“. In: *CVPR 2011*. IEEE. 2011, pp. 441–448.

[182] Niklas Wilming, Simon Harst, Nico Schmidt, and Peter König. „Saccadic Momentum and Facilitation of Return Saccades Contribute to an Optimal Foraging Strategy“. In: *PLoS Computational Biology* 9.1 (2013), e1002871.

[183] Robert L Wolpert and Katja Ickstadt. „Poisson/Gamma Random Field Models for Spatial Statistics“. In: *Biometrika* 85.2 (1998), pp. 251–267.

[184] Baichuan Yuan, Hao Li, Andrea L Bertozzi, P Jeffrey Brantingham, and Mason A Porter. „Multivariate Spatiotemporal Hawkes Processes and Network Reconstruction“. In: *SIAM Journal on Mathematics of Data Science* 1.2 (2019), pp. 356–382.

[185] Dario Zanca, Marco Gori, Stefano Melacci, and Alessandra Rufa. „Gravitational Models Explain Shifts on Human Visual Attention“. In: *Scientific Reports* 10.1 (2020), pp. 1–9.

[186] Dario Zanca, Stefano Melacci, and Marco Gori. „Gravitational Laws of Focus of Attention“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42.12 (2019), pp. 2983–2995.

[187] Gregory J Zelinsky. „A Theory of Eye Movements During Target Acquisition.“ In: *Psychological Review* 115.4 (2008), 787–?

[188] Rui Zhang, Christian Walder, Marian-Andrei Rizoiu, and Lexing Xie. „Efficient Non-Parametric Bayesian Hawkes Processes“. In: *arXiv Preprint arXiv:1810.03730* (2018).

[189] Rui Zhang, Christian J Walder, and Marian-Andrei Rizoiu. „Variational Inference for Sparse Gaussian Process Modulated Hawkes Process.“ In: *Proceedings of the AAAI Conference on Artificial Intelligence*. New York, NY, USA: AAAI Press, 2020, pp. 6803–6810.

[190] Feng Zhou, Zhidong Li, Xuhui Fan, Yang Wang, Arcot Sowmya, and Fang Chen. „Efficient EM-Variational Inference for Hawkes Process“. In: *arXiv Preprint arXiv:1905.12251* (2019).

[191] Feng Zhou, Zhidong Li, Xuhui Fan, Yang Wang, Arcot Sowmya, and Fang Chen. „Efficient Inference for Nonparametric Hawkes Processes Using Auxiliary Latent Variables“. In: *Journal of Machine Learning Research* 21.241 (2020), pp. 1–31.

[192] Feng Zhou, Yixuan Zhang, and Jun Zhu. „Efficient Inference of Flexible Interaction in Spiking-Neuron Networks“. In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[193] Lingjiong Zhu. „Central Limit Theorem for Nonlinear Hawkes Processes“. In: *Journal of Applied Probability* 50.3 (2013), pp. 760–771.

[194] Jiancang Zhuang and Jorge Mateu. „A Semiparametric Spatiotemporal Hawkes-Type Point Process Model With Periodic Background for Crime Data“. In: *Journal of the Royal Statistical Society: Series a (Statistics in Society)* 182.3 (2019), pp. 919–942.

Appendix Related to Chapte 5

A

A.1 Samples Paths for the Exploration Exploitation Model

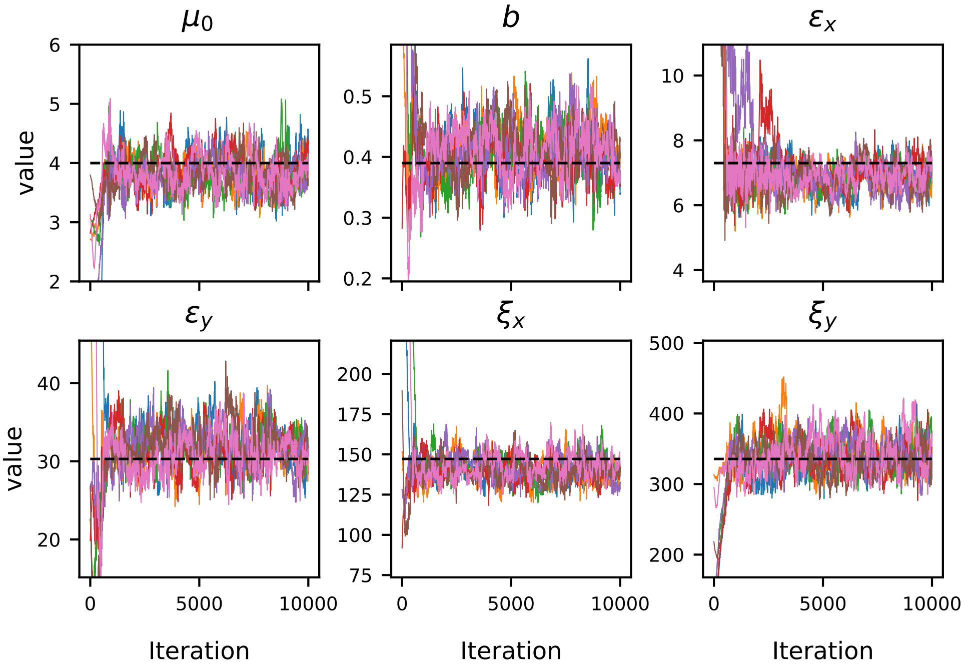


Fig. A.1.: Samples path for the ALDI sampler for the EE model. All the chains mix well and converge to the ground truth value. Autocorrelations can be seen in the sample paths of all the parameters.

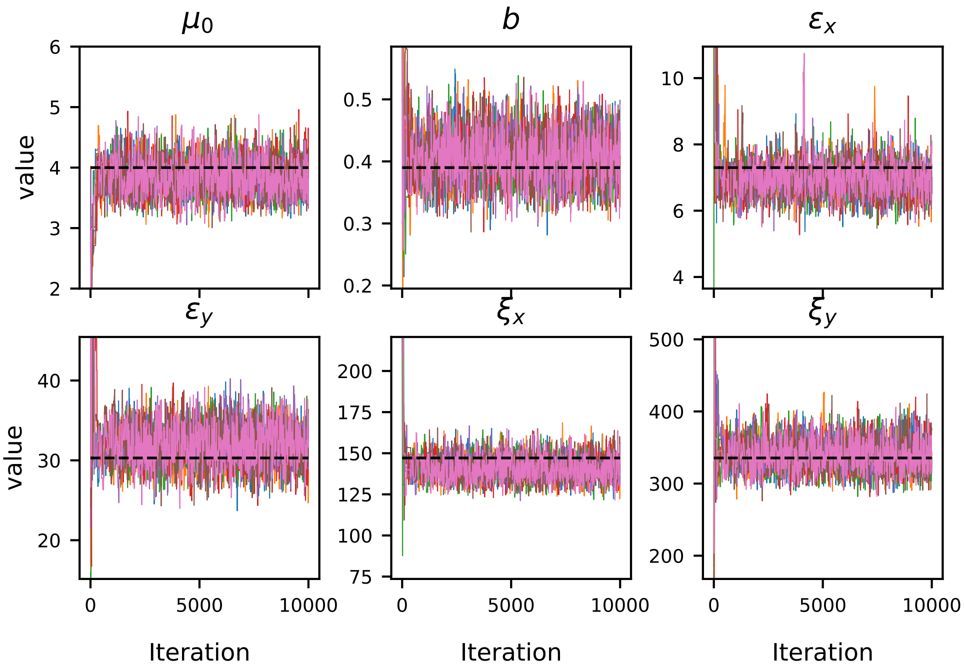


Fig. A.2.: Samples path for the DREAM sampler for the EE model. All the chains mix well and converge to the ground truth value.

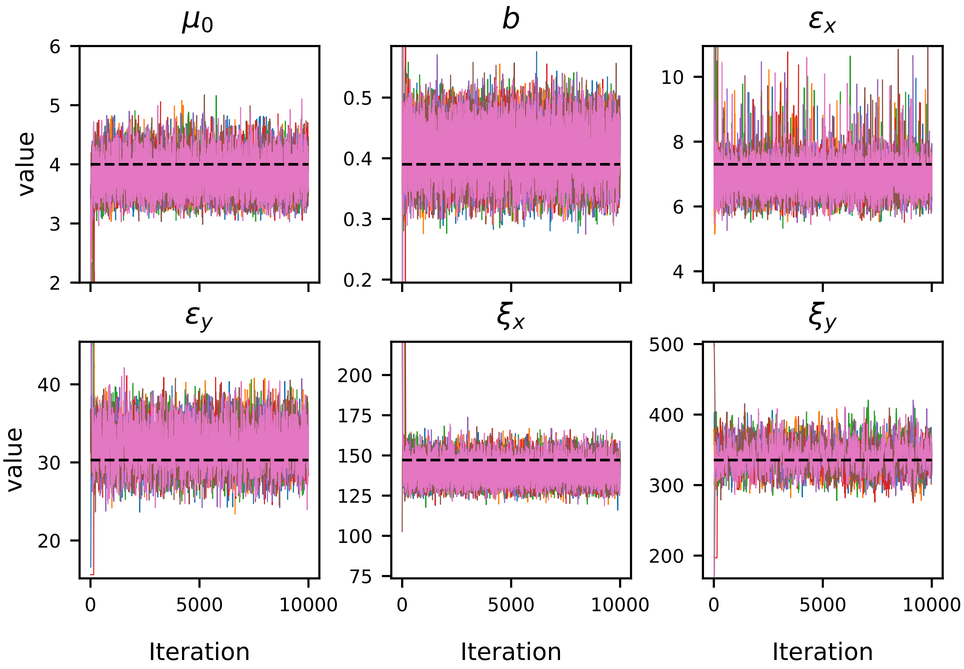


Fig. A.3.: Samples path for the HMC sampler for the EE model. All the chains mix well and converge to the ground truth value. Autocorrelations can be seen in the samples path of the parameter ξ_y.

A.2 Samples Paths for the Scenewalk Model

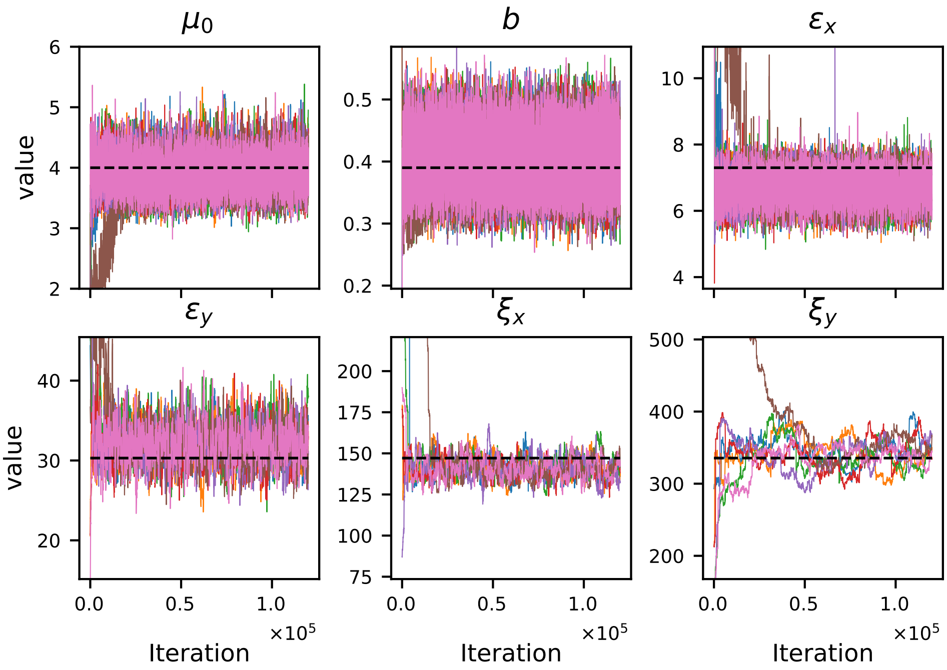


Fig. A.4.: Samples path for the NUTS sampler for the EE model. Eventually, all the chains mix and concentrate around the ground truth value. In comparison to the other samplers, the sampler requires more samples to reach stationarity. Autocorrelations can be seen in the sample paths of all the parameters.

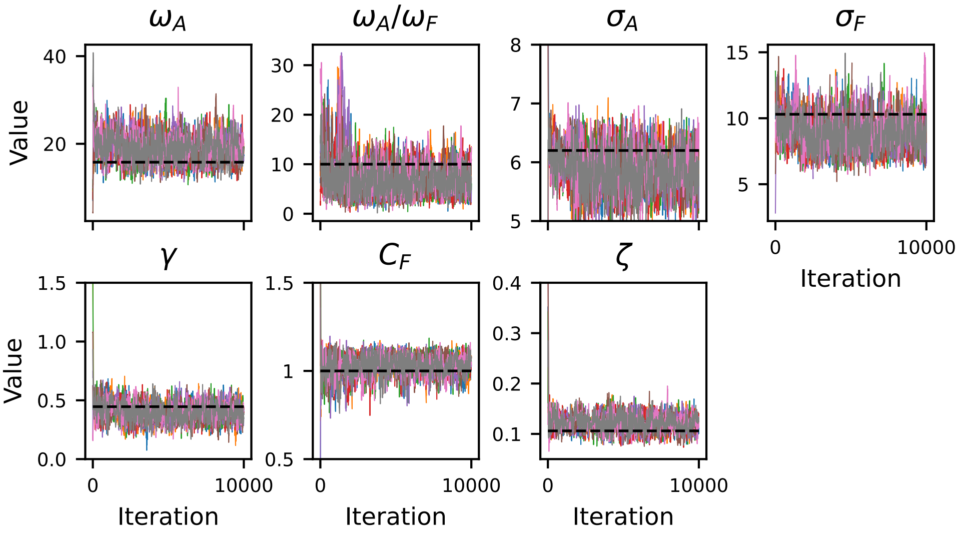


Fig. A.5.: Samples path for the ALDI sampler for the SW model. All the chains mix well and converge to the ground truth value. Autocorrelations can be seen in the sample paths of all the parameters.

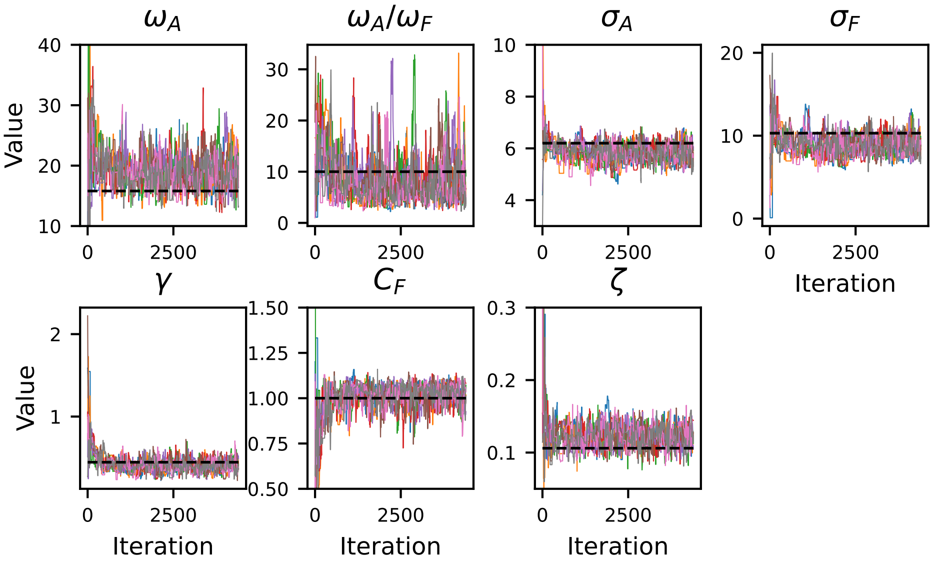


Fig. A.6.: Samples path for the DREAM sampler for the SW model. Autocorrelations can be seen in the sample paths of all the parameters.

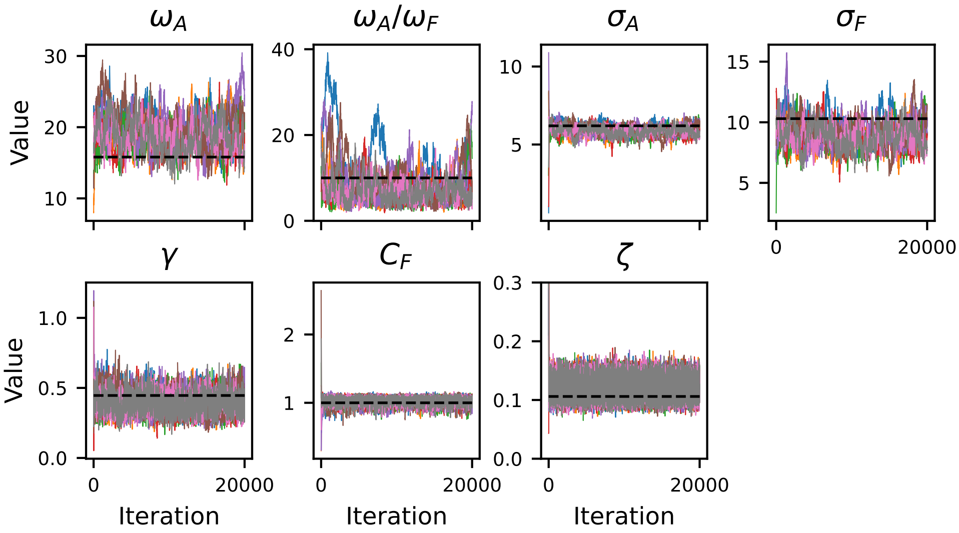


Fig. A.7.: Samples path for the HMC sampler for the SW model. All the chains mix well and converge to the ground truth value. The chains for some of the parameters reach stationarity fast and mix well (ζ, for example), whereas, for others, the chains contain noticeable autocorrelation (ω_A, for example).

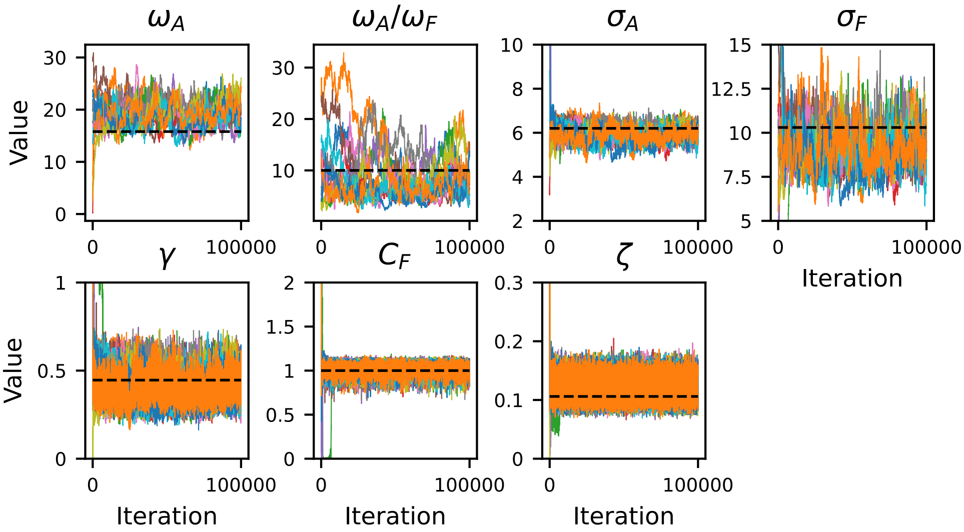


Fig. A.8.: Samples path for the NUTS sampler for the SW model. The chains for some of the parameters reach stationarity fast and mix well (ζ, for example), whereas, for others, the chains contain noticeable autocorrelation (ω_A, for example).

A.3 ESS Plots for the Exploration Exploitation Model

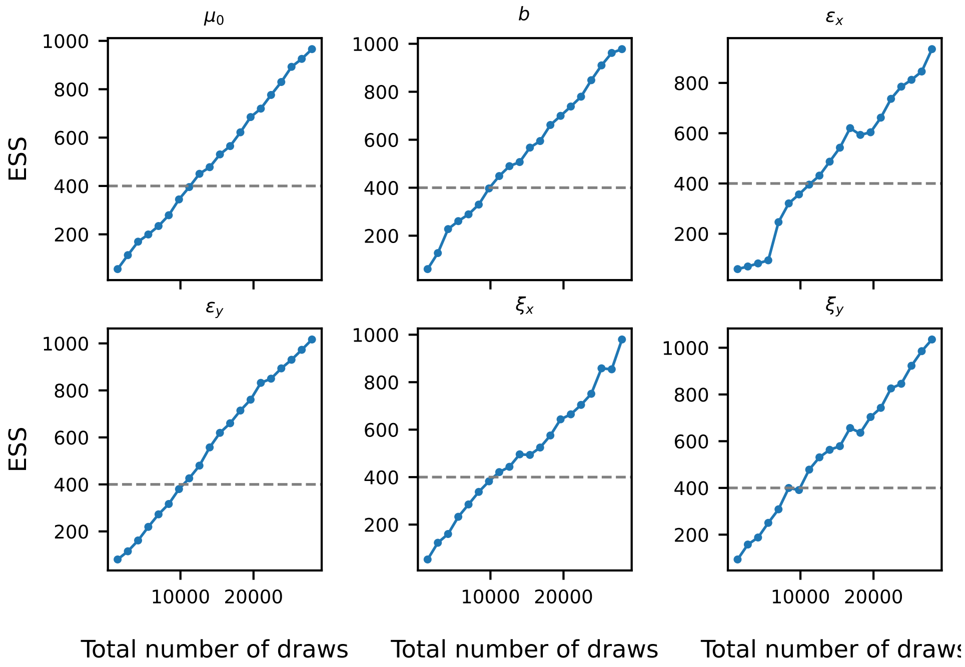


Fig. A.9.: ESS evolution for the DREAM sampler for the EE model. The horizontal dashed line indicates the goal of 400 ESS. All the parameters reach the ESS goal after a similar amount of iterations.

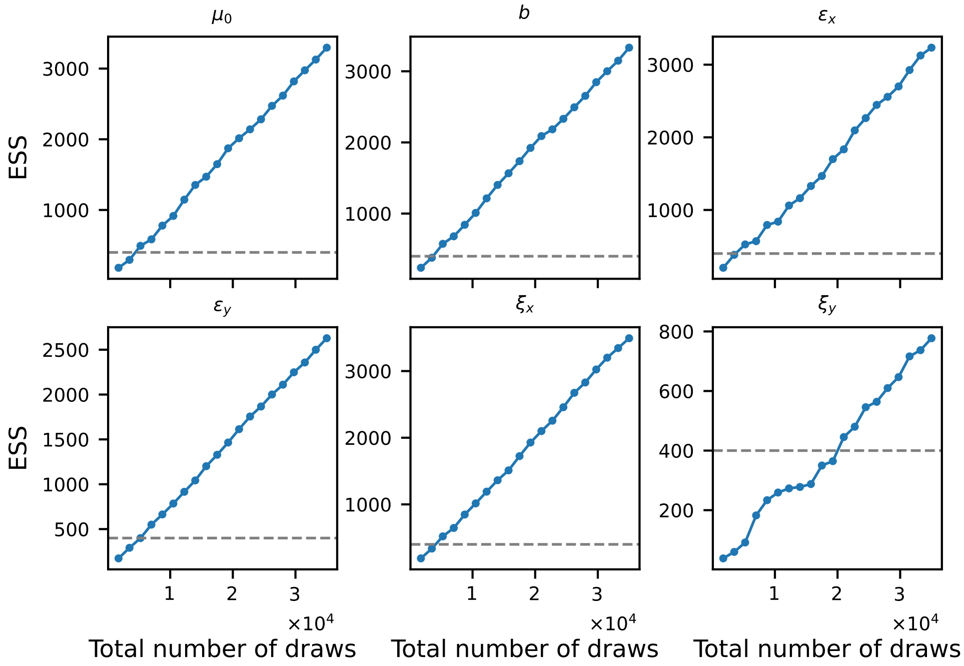


Fig. A.10.: ESS evolution for the HMC sampler for the EE model. The horizontal dashed line indicates the goal of 400 ESS. The parameter ξ_y reaches the ESS goal much later than the other parameters.

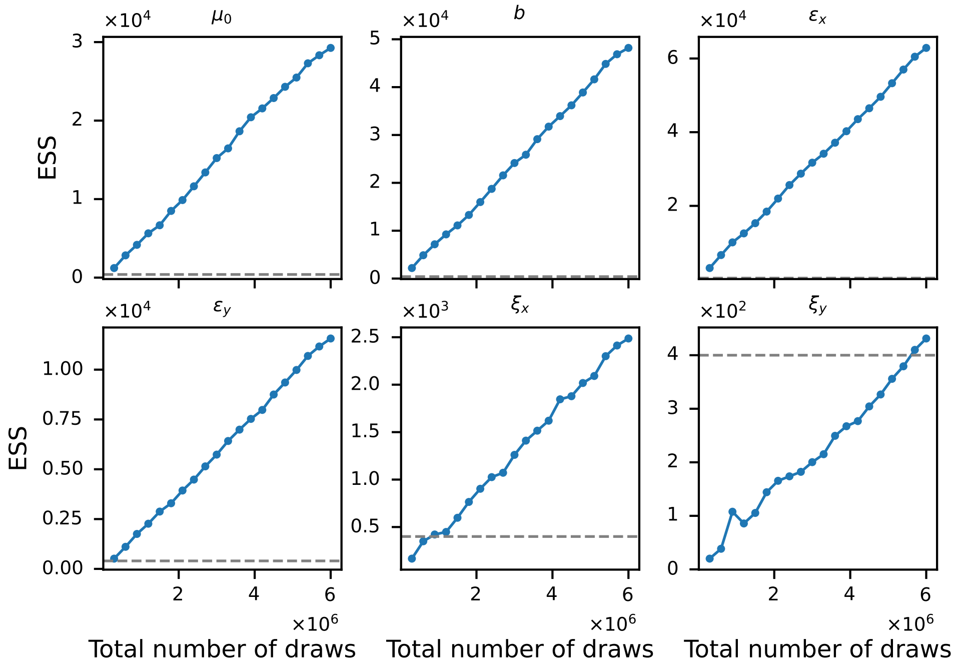


Fig. A.11.: ESS evolution for the NUTS sampler for the EE model. The horizontal dashed line indicates the goal of 400 ESS. The parameter ξ_y reaches the ESS goal much later than the other parameters.

A.4 ESS Plots for the Scenewalk Model

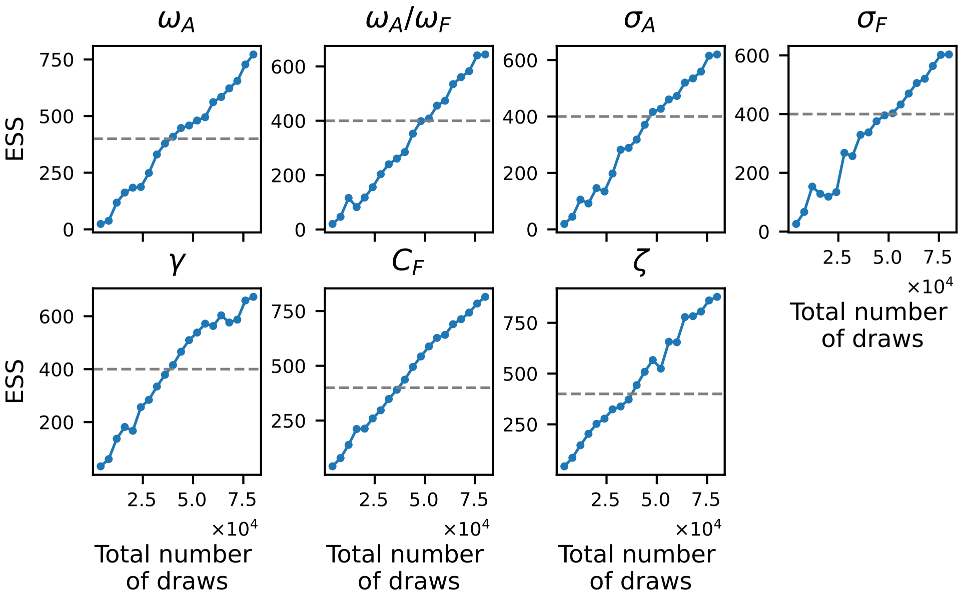


Fig. A.12.: ESS evolution for the ALDI sampler for the SW model. The horizontal dashed line indicates the goal of 400 ESS. All the parameters require a similar amount of draws to reach the goal.

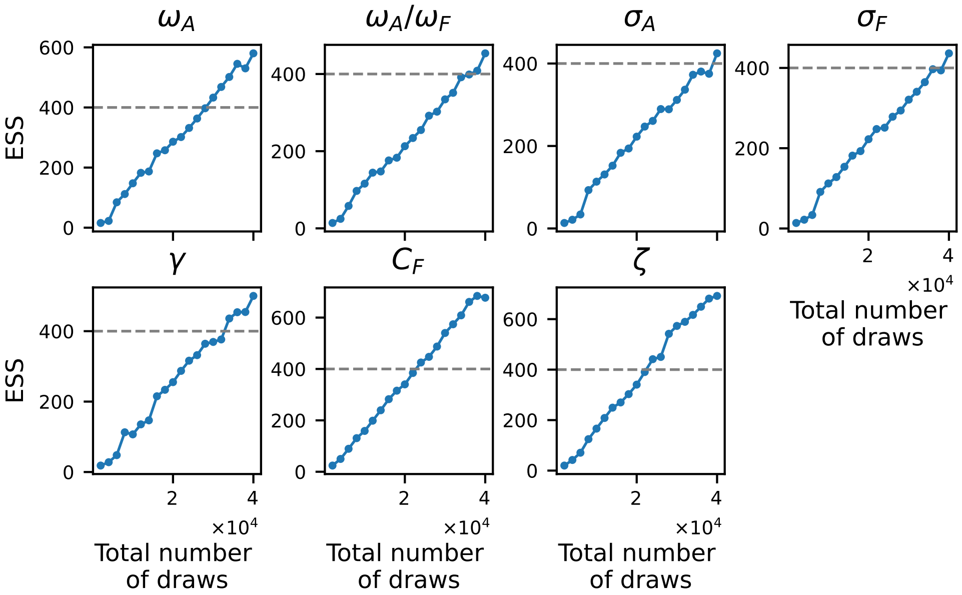


Fig. A.13.: ESS evolution for the DREAM sampler for the SW model. The horizontal dashed line indicates the goal of 400 ESS. All the parameters reach the ESS goal after a similar amount of iterations.

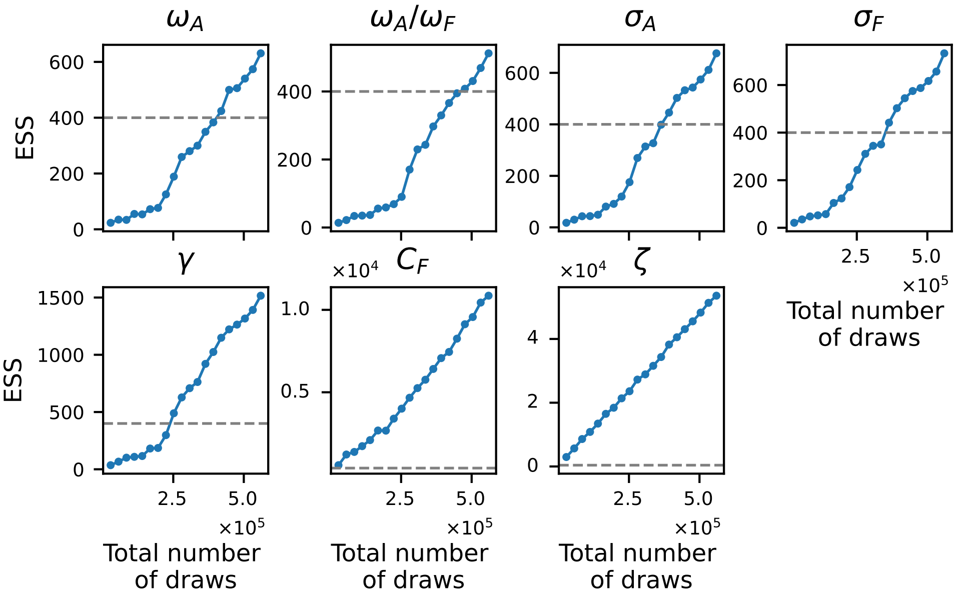


Fig. A.14.: ESS evolution for the SW sampler for the EE model. The horizontal dashed line indicates the goal of 400 ESS. Some parameters reach the goal very fast (ζ, for example), whereas others require longer simulation (ω_A, for example).

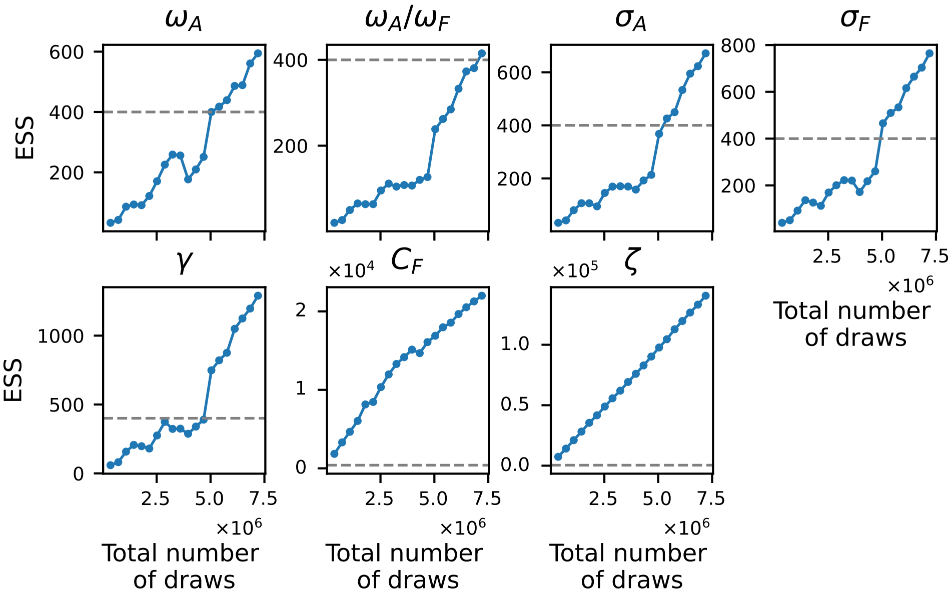


Fig. A.15.: ESS evolution for the NUTS sampler for the SW model. The horizontal dashed line indicates the goal of 400 ESS.Some parameters reach the goal very fast (ζ, for example), whereas others require longer simulation (ω_A, for example).

A.5 PSRF Plots for the Exploration Exploitation Model

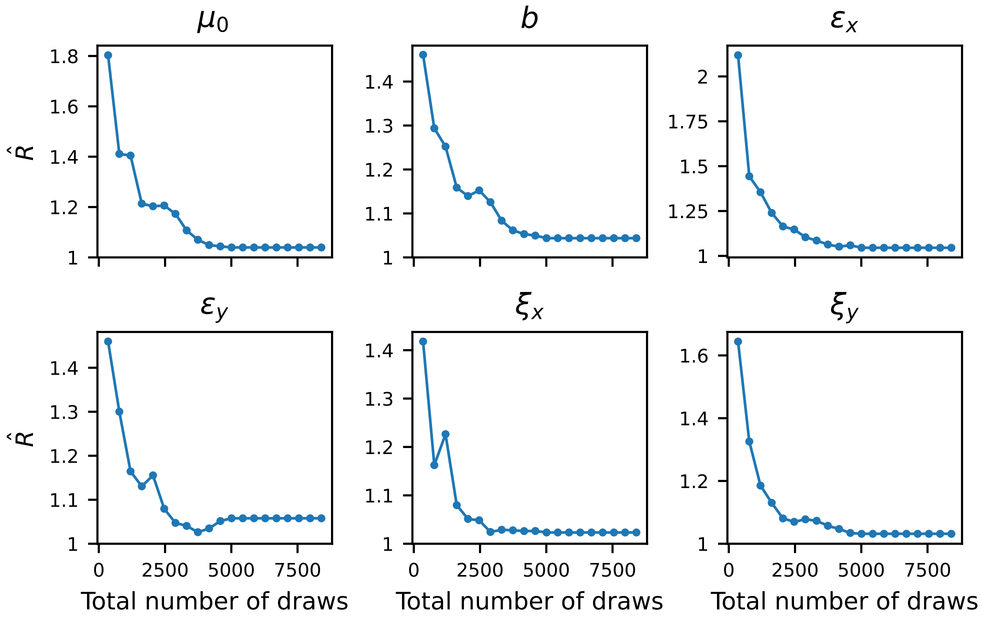


Fig. A.16.: PSRF evolution for the DREAM sampler for the EE model. All the parameters require a similar amount of draws for the $\hat{R}$ to converge to 1.

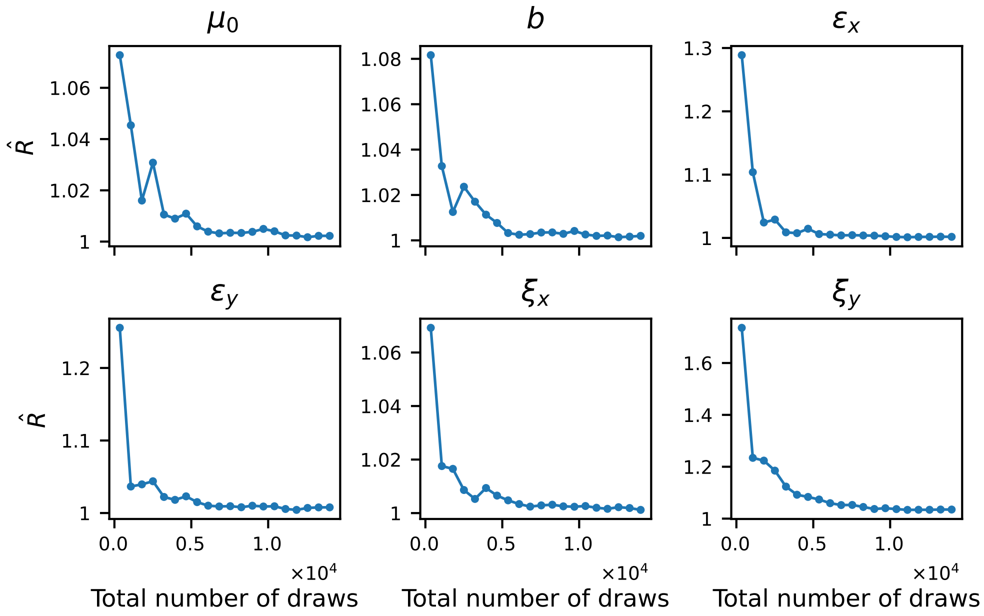


Fig. A.17.: PSRF evolution for the HMC sampler for the EE model. Some of the parameters (μ_0 and b, for example) converge to 1 faster than the others.

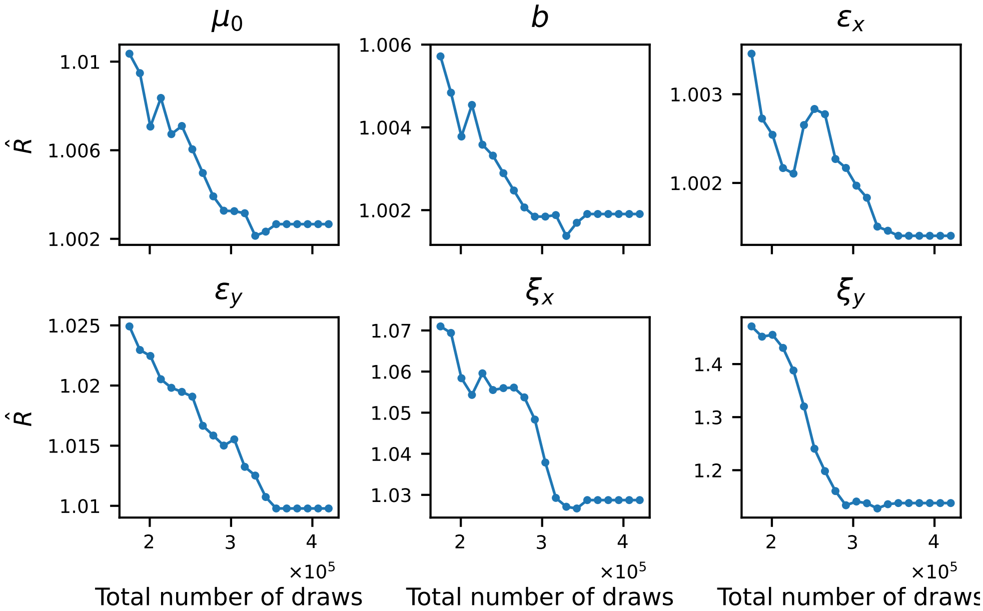


Fig. A.18.: PSRF evolution for the NUTS sampler for the EE model. Some of the parameters (μ_0 and b, for example) converge to 1 faster than the others.

A.6 PSRF Plots for the Scenewalk Model

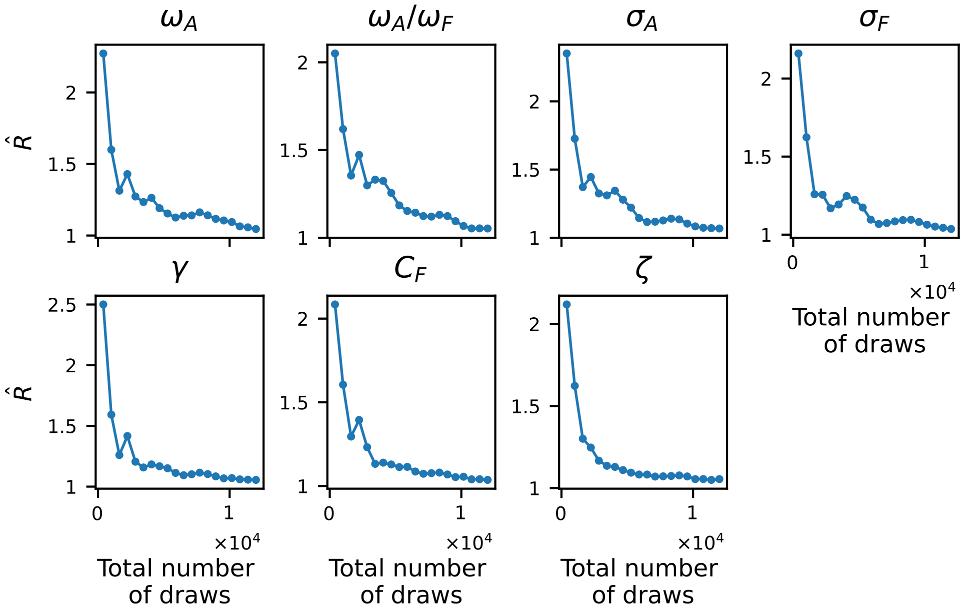


Fig. A.19.: PSRF evolution for the ALDI sampler for the SW model. All the parameters require a similar amount of draws for the $\hat{R}$ to converge to 1.

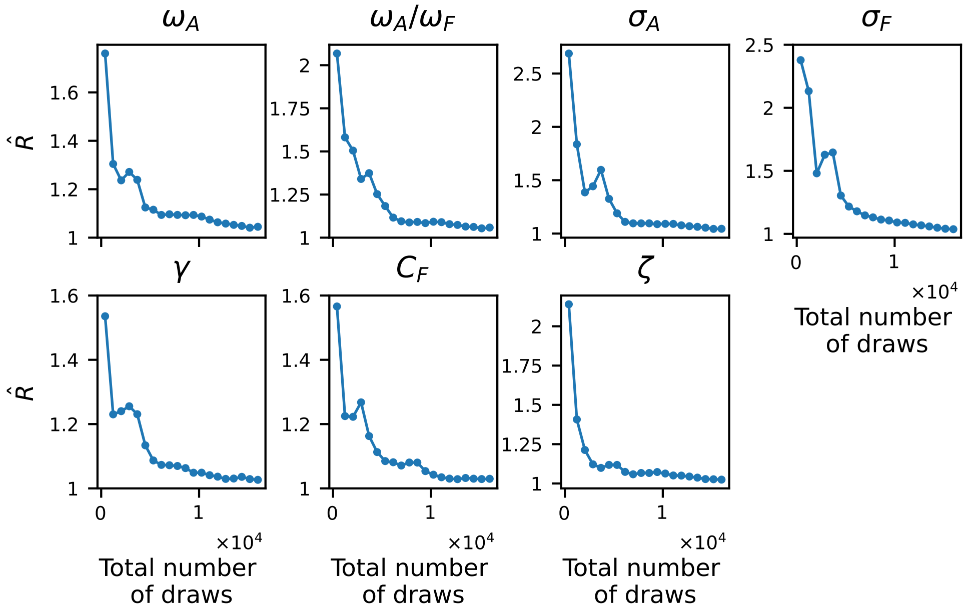


Fig. A.20.: PSRF evolution for the DREAM sampler for the SW model. All the parameters require a similar amount of draws for the $\hat{R}$ to converge to 1.

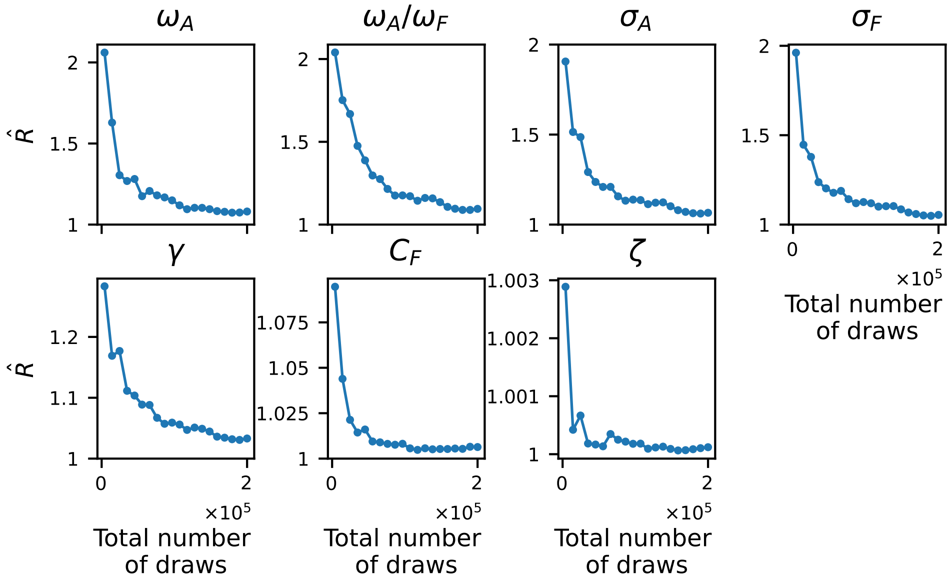


Fig. A.21.: PSRF evolution for the SW sampler for the EE model. Some of the parameters (ζ, for example) converge to 1 faster than the others.

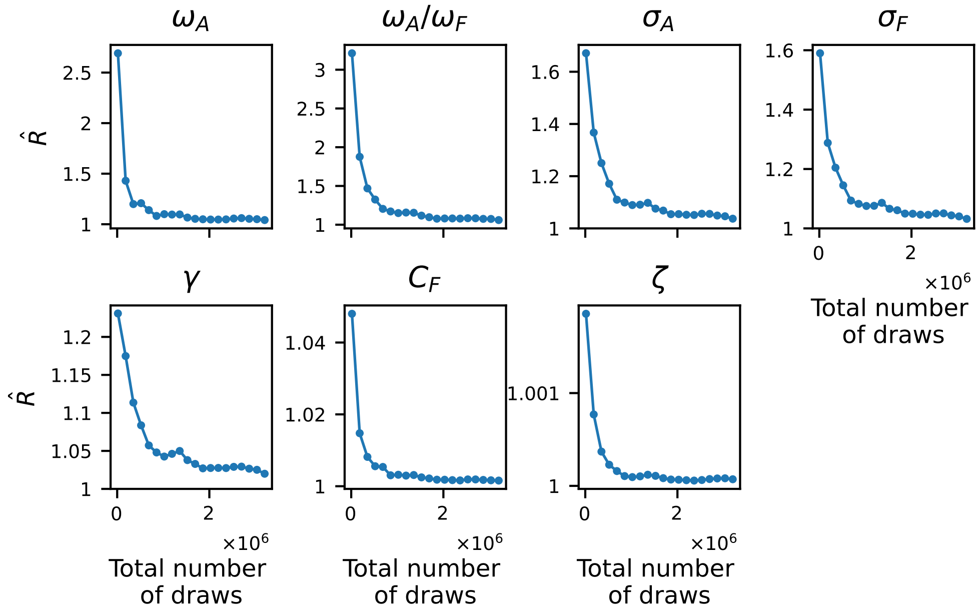


Fig. A.22.: PSRF evolution for the NUTS sampler for the SW model. Some of the parameters (ζ, for example) converge to 1 faster than the others.