UNIVERSITY OF
CAMBRIDGE

Information and generative deep learning

with applications to medical time-series

Tom Edinburgh

Clare College

This dissertation is submitted in July 2023 for the degree of
Doctor of Philosophy

Abstract

Title: Information and generative deep learning with applications to medical time-series
Author: Tom Edinburgh

Physiological time-series data are a valuable but under-utilised resource in intensive care medicine. These data are highly-structured and contain a wealth of information about the patient state, but can be very high-dimensional and difficult to interpret. Understanding temporal relationships between time-series variables is crucial for many important tasks, in particular identifying patient phenotypes within large heterogeneous cohorts, and predicting and explaining physiological changes to a patient over time. There are wide-ranging complexities involved in learning such insights from longitudinal data, including a lack of a universal accepted framework for understanding causal influence in time-series, issues with poor quality data segments that bias downstream tasks, and important privacy concerns around releasing sensitive personal data. These challenges are by no means unique to this clinical application, and there are significant domain-agnostic elements within this thesis that have a broad scope to any research area that is centred around time-series monitoring (e.g. climate science, mathematical finance, signal processing).

In the first half of this thesis, I focused firstly on information and causal influence in time-series data and then on flexible time-series modelling and hierarchical model comparison using Bayesian methods. To aid these tasks, I reviewed and developed new statistical methodology, particularly using integrated likelihoods for model evidence estimation. Together, this provided a framework for evaluating trajectories of the information contained within and between physiological variables, and allowed a comparison between patient cohorts that showed evidence of impaired physiological regulation in Covid-19 patients. The second half of this thesis introduced generative deep learning models as a tool to address some of the key difficulties in clinical time-series data, including artefact detection, imputation and synthetic dataset generation. The latter is especially important in the future of critical care research, because of the inherent challenges in publishing clinical datasets. However, I showed that that there are many obstacles that must be addressed before large-scale synthetic datasets can be utilised fully, including preserving complex relationships between physiological time-series variables within the synthetic data.

Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the Preface and specified in the text. It is not substantially the same as any that I have submitted, or am concurrently submitting, for a degree or diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. I further state that no substantial part of my dissertation has already been submitted, or is being concurrently submitted, for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text.

Tom Edinburgh
July 2023

Acknowledgements

Writing acknowledgements to my thesis felt strange, because it inevitably encouraged me to reflect on the journey I have made through my doctorate. I have been fortunate to work alongside many brilliant colleagues and friends, including but not limited to TS, DD, BS, JS, PE, PT, CW and TG. I am very grateful to my family and friends, especially all those at CC and CSLE. I'd particularly like to mention everyone I've lived with during the last few years (HS, MC, CW, HJ, M&M, FP and JH), who have consistently been wonderful company and have somehow managed to keep me sane throughout the whole thing.

My biggest thanks go to AE and SE, whose unfailing guidance and wisdom has been invaluable. I'm very grateful for your patience and for helping me to keep sight of the bigger picture whenever I started to lose momentum in all the minor details. I am also grateful for all the feedback from my examiners, SB and GC, which has undoubtedly made this thesis stronger.

Finally, I wanted to take a moment to recognise the journey I have been on in the four and half years since I started this PhD. I know I've learnt so much about myself (as well as a lot of interesting academic things) and I hope I've become a better person for the experience!

Contents

Abbreviations and acronyms

ABC Approximate Bayesian computation.

ABP Arterial blood pressure.

AIC Akiake information criterion.

Amsterdam UMC Amsterdam University Medical Centres.

AmsterdamUMCdb Amsterdam University Medical Centres database.

ARIMA Autoregressive integrated moving average.

AUC Area under curve.

BF Bayes factor.

CCIMI Cantab Capital Institute for Mathematics of Information.

CCM Convergent cross mapping.

CNN Convolutional neural network.

CNS Central nervous system.

CO Cardiac output.

CRP C-reactive protein.

CTIR Coarse-grained transinformation rate.

DDW Dutch Data Warehouse for Covid-19.

EGC Extended Granger causality.

ELBO Evidence lower bound objective.

ESICM European Society of Intensive Care Medicine.

ETE Effective transfer entropy.

FC Fully connected.

FPR False positive rate.

GAM Generalised additive model.

GAMM Generalised additive mixed model.

GAN Generative adversarial network.

GC Granger causality.

GCS Glasgow Coma Scale.

GDPR General Data Protection Regulation.

GN Gaussian noise.

GPU Graphics processing unit.

HB Bidirectional coupled Hénon maps.

HB(I) Identical bidirectional coupled Hénon maps.

HB(NI) Non-identical bidirectional coupled Hénon maps.

HDI Highest density interval.

HIPAA Health Insurance Portability and Accountability Act.

HR Heart rate.

HU Unidirectional coupled Hénon maps.

IAF Inverse autoregressive flow.

ICU Intensive Care Unit.

IQR Interquartile range.

KL-divergence Kullback-Liebler divergence.

KS Kolmogorov-Smirnov (statistic).

KSG Kraskov-Stögbauer-Grassberger (algorithm).

LOCF Last observation carried forward.

LP Linear process.

LVM Latent variable model.

MAP Mean arterial pressure.

MCMC Markov chain Monte Carlo.

MI Mutual information.

MLE Maximum likelihood estimate.

MPLE Maximum penalised likelihood estimate.

MRI Magnetic resonance imaging.

MSE Mean squared error.

NaN Not a number (undefined).

NIG Normal-inverse-gamma.

NLGC Nonlinear Granger causality.

NN Nearest neighbour.

PCA Principal component analysis.

PI Predictability improvement.

RBF Radial basis function.

RNN Recurrent neural network.

ROC Receiver operating characteristic.

ROC AUC Area under curve of receiver operating characteristic.

RSS Residual sum of squares.

RwGN Resampling with Gaussian noise.

SI Similarity index.

SMC Sequential Monte Carlo.

SOFA Sequential Organ Failure Assessment.

SVD Singular value decomposition.

T Temperature.

TE Transfer entropy.

TimeGAN Time-series generative adversarial network.

TPR True positive rate.

TSTR Train on synthetic, test on real.

UL Ulam lattice.

VAE Variational autoencoder.

WBC White blood cell.

CHAPTER 1

Introduction

1.1 Intensive care and medical time-series data

Multimodality monitoring of a vast number of physiological variables forms the basis of data-driven clinical care in the intensive care unit (ICU). ICU is a data-rich environment, perhaps more than any other healthcare setting, with the physiological patient state described by a wealth of complex and yet highly-structured data [1], including demographic information, diagnoses, laboratory and imaging results, treatments and drugs, and constantly-recorded routine vitals. This final group, containing physiological time-series data, are particularly useful, for alerting clinicians to real-time clinical deterioration, especially since many ICU patients are unable to communicate, and for deriving key clinical parameters that are closely associated with clinical outcomes, e.g. optimal cerebral autoregulation [2] and heart rate variability [3]. Additionally, new sets of derived measures based on physiological time-series variables, e.g. from signal theory [4] and causal influence [5, 6], may gain increasing relevance as clinical care experiences a shift towards personalised medicine [7].

Intensive care is also defined by heterogeneity in the patient state and disease profile, with patients sometimes experiencing multiple life-threatening conditions concurrently. Among the primary causes of admission to ICU are traumatic brain injury [8], sepsis [9], and cardiac or respiratory failure. One of the defining global events during my PhD was the Covid-19 pandemic, which placed huge burdens on clinical practice, resources and staff, and introduced a new (and initially poorly-understood) cohort of ICU patients. Heterogeneity in patient phenotypes, multiple diagnoses and treatment response creates a complicated picture for clinical care. Clinical decisions about interventions and treatment are often underpinned by extensive clinical experience, but can sometimes lack a conclusive evidence base beyond this, particularly since randomised control trials for specific interventions in intensive care have often been lacking or have been ineffective in their outcomes [1, 10]. Multi-centre longitudinal observation studies [7, 11] offer a pathway to improving

characterisation and classification of patient phenotypes, as an alternative approach to clinical trials of specific interventions.

Freely-accessible large-scale ICU databases are an incredibly rich resource for observational clinical research, enabling development new and existing methodologies for uncovering structure and association within complex multi-modal data. Releasing a large-scale intensive care database is a complicated technical challenge, requiring careful data preprocessing to preserve the privacy of individuals whose sensitive data is included in the database. Widely-used ICU databases include MIMIC-III [12], eICU [13], CCHIC [14], AmsterdamUMCdb [15] and, during the Covid-19 pandemic, the Dutch Data Warehouse for Covid-19 (DDW) [16, 17]. In this thesis, I used both AmsterdamUMCdb and DDW, which I have described more fully in Sections 2.3, 3.3 and 5.4.

Physiological time-series contain information at multiple frequencies, and multi-scale waveform metrics can help forecast clinical events [18]. In practice, a significant amount of information is lost by focusing on simple summary measures [19]. For instance, increased multifractal signal fluctuations during hypotension events suggests the presence of physiological regulatory mechanisms [4], but high-frequency arterial blood pressure waveforms are typically summarised only as diastolic, systolic and mean arterial pressures over some time interval. For many known clinically-relevant parameters, direct and minimally-invasive measurement is often not possible, or can be confounded by complex physiological dynamics. As a result, a clearer understanding of the interdependent relationships between physiological variables is crucial, especially since physiological systems rarely act in isolation to each other.

ICU patients, even those with favourable outcomes, face increased risks of devastating long-term burdens to health and socioeconomic status, both for the patient and for their immediate support network. There is an obvious need to address all parts of the clinical journey from initial medical incident and admission to outcomes on multiple timescales [10, 20]. This goal is aided by a well-defined personalised medicine approach within the ICU admission period. One key element of this involves interpretable patient representations, and their trajectories within a well-defined representation space during ICU stay.

Mathematical modelling, information theory and deep learning will play an important role in the future of clinical care, complementing existing clinical knowledge and experience in order to better understand individual patient states and to identify optimal personalised treatment strategies. Integrated successfully into clinical care, these approaches can help to reduce uncertainty for clinicians and for patients alike. This thesis explores the application of ideas from these mathematical and statistical areas to medical time-series data from ICU. This includes domain-agnostic advances in time-series modelling methodology, which can be applied to a much wider range of scientific disciplines both inside and outside of clinical medicine.

1.2 Thesis outline

This thesis covers a range of methodology for understanding and modelling time-series, in particular with applications to medical time-series data from intensive care. The structure of this thesis is as follows:

- In **Chapter 2**, I reviewed causal influence in bivariate time-series, via a qualitative overview of the literature and a quantitative evaluation of performance. This quantitative review, which I published in [21], tested a set of causal influence indices from the literature, using multiple simulated systems and common real-world data issues. I then summarised information-theoretic trajectories of patients in ICU with Covid-19, using some of the causal influence indices evaluated on physiological time-series.
- In **Chapter 3**, I modelled the physiological information-theoretic trajectories using a modular multilevel modelling framework. I developed a semi-conjugate integrated likelihood approach for estimating the Bayesian model evidence. I published this approach, which has general applicability to any high-dimensional multilevel model setting, in [22]. I used this Bayesian methodology to provide new insights into a clinical hypothesis of brainstem dysfunction in Covid-19 ICU patients, comparing the Covid-19 cohort against a second cohort of ICU patients with respiratory sepsis and similar disease severity.
- In **Chapter 4**, I introduced a fully unsupervised artefact detection framework, called DeepClean, which I have published in [23]. This framework uses a generative deep learning model, trained on artefact-free real observations, to create real-synthetic observation pairs. Alongside this generative model, I used an automatic threshold in post-processing to identify artefactual real observations, while using the generative model to provide a mechanism for data imputation when real observations were invalid. I illustrated the performance of the DeepClean framework on ABP waveform data.
- I investigated the potential, and challenges, of large-scale synthetic medical data in **Chapter 5**. I extended a privacy-related identifiability score, generalising this to a property of the underlying generative model. I then demonstrated the unsuitability of a state-of-the-art generative deep learning model, firstly at preserving information-theoretic measures within synthetic time-series observations and then on a downstream descriptive analysis of sepsis epidemiology in ICU.
- Lastly, I summarised my contributions in **Chapter 6**, with a discussion of their limitations and of possible future research directions.

As the statistical and mathematical methodology running through this thesis are wide-ranging, I begin each chapter with a brief motivation and introduction, mathematical problem definition, and a separate background and overview. I have endeavoured to make the mathematical notation I have used as consistent as possible (and then as concise as possible), particularly as there are common themes and ideas threaded through multiple chapters.

In particular, two main threads connect sections of this work. In Chapter 2, I introduced concepts from information theory and estimated their value on physiological time-series data. I then sought to model the temporal trajectories of information-theoretic measures in Chapter 3. I returned to these information-theoretic measures again in Chapter 5, comparing the empirical distribution of information-theoretic values across real and synthetic datasets. I also related the problem of latent representation learning in generative deep learning to information-theoretic ideas, in Chapter 4. This chapter explicitly introduced another central thread within my thesis, which is generative deep learning and synthetic data. Synthetic data generation played minor roles in Chapters 2 and 3, mostly in simulation studies. Both Chapters 4 and 5 built on ideas relating to this, with the former a use-case for synthetic data as a tool for artefact detection and the latter a cautionary summary analysis about the suitability and usability of synthetic datasets in modelling clinical time-series data.

CHAPTER 2

Information and causal influence between bivariate time series

This chapter presents my work towards describing and evaluating causal relationships between time-series variables. The context behind this chapter involved one of the defining global events of my PhD, the Covid-19 pandemic of the SARS-CoV-2 virus. Prior to Covid-19, I had been working on unsupervised representation learning to describe temporal trajectories of a patient's physiological state during ICU stay. Whilst on clinical duty in Addenbrooke's Hospital during the pandemic, Ari Ercole and his colleagues observed indications that Covid-19 appeared to damage brainstem function for some patients in ICU. This resulted in a new clinical hypothesis that patients with severe illness from respiratory viruses experienced impaired regulation of cardiovascular, respiratory and other physiological systems, and consequently reduced causal interaction between these systems. Evidence in support of neurological dysfunction in Covid-19 patients has since appeared in neurobiology and immunology studies [24, 25]. To explore this hypothesis in the context of physiological time-series, I shifted my focus from generative deep learning to causal influence estimation between multivariate physiological time-series, and whether this provided insights into the patient condition and their physiological regulation.

My initial literature review on the relevant theory revealed a crowded field of indices for estimating causal influence in time-series, and there was little clarity about the strengths and weaknesses of each method and the level of agreement between them. As a result, the first steps towards this research question were to categorise and evaluate a wide array of causal influence indices, including assessing their performance under various issues that systematically affect real-world data. My work on this was published in the journal *Chaos* [21]. I also presented this work on invitation at the conference 'The Flip Side of the Pandemic', which was held by the Isaac Newton Institute in Cambridge in May 2021.

2.1 Introduction

Uncovering causal structure is essential to a fuller understanding of interactions between sub-components of a system and, in turn, to building better and more parsimonious models [26, 27]. This means that inferring causal structure is very desirable across a wide array of scientific and data-driven fields. Causality is still a difficult concept in mathematical terms, having been mostly sidelined in favour of correlation and covariance based statistics for much of the 20^{th} century. In recent decades, different strands of causality theory (e.g. graph-based, structural equations) have been brought together and unified, but this theory is predominantly focused on static causality rather than dynamic causality. Judea Pearl identified a ladder of causation, in which there is a three-tier hierarchy of causal questions from association to intervention to counterfactuals (i.e. retrospection) [28], but this framework neglects temporal elements [29]. Methods for estimating causal structure within dynamic multivariate systems are generally not viewed in terms of a direct 'true' causality of 'event A causes event B', and for the most part these methods exist on the lowest rung of Pearl's ladder (i.e. association). However, these go beyond simple correlation-based measures, and are instead measures of directed causal influence from past states of the system to its current state.

The development of statistical tools for describing causal structure in multivariate time-series settings is a growing area of research, but estimating asymmetric and nonlinear causal relationships is a challenging task in practice, because data is inherently messy and unobserved confounding variables are difficult to handle. In many real-world applications, we are rarely able to describe some underlying causal structure beforehand. We are also typically limited to observing a small set of simultaneously recorded variables from different subsystems, but without a priori knowledge we may not even know which variables are important to measure. Furthermore, there are many domains in which measurements are over-invasive, expensive or difficult to make, which is certainly true of intensive care medicine. As an example, cardiac output (CO), which is the volume of blood ejected per minute by each ventricle, is a useful measurement because it can help identify cardiac tissue perfusion. CO cannot be directly monitored without invasive intra-cardiac procedures but can be estimated using waveform analysis of arterial blood pressure. However, the relationship between ABP and CO can be confounded by changes in ABP associated with arterial function instead of cardiac function, such as compliance or impedance [30]. Describing relationships between these and other cardiovascular variables in terms of asymmetric causal influence can help to guide decisions about when to measure invasively and create a more complete picture of cardiac function in general. In another example from cardiology, described in [5], causal analysis reveals lagged relationships between three variables (beat-to-beat intervals, systolic blood pressure and diastolic blood

pressure) derived from arterial blood pressure. These relationships depict the Frank-Starling mechanism, sympathovagal balance, and vasoconstriction/vasodilation due to respiration. It is clear from this example that causal influence in physiological systems can be very informative.

Various mathematical frameworks [26, 31, 32] have been described to allow identification of asymmetric and nonlinear causal structure within complex systems. This has been driven primarily by domain-specific applications, from diverse subjects including as statistical economics [33, 34], climate science [35–37] and computational neuroscience [38, 39]. In a longitudinal setting, two of the most important properties for causal influence are: that the effect is temporally preceded by the cause, and that external changes to values of the causal variable are propagated to changes in the values of the effect variable without breaking the causal association [40]. In many settings, it is difficult to actively intervene on the system and therefore it may only be possible to observe these external changes passively, which can conceal the role of unobserved confounding variables within the system. Correlation or synchronisation between observations does not necessarily imply a causal relationship between the time-series variables, and it is relatively straightforward to construct or find counterexamples [41]. Conversely, a lack of correlation does not imply a lack of causality and a reliance on correlation-based measures may result in nonlinear causal relationships being obscured [42].

Many real-world systems are intrinsically stochastic, with some degree of randomness in parts of the system. Other systems appear to be stochastic, because the system is too complicated and can only be modelled imperfectly, with unexplained elements treated as random noise. In stochastic systems, a further standard assumption for causal influence is separability. Separability is the statement that the causal variable alone contains unique information about the future of effect variable, which cannot be recovered from any other variable in the system. Most well-established techniques for identifying causal relationships function by describing the current state of the effect variable twice, first in terms of the 'recent history' of all variables and then in terms of the 'recent history' of variables excluding a potential causal variable. If there exists a causal relationship that is not mediated by other variables, then there will be some significant difference between the inclusion and exclusion of the 'recent history' of the causal variable in describing the current value of the effect variable. However, this is not the case universally. In particular, some systems with a very high signal-to-noise ratio are best viewed as deterministic dynamical systems, which evolve according to a group of differential equations or difference equations. These systems do not necessarily satisfy the separability condition, since the equations can often be reformulated in such a way that the effect variable can be rewritten purely as a function of past values of itself, even if there exist established causal relationships within the system. The consequence of this is that separability is not a strictly necessary

condition for identifying causal relationships, and methods that assume separability may perform inadequately in closed systems with minimal noise.

As a result, there is no general method or unifying notion of quantitative causality estimation in time-series data. Current methods can be broadly categorised into the following groups:

1. regression-based indices that use 'recent history' vectors as predictors in an autoregressive model (e.g. Granger causality),
2. information-theoretic indices based on conditional mutual information or other entropy-like measures (e.g. transfer entropy),
3. cross mapped indices based on state space trajectories and transitivity of local neighbourhood (e.g. convergent cross mapping),
4. network graph-based models that scale bivariate causal influence to high-dimensional multivariate systems.

These groups are not strictly mutually-exclusive, and there are some common themes and ideas between them. The suitability and interchangeability of different published methods has received relatively little attention, particularly between methods arising from different theoretical foundations. Previous meta-reviews of the literature [43–46] typically focused on a subset of methods from one of the groups. One exception to this was a review by Lungarella *et al.* [47].

2.1.1 Key contributions

In the paragraphs above, I identified three groups of causal influence indices for bivariate time-series (and a fourth group for multivariate time-series). The focus of the first half of this chapter was to reproduce, update and extend the analysis the review by Lungarella *et al.* [47]. The key contributions in this part were as follows:

1. I provided a comprehensive overview of causal influence indices for bivariate time-series, which included methods with origins in information theory, dynamical systems, and autoregressive modelling (Figures 2.1 and 2.2). This included the indices reviewed in [47], alongside two additional approaches from a literature search (excluding indices with non-uniform embeddings e.g. [48]). In Figure 2.1, I set out key properties and similarities between these methods. I first detailed motivation, mathematical formulation and estimation of this widely-used subset of indices. Previously, each index was introduced separately in the literature without a consistent unifying notation, despite there being common elements between them, i.e. univariate embeddings that describe the 'recent history' of the system. I therefore sought to

provide more consistency between their definitions, with minimal changes to the original formulation.

2. Following [47], I evaluated this set of causal influence indices on four simulated systems, which had different structures that are controlled by one or two 'coupling strength' parameters. I extended previous analysis by also investigating sensitivity to phenomena that regularly occur in real-world data, e.g. limited data availability, unequal data scaling, missing data, fixed measurement precision and rounding error, and noisy observation. These new tests are rarely considered in the literature, but should be seen as in-depth benchmarking criteria for new proposed methodologies in the future.

The goal of this review was to provide some clarity about which measures would be suitable for application to physiological time-series from Covid-19 ICU patients. In the second part of the chapter, I described these physiological relationships using an information-theoretic framework (including entropy, mutual information and transfer entropy). I provided examples of physiological time-series that have low and high transfer entropy, and visualised the trajectory of information-theoretic measures over time, where they were evaluated over 24hr windows over a period of multiple days in ICU.

2.1.2 Mathematical definition and notation

In a multivariate time-series, the state of the system at time t is defined as s_t, for $t = 1 \ldots, T$. Within this chapter, it is implicitly assumed that each time-series has unit time, i.e. data is observed or sampled at a fixed frequency. I also focused only on bivariate time-series, where the two variables are $x_t \in \mathcal{X}$ and $y_t \in \mathcal{Y}$. I denoted each full univariate time-series as $X = (x_1, \ldots, x_T)$ and $Y = (y_1, \ldots, y_T)$. The causal influence indices introduced later are all asymmetric, with directionality denoted by arrows. For simplicity, in most cases I introduced these indices in the direction $Y \to X$, but equivalent expressions exist for $X \to Y$. I denote a (general) causal influence index as $i_{Y \to X}$, and a relative (or net directed) causal influence index as $r_{Y \to X} = (i_{Y \to X} - i_{X \to Y})$.

Underpinning the notion of causal influence in time-series is the assumption that a cause strictly precedes its effect, so an important intermediate construct is the 'recent history' time-delay embedding vector $\boldsymbol{x}_t^{m,\tau}$ in m-dimensional state space $\boldsymbol{\mathcal{X}} \subset \mathbb{R}^m$, with lag τ. This is defined as:

$$\boldsymbol{x}_t^{m,\tau} = \left(x_{t-(m-1)\tau}, x_{t-(m-2)\tau}, \ldots, x_{t-\tau}, x_t\right)^T \in \boldsymbol{\mathcal{X}}, \quad t = (m-1)\tau + 1, \ldots, T$$

In the rest of this chapter, I omitted m and τ from the embedding vectors, unless explicitly stated otherwise. Instead, the time-delay embedding vector is denoted as $\boldsymbol{x}_t \in \boldsymbol{\mathcal{X}}$, while

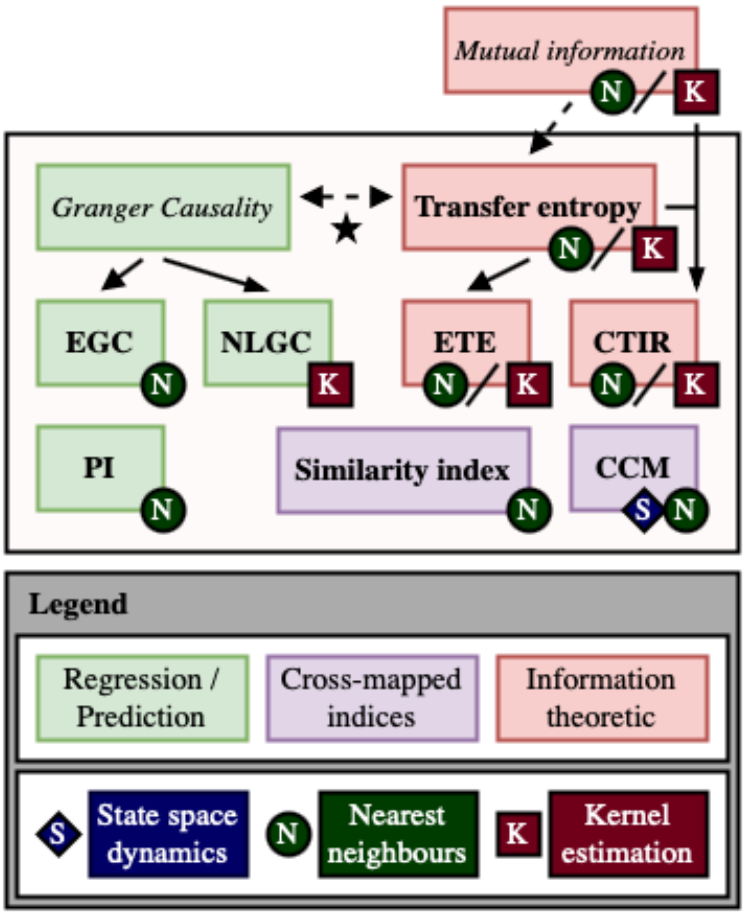

Figure 2.1: Categorisation of a widely-used subset of causal influence indices described in this chapter. The indices are as follows: extended Granger causality (EGC), nonlinear Granger causality (NLGC), predictability improvement (PI), transfer entropy (TE), effective transfer entropy (ETE), coarse-grained transinformation rate (CTIR), similarity indices (SI), convergent cross mapping (CCM). I identified three broad classes of bivariate causal influence indices, and highlighted similarities between the indices and their estimation (state space dynamics, nearest neighbour computation, kernel estimation). Transfer entropy and Granger causality are equivalent under Gaussian assumptions in the former ($\star$) [49]. Figure 2.2 shows a rough approximation of how these methods work.

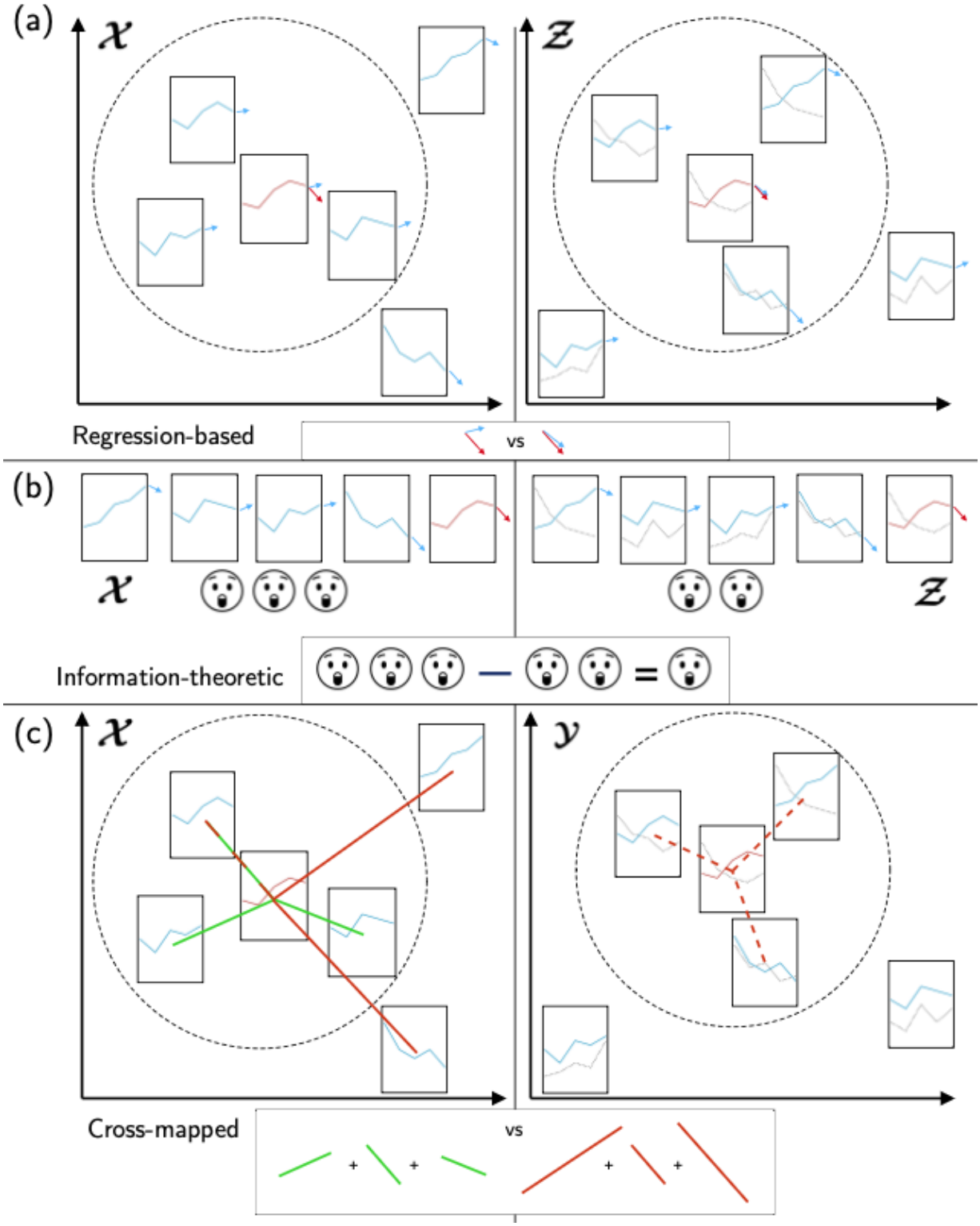

Figure 2.2: A schematic of common causal influence indices categories. The small time-series segments represent 'recent history' embedding vectors $\boldsymbol{x}_t$ (blue/red) and embedding vectors in $\boldsymbol{y}_t$ (black), with arrows showing the 'current value' x_t. Dotted circles indicate a set of nearest neighbours to the embedding vectors, in one of the embedding spaces. (a) Regression-based indices usually predict the 'current value' based on similar 'recent history' embeddings in $\boldsymbol{\mathcal{X}}$ and in $\boldsymbol{\mathcal{Z}}$, and compares these predictions. (b) Information-theoretic indices measure the reduction in 'expected surprise' of x_{t+1} from knowing $\boldsymbol{x}_t$ minus the reduction in 'expected surprise' of x_{t+1} from knowing $\boldsymbol{z}_t$. (c) Cross mapped indices usually compare the distances between $\boldsymbol{x}_t$ and its nearest neighbours in $\mathcal{X}$, and between $\boldsymbol{x}_t$ and points $\mathcal{Y}$ defined in terms of nearest neighbours from $\mathcal{Y}$.

$\boldsymbol{X} = (\boldsymbol{x}_{(m-1)\tau+1}, \ldots, \boldsymbol{x}_T)$. While I tended not to use boldface type for vectors during this thesis, to keep notation consistent between different chapters, I used boldface type for the embedding vectors in this chapter, in order to distinguish between an observation x_t and an embedding vector $\boldsymbol{x}_t$. For a given time-series length T, there are $T_2 = T - T_1$ embedding vectors from X and from Y, where $T_1 = (m-1)\tau + 1$. The embedding state spaces $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$ can be viewed as subspaces of a joint state space $\boldsymbol{\mathcal{Z}}$, where joint time-delay embedding vectors $\boldsymbol{z}_t$ are defined as $\boldsymbol{z}_t^{m,\tau} = \boldsymbol{z}_t = (\boldsymbol{x}_t^T, \boldsymbol{y}_t^T)^T$. The 'recent history' embedding vectors $\boldsymbol{z}_t$ define the potential causes in context of a 'future' effect, a value in X with horizon h, i.e. x_{t+h}. The horizon value is usually $h = 1$ and the interpretation tends to be of a 'current' value and 'recent history', i.e. with time index shifted to x_t and $\boldsymbol{z}_{t-1}$, but this is mathematically equivalent.

Causal influence indices from dynamical systems theory, in which the assumption of separability does not necessarily hold, focus more on the topology of the embedding state spaces $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$. In particular, this involves mappings between local neighbourhoods in $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$, necessitating nearest neighbour computations for $\boldsymbol{x}_t$, $\boldsymbol{y}_t$ and $\boldsymbol{z}_t$. Many indices for stochastic systems also require computation of nearest neighbours, either explicitly or as part of internal estimation algorithms. I denote the (ordered) R nearest neighbours of $\boldsymbol{x}_t$ using subscript indices $\pi_t(r),\ r = 1, \ldots, R < T_2$, i.e. the nearest neighbours are $\boldsymbol{x}_{\pi_t(r)}$. To clearly distinguish between different embedding spaces, I denote the S nearest neighbours of $\boldsymbol{y}_t$ using distinct subscript indices $\sigma_t(s),\ s = 1, \ldots, S < T_2$, and similarly the U nearest neighbours of $\boldsymbol{z}_t$ using subscript indices $\rho_t(u),\ u = 1, \ldots, U < T_2$. Nearest neighbours are defined with respect to a distance metric $d(\cdot, \cdot)$ and norm $\|\cdot\|$, the choice of which is often omitted in the literature. As a default, this refers to the Euclidean distance and ℓ_2 norm unless otherwise specified. Lastly, I denote the indicator function for some event or condition A as $\mathbb{1}\{A\}$, and the Heaviside function $\Theta(x) = \mathbb{1}\{x > 0\}$.

2.1.3 Overview and related work

Granger causality. Granger causality (GC) [26] is one of the most notable and popular concepts of causal influence in time-series. This is built on the principles of (i) temporal precedence ('cause precede effect') and (ii) separability ('cause contains unique information about effect'), and is formalised as the following: Y 'Granger-causes' X if it does not satisfy the condition $x_{t+1} \perp\!\!\!\perp Y^t \mid (U^t \setminus Y^t)\ \ \forall t$, where U^t is all the information in the universe up to time t, Y^t is all the information in Y up to time t and $U^t \setminus Y^t$ is all the information in the universe up to time t excluding that in Y (notation $A \perp\!\!\!\perp B \mid C$ denotes that A and B are conditionally independent given C). In practice, this is assessed using the weaker statement $x_{t+1} \perp\!\!\!\perp \boldsymbol{y}_t \mid \boldsymbol{x}_t$, by fitting separate autoregressive models with and without the potential causal variable (referred to as the full and reduced models respectively) and then

comparing the magnitude of error terms in each case, e.g.:

$$\mathcal{M}_r : x_{t+1} = \alpha^T \boldsymbol{x}_t + \epsilon_x$$
$$\mathcal{M}_f : x_{t+1} = \beta^T \boldsymbol{z}_t + \epsilon_z = \beta_1^T \boldsymbol{x}_t + \beta_2^T \boldsymbol{y}_t + \epsilon_z$$

The autoregressive models are often assumed to have Gaussian noise with mean zero and variances σ_x^2 and σ_z^2. In any case, the estimated error term variances are denoted s_x^2 and s_y^2 respectively. Y is said to have a causal influence on X if $s_x^2 > s_y^2$, and the standard linear GC index is defined as $\mathrm{GC}_{Y\to X} = \log(s_x^2) - \log(s_z^2)$.

Granger causality is a comparatively old but still widely-used concept, and has therefore been extensively studied. Most GC tests are performed using linear autoregressive models, but these are susceptible to overfitting when there are a large number of covariates and an insufficient quantity of data. It has also been shown that this comparison of variance between reduced and full regression models is itself biased with high variance [50]. However, unlike many other indices, there exist formal hypothesis tests for the statistical significance of Granger causality values [46], include an RSS-based F-test [51, 52] and the Hiemstra-Jones [53, 54] and Diks-Panchenko [55, 56] non-parametric tests. Both non-parametric tests are capable of generalisation to complex nonlinear systems, and the latter of the two has greater size and power when the linearity of the system is not known. Additionally, a number of extensions to this framework have been proposed to address the failures of the standard linear Granger causality in more complex nonlinear systems [57–59].

Nonlinear Granger causality. One extension to the GC approach is a 'global' nonlinear autoregressive model, called nonlinear Granger causality (NLGC) [57], which uses a radial basis function transformation on the autoregression covariates. This is defined as the following, where, as before, s_x^2 and s_z^2 are estimates of the error term variances:

$$\begin{aligned}
\mathcal{M}_r :\ & x_{t+1} = \alpha^T \Phi(\boldsymbol{x}_t) + \epsilon_x, \quad \Phi(\boldsymbol{x}_t) = (\phi_1(\boldsymbol{x}_t), \ldots, \phi_P(\boldsymbol{x}_t))^T \\
\mathcal{M}_f :\ & x_{t+1} = \beta^T \Psi(\boldsymbol{z}_t) + \epsilon_z, \quad \Psi(\boldsymbol{z}_t) = (\psi_1(\boldsymbol{z}_t), \ldots, \psi_P(\boldsymbol{z}_t))^T \\
& \mathrm{NLGC}_{Y\to X} = s_x^2 - s_z^2
\end{aligned} \tag{2.1}$$

The standard choice of radial basis functions (RBFs) are Gaussian RBFs with fixed variance, i.e. for $\mathcal{M}_r$, $\phi_p(\boldsymbol{x}_t) = \phi(||\boldsymbol{x}_t - \boldsymbol{c}_p||) = \exp(-||\boldsymbol{x}_t - \boldsymbol{c}_p||^2/(2\sigma^2))$, where centres $\boldsymbol{c}_p \in \boldsymbol{\mathcal{X}}$ are determined using a clustering algorithm, such as k-means or fuzzy c-means. Gaussian RBFs are used in both [47] and [57].

Extended Granger causality. Instead of a global autoregressive model, the extended Granger causality (EGC) [58] uses locally linear autoregression, where 'locally linear' refers to the joint embedding space $\boldsymbol{\mathcal{Z}}$. This involves applying Granger causality over

neighbourhoods $\mathcal{B}_l = \{\boldsymbol{z}_t : \|\boldsymbol{c}_l - \boldsymbol{z}_t\| < \delta\}$ for $l = 1, \ldots, L$, with centres $\boldsymbol{c}_l$ randomly sampled from the embedding vectors $\boldsymbol{Z}$, then averaging over the neighbourhoods:

$$\mathrm{EGC}_{Y \to X} = 1 - \frac{1}{L} \sum_l \frac{s^2_{x,l}}{s^2_{z,l}} \tag{2.2}$$

Predictability improvement. Another (non-GC) index also based on locally constant linear autoregression is predictability improvement (PI) [60]. In this approach, the 'horizon value' x_{t+h} with 'recent history' embedding $\boldsymbol{x}_t$ or $\boldsymbol{z}_t$ is estimated as an unweighted average of 'horizon values' that correspond to a set of similar 'recent history' embedding vectors. This is repeated for embedding vectors that are similar to $\boldsymbol{x}_t$ in embedding space $\boldsymbol{\mathcal{X}}$ and to $\boldsymbol{z}_t$ in embedding space $\boldsymbol{\mathcal{Z}}$ respectively, where the 'similar' embedding vectors are nearest neighbours of $\boldsymbol{x}_t$ and of $\boldsymbol{z}_t$ respectively (with $R = U$). The predicted 'horizon values' are then:

$$\tilde{x}_{t+h}|\boldsymbol{X} = \frac{1}{R} \sum_{r=1}^{R} x_{\pi_t(r)+h}, \quad \tilde{x}_{t+h}|\boldsymbol{Z} = \frac{1}{R} \sum_{u=1}^{R} x_{\rho_t(u)+h}$$

The notation used previously in [47] for this was not always clear, so for clarity, the prediction is for x_{t+h} in X rather than $\boldsymbol{x}_{t+h}$ in $\boldsymbol{X}$. The predictability improvement is the difference in mean squared error (MSE) between these predictions, over all possible 'horizon values':

$$\begin{aligned}
\mathrm{MSE}(X|\boldsymbol{X}) &= \frac{1}{T - h - T_1} \sum_{t=T_1}^{T-h} \left(x_{t+h} - \frac{1}{R} \sum_{r=1}^{R} x_{\pi_t(r)+h} \right)^2 \\
\mathrm{MSE}(X|\boldsymbol{Z}) &= \frac{1}{T - h - T_1} \sum_{t=T_1}^{T-h} \left(x_{t+h} - \frac{1}{R} \sum_{u=1}^{R} x_{\rho_t(u)+h} \right)^2 \\
\mathrm{PI}_{Y \to X} &= \mathrm{MSE}(X|\boldsymbol{X}) - \mathrm{MSE}(X|\boldsymbol{Z})
\end{aligned} \tag{2.3}$$

Information theory. Information theory is a natural framework for describing causal relationships, but estimating information-theoretic measures is generally much more challenging with continuous data than with discrete data. One of the fundamental building blocks of information theory is entropy, which is a measure of 'average uncertainty' or 'expected surprise' within the outcomes of a random variable. For a discrete random variable X that takes values x_i with non-zero probability $p(x_i)$, the Shannon entropy [61]

is defined as[1]:

$$H(X) = -\sum_{x_i \in \mathcal{X}} p(x_i) \log p(x_i) \tag{2.4}$$

The logarithmic base determines the unit of information, which is commonly 'bits' (base 2) or 'nats' (base e). I used the latter throughout this thesis. For discrete random variables X and Y, where the latter takes possible values y_j with marginal probability $p(y_j)$ and joint probability $p(x_i, y_j)$, two further information-theoretic measures are the joint entropy $H(X, Y)$ and the conditional entropy $H(X|Y)$, which are defined as:

$$H(X,Y) = -\sum_{x_i \in \mathcal{X}} \sum_{y_j \in \mathcal{Y}} p(x_i, y_j) \log p(x_i, y_j), \quad H(X|Y) = -\sum_{x_i \in \mathcal{X}} \sum_{y_j \in \mathcal{Y}} p(x_i, y_j) \log p(x_i|y_j)$$

The conditional entropy $H(X|Y)$ describes the 'average uncertainty' or 'expected surprise' remaining in X when the values of Y are known. These three entropy quantities together satisfy the equations $H(X, Y) = H(Y) + H(X|Y) = H(X) + H(Y|X)$.

The relative entropy between the probability distribution p and a reference distribution q, which is more commonly known as the Kullback-Leibler (KL) divergence [62], is a measure of the 'expected surprise' when an approximate distribution q is used to describe the X in place of the true probability distribution p, defined as:

$$D_{KL}(p(x) \,\|\, q(x)) = \sum_{x_i \in \mathcal{X}} p(x_i) \log \frac{p(x_i)}{q(x_i)} \tag{2.5}$$

The KL-divergence is non-symmetric in p and q, and non-negative unless $p = q$ almost everywhere. This is also widely-used in machine learning (and appears again in Chapter 4). This also defines another crucial quantity in information theory, the mutual information (MI) between X and Y:

$$\begin{aligned} I(X,Y) &= \sum_{x_i \in \mathcal{X}} \sum_{y_j \in \mathcal{Y}} p(x_i, y_j) \log \frac{p(x_i, y_j)}{p(x_i)p(y_j)} = D_{KL}(p(x,y) \,\|\, p(x)p(y)) \\ &= H(X) - H(X|Y) = H(X) + H(Y) - H(X,Y) \end{aligned} \tag{2.6}$$

Mutual information is the amount of 'information' communicated about each variable by the other, or equivalently the 'average uncertainty' in one of the variables that is removed by knowing the other. This latter description is apparent in the relationships between MI, Shannon entropy, joint entropy and conditional entropy. However, mutual information

[1]Throughout this subsection about information theory, X and Y are any random variables. This is a departure from the rest of this chapter, notably Section 2.1.2, in which X and Y are complete time-series data.

is symmetric in X and Y and so it is not informative about directional information flow from X to Y or vice versa.

For continuous variables, most of these definitions hold in the continuous limit, i.e. with sums replaced by integrals. For example, the mutual information is:

$$I(X,Y) = \int_{\mathcal{X},\mathcal{Y}} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} dxdy$$

The exceptions to this are Shannon entropy and joint entropy. The continuous extension of Shannon entropy is called the limiting density of discrete points (LDDP) [63]. Defining $m(x)$ as the 'invariant measure' of the set of discrete points $\{x_i\}$ in the infinite limit, the LDDP is defined as:

$$H(X) = -\int_{\mathcal{X}} p(x) \log \frac{p(x)}{m(x)} dx$$

In contrast, the (continuous) differential entropy, as introduced by Shannon, is neither invariant under coordinate transformation nor dimensionless, but is very closely related to LDDP when $m(x)$ is approximately constant over the support of the distribution $p(x)$. Differential entropy is defined as:

$$h(X) = -\int_{\mathcal{X}} p(x) \log p(x) dx$$

In practice, a finite amount of data is observed, so entropies must be estimated non-parametrically or by assuming some distributional forms. The simplest method, and therefore one of the most common, is to use equidistant partitions in each joint probability space and count the number of observations in each bin in order to estimate the joint probabilities with a discrete approximation. However, this performs sub-optimally because of the 'curse of dimensionality', e.g. resulting in systematic overestimation of mutual information [64]. Another approach is multivariate kernel density estimation, e.g. $\hat{p}(x) = 1/h^m \sum_{i=1}^{m} K((x - x_i)/h)$ for some kernel function $K(\cdot)$ and width parameter h. The most common choice of kernel is a standard Gaussian, i.e. $K(x) = (2\pi)^{-m/2} \exp(-x^T x/2)$, where m is the dimension of $\mathcal{X}$. Finally, adaptive partitioning methods are typically superior to fixed partition methods [43]. These often involve nearest-neighbour distances in estimating local density, e.g. weighted Kozachenko-Leonenko estimators [65] and higher-order extensions of these [66]. An approach of this type is described in the next section.

Transfer entropy. In the previous paragraphs, I described information theoretic in terms of discrete and continuous random variables. In the following paragraphs ('Transfer entropy', 'Effective transfer entropy' and 'Coarse-grained transinformation rate'), I discuss

these concepts in terms of ergodic random processes with sequential observations. Following the notation in [67], these random processes are denoted as X_t and Y_t, and take states x_t and y_t. The probability distribution of process X_t is denoted $\mathbb{P}_{X_t}$, with $p(x_t) = P_{X_t}(\{x_t\})$. The embedding vectors $\boldsymbol{x}_t$, $\boldsymbol{y}_t$ and $\boldsymbol{z}_t$ are states of composite random processes $\boldsymbol{X}_t$, $\boldsymbol{Y}_t$ and $\boldsymbol{Z}_t$ respectively. Throughout the following, it is assumed that X_t and Y_t are Markov processes with suitably-defined transition probabilities, e.g. $p(x_{t+1}|\boldsymbol{x}_t) = P_{X_{t+1}|\boldsymbol{X}_t}(\{x_{t+1}\})$ or $p(x_{t+1}|\boldsymbol{z}_t) = P_{X_{t+1}|\boldsymbol{Z}_t}(\{x_{t+1}\})$. In order to estimate transition probabilities, it is often necessary to assume that the process is stationary, though there are some exceptions to this e.g. multiple recordings of evoked potentials [32]. As in Section 2.1.2, X and $\boldsymbol{X}$ denote the notation for the full time-series recording and corresponding embedding, rather than a random variable or random process.

Transfer entropy (TE) is a KL-divergence that measures the additional information that the 'recent history' of the causal variable provides about the 'current' value of the effect variable, above and beyond knowing the 'recent history' of just the effect variable. If the generalised Markov property holds, i.e. $p(x_{t+1}|\boldsymbol{z}_t) = p(x_{t+1}|\boldsymbol{x}_t)$, then the embeddings $\boldsymbol{Y}$ have no relevance to the transition probabilities of X and there is no information transfer from Y to X. A deviance from the generalised Markov property is measured by the KL-divergence, which can then also be rewritten as a sum of entropy terms:

$$\begin{aligned}
\mathrm{TE}_{Y\to X} &= \sum_t p(\boldsymbol{z}_t, x_{t+1}) \log \frac{p(\boldsymbol{z}_t, x_{t+1})}{p(x_{t+1}|\boldsymbol{x}_t)p(\boldsymbol{z}_t)} = D_{KL}(p(\boldsymbol{z}_t, x_{t+1}), p(x_{t+1}|\boldsymbol{x}_t)p(\boldsymbol{z}_t)) \\
&= \sum_t p(\boldsymbol{z}_t, x_{t+1}) \log \frac{p(x_{t+1}|\boldsymbol{z}_t)}{p(x_{t+1}|\boldsymbol{x}_t)} = H(X_{t+1}|\boldsymbol{X}_t) - H(X_{t+1}|\boldsymbol{Z}_t) = I(\boldsymbol{Y}_t, X_{t+1}|\boldsymbol{X}_t) \\
&= \sum_t p(\boldsymbol{z}_t, x_{t+1}) \log \frac{p(\boldsymbol{z}_t, x_{t+1})p(\boldsymbol{x}_t)}{p(\boldsymbol{x}_t, x_{t+1})p(\boldsymbol{z}_t)} \\
&= -H(\boldsymbol{X}_t) + H(\boldsymbol{X}_t, X_{t+1}) - H(\boldsymbol{Z}_t, X_{t+1}) + H(\boldsymbol{Z}_t) \qquad (2.7)
\end{aligned}$$

This highlights an equivalence between transfer entropy and conditional mutual information [68]. The second line in the equations above also shows that transfer entropy can be understood as a difference in conditional entropies, i.e. the reduction in 'average uncertainty' of x_{t+1} from knowing only $\boldsymbol{x}_t$, minus the reduction in 'average uncertainty' in x_{t+1} from knowing $\boldsymbol{z}_t$. However, this definition of transfer entropy is a function of the random process, and it must be estimated, directly or through estimates of the transition probabilities.

As noted in Figure 2.1, transfer entropy reduces to linear autoregressive Granger causality under the assumption of multivariate Gaussian variables in the former [49], up to a factor of multiplicative factor. The two concepts are often framed differently however, i.e. Granger causality as 'prediction' and TE as 'disambiguation' (removing uncertainty). Non-zero TE is in fact a stronger statement than Granger causality, since the latter implies violation of the generalised Markov property [59]. Transfer entropy can therefore be

considered a generalised information-theoretic extension of Granger causality, which is both non-parametric and nonlinear (though more difficult to estimate).

When transfer entropy was first introduced by Schreiber in [32], he proposed that the transition probabilities $p(\boldsymbol{z}_t, x_{t+1})$ and $p(\boldsymbol{x}_t, x_{t+1})$ could be estimated using correlation integrals from chaos theory. For a fixed coarse-grained resolution r, the correlation integrals are of the form:

$$\hat{p}(\boldsymbol{z}_t, x_{t+1}) = \frac{1}{T-1-T_1} \sum_{\substack{j=T_1 \\ j \neq t}}^{T-1} \Theta\left(r - \left\| \begin{pmatrix} \boldsymbol{z}_t \\ x_{t+1} \end{pmatrix} - \begin{pmatrix} \boldsymbol{z}_j \\ x_{j+1} \end{pmatrix} \right\| \right)$$

In this chapter, I investigated two different partition-based methods instead. The first was the commonly-used histogram method, in which the (continuous) data is discretised using histogram bins and transition probabilities are estimated using counts from each partition bin. The second estimation method, called Kraskov-Stögbauer-Grassberger (KSG) after [69], uses adaptive partitioning, via k-nearest neighbour distances. This was initially introduced for estimation of mutual information, as a more robust version of the mutual information estimate derived from the Kozachenko-Leonenko estimate [65]. It has since been generalised to a class of functionals called 'entropy combinations' [70], which includes transfer entropy. The first step of this algorithm defines a ball $\mathcal{B}_t$ around the point $(\boldsymbol{z}_t^T, x_{t+1})^T \in \boldsymbol{\mathcal{Z}} \times \mathcal{X}$, with radius r_t equal to the k^{th} nearest neighbour distance in this joint space, where k is small (e.g. $k = 2, 3, 4$). This ball $\mathcal{B}_t$ is in fact a hypercube, since both ball and radius are defined using the maximum ℓ_∞ norm. Next, the hypercube is projected into each of the relevant subspaces ($\boldsymbol{\mathcal{Z}}$, $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{X}} \times \mathcal{X}$) to calculate the number of points in that subspace that lie within the hypercube, i.e.:

$$n(\boldsymbol{z}_t) = \sum_{j \neq t} \mathbb{1}\{\|\boldsymbol{z}_t - \boldsymbol{z}_j\|_\infty < r_t,\ \boldsymbol{z}_j \in \boldsymbol{\mathcal{Z}}\}, \quad n(\boldsymbol{x}_t) = \sum_{j \neq t} \mathbb{1}\{\|\boldsymbol{x}_t - \boldsymbol{x}_j\|_\infty < r_t,\ \boldsymbol{x}_j \in \boldsymbol{\mathcal{X}}\}$$

$$n(\boldsymbol{x}_t, x_{t+1}) = \sum_{j \neq t} \mathbb{1}\left\{ \left\| \begin{pmatrix} \boldsymbol{x}_t \\ x_{t+1} \end{pmatrix} - \begin{pmatrix} \boldsymbol{x}_j \\ x_{j+1} \end{pmatrix} \right\|_\infty < r_t,\ \begin{pmatrix} \boldsymbol{x}_j \\ x_{j+1} \end{pmatrix} \in \boldsymbol{\mathcal{X}} \times \mathcal{X} \right\}$$

The KSG estimate is then defined as the following, where $\psi(\cdot)$ is the digamma function:

$$\mathrm{TE}^{\mathrm{KSG}}_{Y \to X} = \psi(k) + \frac{1}{T-T_1} \sum_{t=T_1}^{T} \Big(\psi(1 + n(\boldsymbol{x}_t)) - \psi(1 + n(\boldsymbol{z}_t)) - \psi(1 + n(\boldsymbol{x}_t, x_{t+1})) \Big) \quad (2.8)$$

Similar estimates for entropy and mutual information in the embedding spaces are:

$$H_X^{\mathrm{KSG}} = 1 + \psi(T) - \frac{1}{T - T_1} \sum_{t=T_1}^{T} \Big(\psi(1 + n(\boldsymbol{x}_t)) - m \log(2r_t) \Big) \tag{2.9}$$

$$\mathrm{MI}_{X,Y}^{\mathrm{KSG}} = \psi(k) + \psi(T) - \frac{1}{T - T_1} \sum_{t=T_1}^{T} \Big(\psi(1 + n(\boldsymbol{x}_t)) + \psi(1 + n(\boldsymbol{y}_t)) \Big) \tag{2.10}$$

Effective transfer entropy. When estimated using an equidistant partition, transfer entropy can be biased by finite sample effect, which is the same source of bias that is accounted for in LDDP. Empirical evidence suggests this is particularly true for weak causal influence or limited data quantity [71]. For transfer entropy estimation, this bias is estimated as the transfer entropy calculated for the same effect variable X and a block-shuffled version of the causal variable Y, which is denoted Y_s. The block-shuffle is performed by breaking the time-series Y into blocks of length l and reordering these blocks (i.e. sampling without replacement) into a time-series of the same length. The block length is typically taken to be $l = 1$ [71]. Correcting for the finite sample bias by averaging over multiple block-shuffled $Y_{s,j}$, $j = 1 \ldots, S$, the effective transfer entropy (ETE) is defined as:

$$\mathrm{ETE}_{Y \to X} = \mathrm{TE}_{Y \to X} - \frac{1}{S} \sum_{j=1}^{S} \mathrm{TE}_{Y_{s,j} \to X} \tag{2.11}$$

Coarse-grained transinformation rate. The rate at which a random process X_t 'forgets' its history, and consequently the 'information creation' of the process, is estimated using a concept called coarse-grained entropy rates [72]. This is extended to bivariate time-series in the coarse-grained transinformation rate (CTIR) [73], which measures the rate of net information flow averaged over multiple lags τ, i.e.:

$$\begin{aligned} \mathrm{CTIR}_{Y \to X} &= \frac{1}{\tau_{\max}} \sum_{\tau=1}^{\tau_{\max}} \hat{I}(X_{t+\tau}, Y_t | X_t) - \frac{1}{2\tau_{\max}} \sum_{\substack{\tau=-\tau_{\max} \\ \tau \neq 0}}^{\tau_{\max}} \hat{I}(X_{t+\tau}, Y_t) \\ &= \frac{1}{2\tau_{\max}} \sum_{\tau=1}^{\tau_{\max}} \Big[2\hat{I}(X_{t+\tau}, Y_t | X_t) - \hat{I}(X_{t+\tau}, Y_t) - \hat{I}(Y_t, X_{t-\tau}) \Big] \end{aligned} \tag{2.12}$$

CTIR is defined over states in $\mathcal{X}$ and $\mathcal{Y}$, rather over embedding vectors in the embedding spaces $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$. The mutual information estimates can be made using any of the above techniques, e.g. I used the KSG algorithm (Equation 2.10). The sum is made over lags for which there is non-zero information transfer in $\mathcal{X}$, i.e. $\tau_{\max}$ is defined such that $\hat{I}(X_t, X_{t+\tau}) \approx 0, \ \forall \tau \geq \tau_{\max}$.

Dynamical systems. A deterministic dynamical system evolves within a state space according to a differential equation, e.g. $\dot{\boldsymbol{\eta}}(t) = f(\boldsymbol{\eta}(t))$, or a discrete difference equation, e.g. $\boldsymbol{\eta}_{t+1} = F(\boldsymbol{\eta}_t)$. In general, many systems are governed by higher-order equations rather than these first-order examples. Both types of equation can be directly linked to each other, i.e. a difference equation is the discretisation of an ordinary differential equation under a given numerical method (such as the Euler method). For simplicity, I focused on difference equations in what follows. Given suitable starting conditions, the system may converge to a manifold Γ. If the system does converge to this manifold, it will remain on the manifold thereafter, i.e. the function F maps Γ to Γ. Additionally, if the system experiences a small external perturbation away from the manifold, its subsequent evolution may see it return to the manifold again. In this case, the manifold is called an attractor. The bivariate time-series x_t and y_t are usually coordinate projections of the vector $\boldsymbol{\eta}_t$, i.e. $\boldsymbol{\eta}_t = \boldsymbol{z}_t$, but more generally they may be the output of any injective functions acting on $\boldsymbol{\eta}_t$, i.e. $x_t = g_x(\boldsymbol{\eta}_t)$ and $y_t = g_y(\boldsymbol{\eta}_t)$.

Many multivariate dynamical systems are governed by first-order equations but can be rewritten as a set of univariate higher-order equations, particularly if the function $F(\cdot)$ is fully invertible. For example, the bivariate first-order system $x_{t+1} = y_t,\ y_{t+1} = c - x_t + y_t$ with initial conditions $x_0 = a,\ y_0 = b$, is essentially equivalent to two univariate systems, $x_{t+1} = c + x_t - x_{t-1}$ with $x_0 = a,\ x_1 = b$, and $y_{t+1} = c + y_t - y_{t-1}$ with $y_0 = b,\ y_1 = c + b - a$. Since x_t can be reformulated as a function of only past values in X, it may seem reasonable to conclude that the variable y_t has no direct causal influence on x_t in this case, because the univariate system for X alone appears to fully describe the dynamics of x_t. However, this neglects aspects of a causality that cannot easily be discerned from observation or association alone, in particular intervention on the system. The example first-order bivariate model above is a simplistic closed-system model of breath volume (x_t) and carbon dioxide levels in the blood (y_t), an example from a Cambridge Tripos lecture course. In this model, the level of carbon dioxide in the blood can be fully calculated without knowing the breath volume, but this cannot account for external intervention acting on only the breath volume, e.g. deliberately holding a breath. Incidentally, if this system is not perturbed, it has a periodic orbit of 6 distinct points (a 6-cycle), which is the manifold Γ for this system. These considerations about causal structure are long-standing, and were noted by Granger in a discussion of the limitations of Granger causality [26].

The reason that causal relationships in deterministic coupled systems are difficult to evaluate is that the relationship is usually bidirectional, which creates a feedback loop mechanism. The main methods for identifying causal influence in this setting instead come from observing the event that each component time-series belongs to some shared attractor manifold A within the full manifold Γ. In this case, Takens' embedding theorem [74] states that, for m sufficiently large, there exists a 1-1 mapping between the trajectory

of $\boldsymbol{x}_t$ in m-dimensional space $\boldsymbol{\mathcal{X}}$ and the attractor manifold $A \subset \Gamma$ and a similar 1-1 mapping for $\boldsymbol{\mathcal{Y}}$. A consequence of this result is that the trajectory of $\boldsymbol{x}_t$ (which is referred to in this instance as the 'library of historical behaviour' of X) converges to a 'shadow attractor' manifold $A_{\boldsymbol{X}} \subset \boldsymbol{\mathcal{X}}$ that preserves the topology of the full system, as $T \to \infty$. If X and Y are causally related, and so belong to the same attractor A, then both $A_{\boldsymbol{X}} \subset \boldsymbol{\mathcal{X}}$ and $A_{\boldsymbol{Y}} \subset \boldsymbol{\mathcal{Y}}$ are diffeomorphic to A, and 'local neighbourhoods' on $A_{\boldsymbol{X}}$ will map to 'local neighbourhoods' on $A_{\boldsymbol{Y}}$ by transitivity. Many causal influence indices from dynamical systems theory therefore use a technique called cross mapping, which assumes that neighbourhoods in the trajectories of the embedding vectors $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ coincide, i.e. indices denoting the nearest neighbours of $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ roughly share the same ordering, which means that replacing the nearest-neighbouring indices of one variable with those of the other still preserves its 'local neighbourhood'. As this process does not directly involve a link between the 'current value' x_{t+1} and the 'recent history' $\boldsymbol{z}_t$, some of these indices are more correctly measures of synchrony than measures of causal influence.

Similarity indices. One of the simplest indices that uses cross mapping is the similarity index [75, 76]. The average nearest neighbour distance and average cross mapped distance are defined as the following, where the indices $\pi_t(r)$ and $\sigma_t(s)$ identify nearest neighbours of $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ respectively:

$$d_t^R(\boldsymbol{X}) = \frac{1}{R}\sum_{r=1}^{R} ||\boldsymbol{x}_t - \boldsymbol{x}_{\pi_t(r)}||_2^2, \quad d_t^S(\boldsymbol{X}|\boldsymbol{Y}) = \frac{1}{S}\sum_{s=1}^{S} ||\boldsymbol{x}_t - \boldsymbol{x}_{\sigma_t(s)}||_2^2$$

The former is the minimum average distance between $\boldsymbol{x}_t$ and all possible sets of R points in $\boldsymbol{X}$, which means that the inequality $d_t^R(\boldsymbol{X}) \leq d_t^R(\boldsymbol{X}|\boldsymbol{Y})$ always holds. If there is a strong degree of synchronisation between $\boldsymbol{X}$ and $\boldsymbol{Y}$, then the two sets of nearest neighbours will approximately coincide and $d_t^R(\boldsymbol{X}) \approx d_t^R(\boldsymbol{X}|\boldsymbol{Y})$. However, if the two variables are independent and $R \ll T_2$ (the total number of embedding vectors), then the cross mapped indices $\{\sigma_t(1), \ldots, \sigma_t(R)\}$ will be an almost random sample from within the complete ordered indices $\{\pi_t(1), \ldots, \pi_t(T_2)\}$. In this case, $d_t^R(\boldsymbol{X}|\boldsymbol{Y}) \approx d_t^{T_2}(\boldsymbol{X}) \gg d_t^R(\boldsymbol{X})$. Using this idea, several indices were proposed [75, 76], including:

$$\mathrm{SI}_{Y\to X}^{(1)} = \frac{1}{T_2}\sum_{t=T_1}^{T} \log \frac{d_t^{T_2}(\boldsymbol{X})}{d_t^R(\boldsymbol{X}|\boldsymbol{Y})}, \quad \mathrm{SI}_{Y\to X}^{(2)} = \frac{1}{T_2}\sum_{t=T_1}^{T} \frac{d_t^{2R}(\boldsymbol{X}) - d_t^R(\boldsymbol{X})}{d_t^R(\boldsymbol{X}|\boldsymbol{Y})} \tag{2.13}$$

Both of these causal influence indices adjust the ratio $d_t^R(\boldsymbol{X})/d_t^R(\boldsymbol{X}|\boldsymbol{Y})$ by amplifying the contribution of $d_t^R(\boldsymbol{X}|\boldsymbol{Y})$ in the summand, motivated by empirical evidence that suggested these changes make the indices more robust [77]. In the former [75], this involves averaging over all T_2 nearest neighbours of $\boldsymbol{x}_t$ (in the numerator). The latter [76] uses the R^{th} to $2R^{\text{th}}$

nearest neighbours of $\boldsymbol{x}_t$, which appears to resolve issues that arise when the time-series are noisy and weakly coupled.

Convergent cross mapping. Convergent cross mapping (CCM) [42] is a popular method from dynamical systems theory, which was formulated specifically for non-separable and weakly-coupled deterministic systems. Assuming that the embedding vectors $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ exist on diffeomorphic 'shadow attractor' manifolds $A_{\boldsymbol{X}}$ and $A_{\boldsymbol{Y}}$ respectively, CCM measures the Pearson correlation between the 'current value' y_t and an exponentially weighted average of cross mapped 'current values' $\hat{y}_t|A_{\boldsymbol{X}}$, defined as:

$$\hat{y}_t|A_{\boldsymbol{X}} = \sum_{r=1}^{m+1} w_{t,r} y_{\pi_t(r)}, \quad w_{t,r} = \frac{u_{r,t}}{\sum_{j=1}^{m+1} u_{t,j}}, \quad u_{t,r} = \exp\left(-\frac{||\boldsymbol{x}_t - \boldsymbol{x}_{\pi_t(r)}||_2}{||\boldsymbol{x}_t - \boldsymbol{x}_{\pi_t(1)}||_2}\right)$$
$$\text{CCM}_{Y\to\boldsymbol{X}} = \rho(y_t, (\hat{y}_t|A_{\boldsymbol{X}})) \tag{2.14}$$

This can be calculated with only $R = m + 1$ nearest neighbours, since this number is necessary but not sufficient to form a 'bounding simplex' for $\boldsymbol{x}_t$.

One unique point about CCM is that, somewhat counterintuitively, cross mapping from $A_{\boldsymbol{X}}$ to $\boldsymbol{y}_t$ is used to define causal influence from Y to X rather than from X to Y. Sugihara *et al.* [42] argue that the more intuitive directionality is reversed in their approach because, if Y drives X unilaterally, information about Y should be encoded in the manifold $A_{\boldsymbol{X}}$, but not neccessarily vice versa. Equivalently, under this unidirectional coupling, the 'shadow attractor' manifold $A_{\boldsymbol{X}}$ is diffeomorphic to the manifold A, which means that cross mapping of $\boldsymbol{Y}$ using $A_{\boldsymbol{X}}$ does in fact converge to $\boldsymbol{Y}$, while the same is not true of $A_{\boldsymbol{Y}}$. The crux of the argument is that, in a weakly-coupled system where many processes may causally influence the effect, past values of a given causal variable can be inferred from the response variable, since the dynamics of the causal variable continually propagate to the response variable, whereas forecasting the 'current value' using 'recent history' is ineffective since any given causal variable alone is a poor predictor of the effect variable. As such, they also argue that the direction of causal influence can be wrongly inferred in methods that involve a cross mapped reconstruction of the embedding space manifolds.

Finally, a critical additional step involves ensuring convergence of $\boldsymbol{X}$ to an attractor manifold that is diffeomorphic to $A_{\boldsymbol{Y}}$, as $T \to \infty$. Sugihara *et al.* [42] remark that this is "a key property that distinguishes causation from simple correlation". Convergence is implemented by calculating the average Pearson correlation $\rho(y_t, (\hat{y}_t|A_{\boldsymbol{X}}))$ for multiple smaller time-series segments of length $T' < T$, as T' is increased up to T. This helps to distinguish between causal influence from Y and X (in which case ρ increases with T') and external forcing acting jointly on non-coupled X and Y (which results in positive but non-increasing ρ). There is no clear consensus on how to test this convergence property.

One suggestion [78] is to fit an exponential regression $\rho(T') = (\rho_0 - \rho_\infty)\exp(-\gamma T') + \rho_\infty$, where the reported value of CCM is ρ_∞, provided $\gamma > 0$ and $\rho_\infty - \rho_0 = \delta_\rho > 0$. Another study [79] suggested $\rho(T')$ may be poorly described by a parametric curve and the reported CCM value should simply be the correlation $\rho_\infty = \rho(T)$ for the maximum time-series length T, provided this is larger than $\rho_0 = \rho(T')$ for some minimal time-series length $T' \ll T$, i.e. $\rho_\infty - \rho_0 = \delta_\rho > 0$.

Hyperparameters. Time-delay embedding introduces two hyperparameters, the embedding dimension m and the lag τ. Dynamical systems theory provides a priori 'optimal' choices for both [80–82]. However, this may not always be reliable or consistent, and other authors suggest that the domain-specific or empirical approaches are better [75, 83]. In practice, common choices for these parameters are $\tau = 1$ and $m = 1$ or $m = 2$ [47], which I adopted in this chapter. As well as embedding parameters m and τ, each causal influence index has a number of additional hyperparameters, which are detailed in Tables 2.1 and A.1.

Multivariate systems and confounders. Most bivariate causal influence indices can be be extended to a multivariate setting with multiple possible confounding variables $C = (c_1, \ldots, c_T)$. In low-dimensional settings (in terms of the number of components in the system), it is generally straightforward to investigate whether the causal structure is mediated by a multivariate confounder C, by estimating the causal influence from Y to X conditioned upon C, i.e. $i_{(Y \to X)|C}$. This allows distinction between indirect and direct causation [44]. For instance, if an estimated causal relationship between X and Y subsequently disappears with the inclusion of another time-series variable C, then the causal relationship between X and Y is mediated by C. Conversely, the omission or exclusion of extra variables can create spurious false-positive causal influence due to overfitting [44, 84]. That said, most causal influence methods suffer from the 'curse of dimensionality' when an increasing number of components are included [85], alongside additional combinatorially-increasing computational requirements.

In high-dimensional settings, graph-based network models can be used to generalise Granger causality or transfer entropy to directed acyclic graphs that represent variables as nodes and causal relationships as edges [36]. Until recently, graph-based network models were more commonly used in 'static' causal inference theory rather than for time-series data. However, algorithms like PCMCI [36] have gained significant popularity in recent years. PCMCI is an improved version of full conditional independence tests between all variables and across multiple lags. This involves a modified version of the Peter-Clark algorithm for causal Markov discovery, alongside a momentary conditional independence test. One of the key features of this algorithm is that it automatically identifies values of the embedding hyperparameters (τ, m) to include within this graph-based network.

	Method	Hyperparameters/other choices		Notes and suggestions	Values used here
Embedding	All	T	Time-series length	Depends on data availability	$10^p,\ p = 3, 4, 5$
		h	Time horizon value	$h = 1, 2, \ldots$, but normally $h = 1$	$h = 1$
		m	Embedding dimension	'Optimal' [81] vs empirical (e.g. $m = 1, \ldots, 5$)	$m = 1$ or 2
		τ	Time-delay lag	'Optimal' [82] vs empirical (e.g. $\tau = 1, 2, 3$)	$\tau = 1$
Regression error	EGC [58]	Nearest neighbour (NN) metric		ℓ_p, may depend on state space/distribution	Manhattan, ℓ_1
		L	No. of neighbourhoods	Depends on T. In [47, 58], $L = 100$	$L = 20$ or 100
		$\boldsymbol{c}_l$	Neighbourhood centres	Sampled randomly from $\boldsymbol{Z}$	Sampled from $\boldsymbol{Z}$
		δ	Neighbourhood size	E.g. compute EGC for $\delta \to 0$ [58]	Various (†)
	NLGC [57]	Radial basis function (RBF)		Gaussian RBFs in [47, 57]	Gaussian
		P	No. of RBFs	e.g. 'Optimal' via gap statistics [86]	Various (†)
		$\boldsymbol{c}_p$	Gaussian RBF centres	Clustering centroids via k-means or fuzzy c-means	via k-means
		σ^2	Gaussian RBF variance	Typically fixed, e.g. $\sigma^2 = 0.05$ in [47, 57]	$\sigma^2 = 0.05$
	PI [60]	Nearest neighbour (NN) metric		ℓ_p, may depend on state space/distribution	Euclidean, ℓ_2
		R	No. of NNs	Not clear, but e.g. $R = 1, 10$ in [47, 60]	$R = 1$ or 10
		h	Time horizon value	As above, e.g. $h = 1$ in [47, 60]	$h = 1$
Information theory	TE (KSG) [32, 69]	Nearest neighbour (NN) metric		ℓ_∞ (for the hypercube) [69]	Maximum, ℓ_∞
		k	No. of NNs	Small values e.g. $k = 2, 3, 4$ [69]	$k = 4$
	TE (H) [32]	N	No. of bins	e.g. via minimum description length [87, 88]	$N = 8$
	ETE [71]	S	No. of shuffled Y	Not clear, but e.g. single shuffle in [71]	$S = 10$
		l	Block length	Not clear, but e.g. a simple shuffle in [71]	$l = 1$
	CTIR [73]	$\tau_{\max}$	Max time-delay lag	Such that $\hat{I}(X_t, X_{t+\tau}) \approx 0,\ \forall \tau \geq \tau_{\max}$ [73]	$\tau_{\max} = 5$ or 20
Cross mapped	SI [75, 76]	Nearest neighbour (NN) metric		ℓ_p, may depend on state space/distribution	Euclidean, ℓ_2
		R	No. of NNs	Not clear, but e.g. $R = 10$ in [75, 76]	Various (†)
	CCM [42]	Nearest neighbour (NN) metric		ℓ_p, may depend on state space/distribution	Euclidean, ℓ_2
		$T_{\max}$	Max. segment length	Compute ρ for segment length $T' \to T_{\max}$ [42]	$T_{\max} = T$
		$n_{T'}$	No. segments of size T'	ρ values averaged across $n_{T'}$ segments, size T'	$n_{T'} = 40$
		ρ_∞	Converged CCM value	$\rho_{T_{\max}}$ in [79] or exponential regression [78]	$\rho_{T_{\max}}$
		δ_ρ	ρ tolerance	e.g. ρ_∞ is valid if $\rho_\infty - \rho_{m+2} > \delta_\rho$	$\delta_\rho = 0.05$ [79]

Table 2.1: Causal influence indices reviewed in this chapter, and their hyperparameters. The indices are as follows (where GC is Granger causality): extended GC (EGC), nonlinear GC (NLGC), predictability improvement (PI), transfer entropy (TE), effective transfer entropy (ETE), coarse-grained transinformation rate (CTIR), similarity indices (SI) and convergent cross mapping (CCM). Table A.1 provides more detail on the parameter choices for individual simulation results (†).

However, I did not focus on this type of high-dimensional algorithm in this thesis. I have briefly discussed the reasons for this in Chapter 6.

2.2 Causal influence indices: quantitative review

One part of my quantitative results in this chapter was a reproducibility study of Lungarella *et al.* [47], using the same set of simulated systems to evaluate the performance of all causal influence indices described above. In addition the indices included in [47], which were NLGC (Equation 2.1), EGC (Equation 2.2), PI (Equation 2.3), TE using histogram binning (Equation 2.7) and SI (Equation 2.13), I included four additional approaches not considered in [47], which were TE using KSG (Equation 2.8), ETE (Equation 2.11), CTIR (Equation 2.12) and CCM (Equation 2.14). I also provided theoretical results for information-theoretic indices in one of the simulated systems, and included additional comparisons of the level of agreement between the causal influence indices. The four simulated systems, which I have summarised in Table 2.2 and defined in Equations 2.15-2.18, are widely studied in chaos theory [89] and and have previously been used in other studies relating to causal influence in time-series, e.g. [32]. Each simulated system had either one or two 'coupling strength' parameters that determined the strength of the causal relationships. For simplicity, I chose to repeat the same hyperparameters as in [47] for the causal influence indices and for the simulated systems. These are summarised in Tables 2.1, 2.2 and A.1. Some example transients are provided in Appendix A.1, though the relationships between variables are not easy to interpret visually. There are clearly questions about optimal or consistent hyperparameter choices for all of these methods, but this is a challenging computational task that was beyond the scope of my work in this chapter. I discussed differences between my results and those in [47] at the end of Section 2.2.1. The main takeaway of this was that I managed to reproduce most, but not all, of these results. The second half of this work was a sensitivity analysis, in which I 'corrupted' the data of one simulated system in order to mimic the effect of several common issues that are found in real-world data. This involved evaluating the impact of each of the following on the value of all causal influence indices: data availability, data scaling, rounding or precision error, missingness and noisy data.

Simulation	Coupling	Dynamics	Time-series length
Linear process	$X \leftarrow Y$	Linear, stochastic	$T = 10^4$
Ulam lattice	$X \rightarrow Y$	Nonlinear, deterministic	$T = 10^3, 10^5$
Unidirectional Hénon	$X \rightarrow Y$	Nonlinear, deterministic	$T = 10^3, 10^4, 10^5$
Bidirectional Hénon	$X \leftrightarrow Y$	Nonlinear, deterministic	$T = 10^4$

Simulation	Simulation model parameters	Coupling strength
Linear process	$b_x = 0.8,\ b_y = 0.4,\ \sigma_x^2 = \sigma_y^2 = 0.2$	$\lambda \in [0, 1]$
Ulam lattice	$N_L = 100$ (lattice size)	$\lambda \in [0, 1]$
Unidirectional Hénon	$a = 1.4,\ b_x = 0.3,\ b_y = 0.3$	$\lambda \in [0, 1]$
Bidirectional Hénon (I)	$a = 1.4,\ b_x = 0.3,\ b_y = 0.3$	$\lambda_x, \lambda_y \in [0, 0.4]$
Bidirectional Hénon (NI)	$a = 1.4,\ b_x = 0.3,\ b_y = 0.1$	$\lambda_x, \lambda_y \in [0, 0.4]$

Table 2.2: Brief summary of each simulated system and their hyperparameters. The simulated systems are defined in Equations 2.15-2.18. The difference between identical (I) and non-identical (NI) bidirectional coupled Hénon maps is the value of coupling parameter b_y ($b_y = b_x$ for identical maps and $b_y < b_x$ for non-identical maps). Each simulation was initialised randomly and the first 10^5 iterations were discarded (10^4 for linear process). The 'coupling strength' parameters λ were incremented by 0.01 in all cases.

2.2.1 Reproducibility study

Linear process. The simplest simulation system was a bivariate linear process (LP) with intrinsic Gaussian noise. The system is defined as:

$$\begin{aligned} x_{t+1} &= b_x x_t + \lambda y_t + \epsilon_{x,t}, \qquad & \epsilon_{x,t} &\sim N(0, \sigma_x^2) \\ y_{t+1} &= b_y y_t + \epsilon_{y,t}, & \epsilon_{y,t} &\sim N(0, \sigma_y^2) \end{aligned} \tag{2.15}$$

All indices showed increasing causal influence in the $Y \rightarrow X$ direction as the coupling parameter λ was increased (Figure 2.3). However, there were conflicting results in the $X \rightarrow Y$ direction, where there should be no causal influence regardless of λ. In particular, TE (H) and CTIR decreased with λ, while the cross mapped indices (SI and CCM) increased with λ. TE (H) and SI were non-zero in the $X \rightarrow Y$ direction for all values of λ. As each x_t or y_t is a sum of Gaussian variables, theoretical values can be calculated for all information-theoretic measures, which I derived in Appendix A.1. For the transfer entropy, these are:

$$\begin{aligned} \mathrm{TE}_{X \rightarrow Y} &= 0 \\ \mathrm{TE}_{Y \rightarrow X} &= \frac{1}{2} \log \frac{\sigma_x^4 (1 - b_y^2)(1 - b_x b_y)^2 + 2\lambda^2 \sigma_x^2 \sigma_y^2 (1 - b_x b_y) + \lambda^4 \sigma_y^4}{\sigma_x^4 (1 - b_y^2)(1 - b_x b_y)^2 + \lambda^2 \sigma_x^2 \sigma_y^2 (1 - b_x^2 b_y^2)} \end{aligned}$$

I included the theoretical values for CTIR in Figure 2.3, but but did not derive the full expression, since the algebra is significantly more complicated. TE (KSG) reliably estimated the 'true' transfer entropy but TE (H) underestimated for $Y \rightarrow X$ and overestimated for

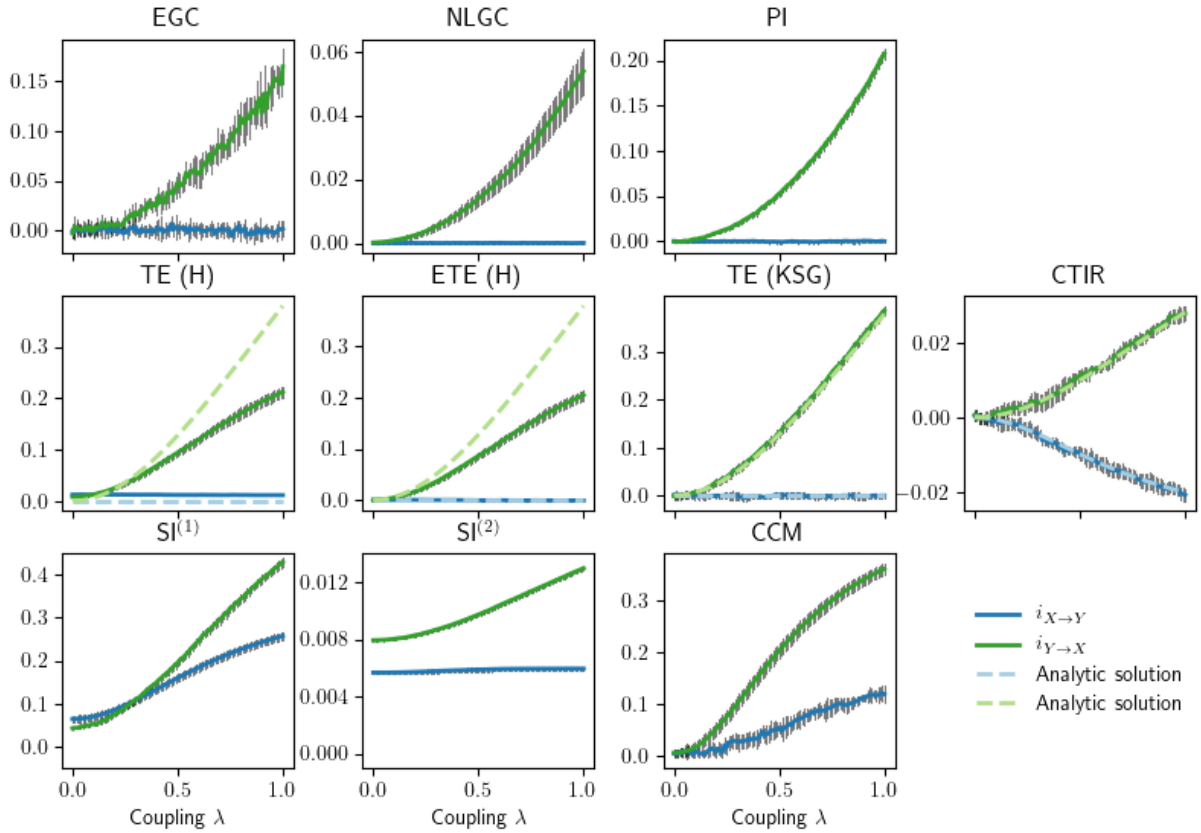

Figure 2.3: Linear Gaussian process (LP) simulation results. The time-series had length $T = 10^4$ and unidirectional ($Y \to X$) coupling. Error bars are for one standard deviation from the mean values, after 10 independent simulations.

$X \to Y$. This is a fundamental flaw of the latter and undermines advantageous properties it has over the KSG estimate. Increasing the time-series length T (not shown) did improve the TE (H) estimates for this system, while TE (KSG) remained accurate. However, this trend was not consistent across all simulated systems, with TE (H) generally more robust to increasing time-series length than TE (KSG). The estimated CTIR values also matched the theoretical values, but did not accurately reflect the causal structure of the system.

Ulam lattice. The Ulam lattice (UL) is a deterministic, nonlinear system that chains together multiple unidirectional Ulam maps:

$$\begin{aligned} s_{t+1,l+1} &= f(\lambda s_{t,l} + (1-\lambda)s_{t,l+1}), \quad l = 1, \ldots, N_L - 1 \\ s_{t+1,1} &= f(\lambda s_{t,N_L} + (1-\lambda)s_{t,1}), \quad f(s) = 2 - s^2 \\ x_t &= s_{t,1}, \quad y_t = s_{t,2} \end{aligned} \tag{2.16}$$

As $N_L \to \infty$, the causal influence from Y to X becomes negligible, so the system should have unidirectional coupling in $X \to Y$ direction. The system converges to a two-state limit cycle when $\lambda \approx 0.18$ (i.e. an attractor manifold containing two values that both X

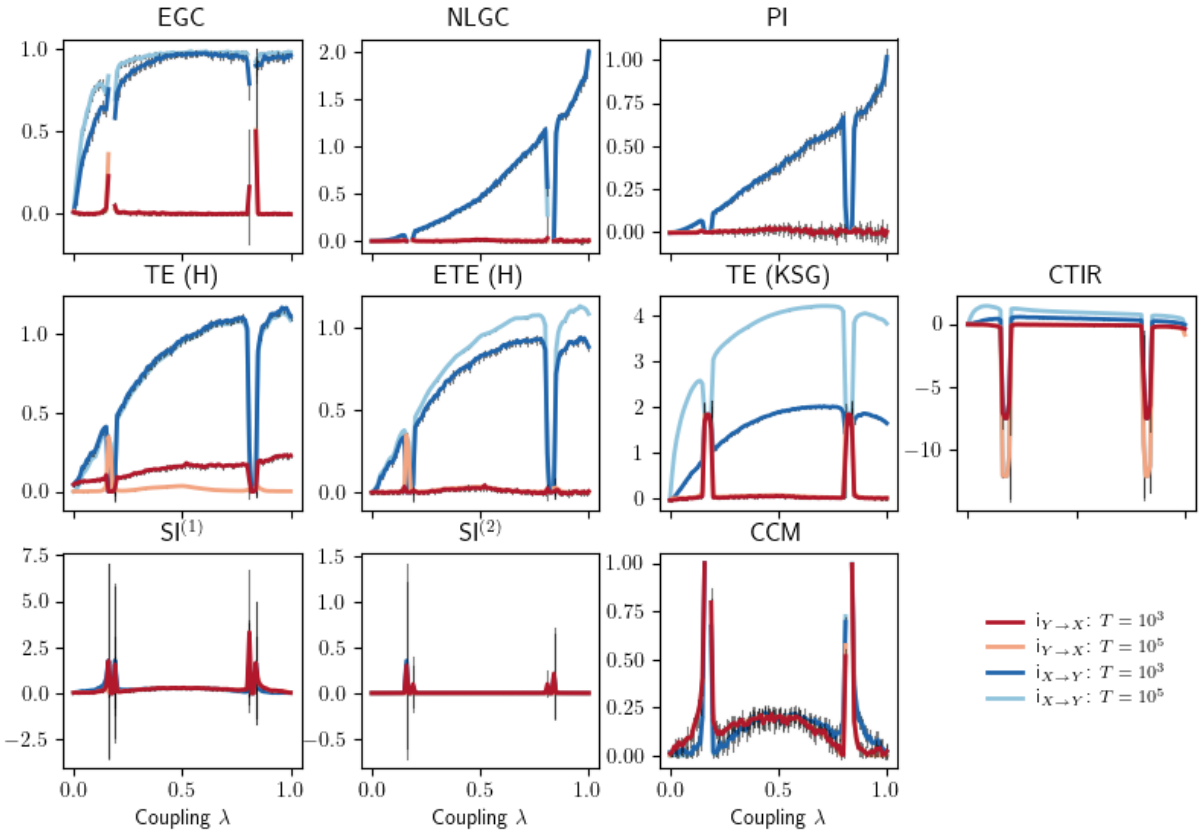

Figure 2.4: Ulam lattice (UL) simulation results. This has unidirectional $X \to Y$ coupling, and was repeated for two time-series lengths, $T = 10^3$ and $T = 10^5$. Error bars are for one standard deviation from the mean values, after 10 independent simulations.

and Y alternate between), and to a fixed point for $\lambda \approx 0.82$. In both cases, cause and effect are indistinguishable and, as a result, most indices either have values approximately equal to zero or suffer from numerical instability with extremely high variance. Outside of these coupling parameter values, the information-theoretic methods and regression-based indices showed reasonable consistency (Figure 2.5). The exception to this was CTIR, which slowly decreased as λ increased, though it still correctly identified the direction of information flow. As was the case for the LP simulated system, ETE (H) successfully corrected for finite sample effects, which gave rise to spurious positive TE (H) results when $T = 10^3$. The cross mapped indices struggled to separate synchronisation from directed causal influence in this system. Both similarity indices failed to identify causal structure and CCM misidentified the direction of causal influence for $\lambda < 0.5$.

Unidirectional coupled Hénon maps. Hénon maps are deterministic systems that are functions of two past states, rather than just one. In this instance, X is governed a classical Hénon map, a chaotic system that involves three steps (folding, contraction and

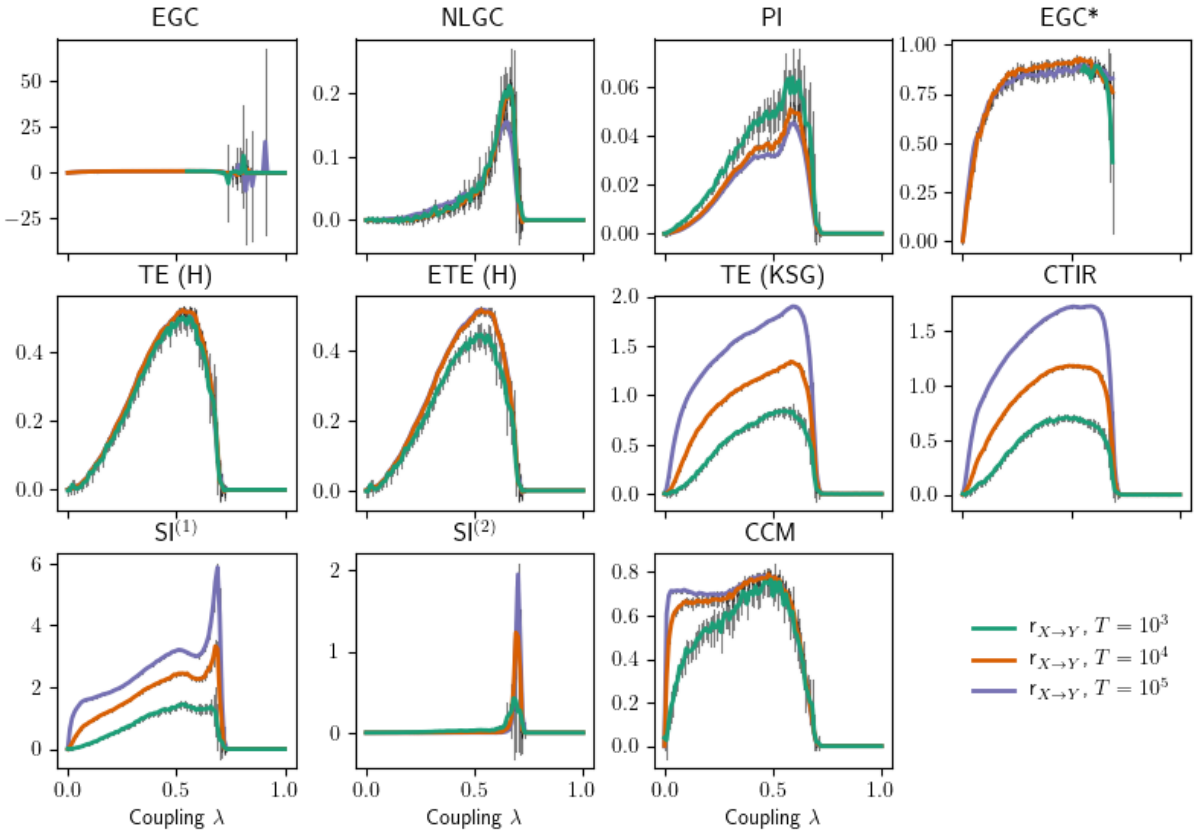

Figure 2.5: Unidirectional coupled Hénon map (HU) simulation results. This shows the net directed index $r_{X\to Y} = (i_{X\to Y} - i_{Y\to X})$ for unidirectional $X \to Y$ coupling, and was repeated for three time-series lengths, $T = 10^3$, 10^4 and 10^5. Error bars are for one standard deviation from the mean values, after 10 independent simulations. Due to extreme results in EGC when has a finite limit cycle (i.e. $\lambda > 0.7$), I set these values to NaN and repeated the subfigure (EGC*).

reflection), while Y is adjusted by a coupling signal proportional to $y_{t+1}(y_{t+1} - x_{t+1})$:

$$\begin{aligned} x_{t+2} &= a - x_{t+1}^2 + b_x x_t \\ y_{t+2} &= a - y_{t+1}^2 + b_y y_t + \lambda y_{t+1}(y_{t+1} - x_{t+1}) \end{aligned} \tag{2.17}$$

For unidirectional coupled Hénon maps (HU), I reported only the net directed index $r_{X\to Y} = (i_{X\to Y} - _{Y\to X})$. The HU system is highly synchronised with a finite limit cycle for $\lambda \in [0.7, 1]$. All causal influence indices were mostly consistent, without the noisy fluctuations observed in [47]. The exception to this was the instability in EGC when $\lambda > 0.7$ (Figure 2.5). It is also notable that the values of approximately half of the indices (TE (KSG), CTIR, SI, CCM) increased significantly as the time-series length T increased in magnitude, in some instances tripling in value. The remaining indices (TE (H), ETE (H), EGC, NLGC, PI) were reasonably consistent as T increased.

Bidirectional coupled Hénon maps. In the bidirectional coupled Hénon maps (HB), coupling signals are involved in both directions:

$$x_{t+2} = a - x_{t+1}^2 + b_x x_t + \lambda_x (x_{t+1} + y_{t+1})(x_{t+1} - y_{t+1}) \tag{2.18}$$
$$y_{t+2} = a - y_{t+1}^2 + b_y y_t + \lambda_y (x_{t+1} + y_{t+1})(y_{t+1} - x_{t+1})$$

As for the HU system, I reported only the net directed index $r_{X \to Y} = (i_{X \to Y} - i_{Y \to X})$ (Figure 2.6). There was generally a high level of similarity between all methods for the bidirectional coupled Hénon maps (relatively, rather than in absolute magnitude). The exceptions to this in the identical HB(I) system were NLGC and CCM. In the case of NLGC, I found that increasing the number of RBFs resulted in values that were more consistent with other indices (not shown). For CCM, the direction of causality was inconsistent and often incorrect. The reason for this was unclear, though it may also have been related to hyperparameter choices. As HB(I) is symmetrical in X and Y, there should be mirrored symmetry in the line $\lambda_x = \lambda_y$, which was observed as expected. In the region approximately equal to $\{(\lambda_x, \lambda_y) : \lambda_y + \lambda_x > 0.28\}$, the system converges to a plane in (x, y)-space with $x_t = y_t$, though without any periodic orbit. Since $x_t = y_t$, $i_{X \to Y} = i_{Y \to X}$ and $r_{X \to Y} = 0$. There were a small number of points in which zero values or numerical instabilities were present in all indices, but these were all instances of the system converging to finite limit cycles.

The differences in results between causal influence indices were more pronounced in the non-identical HB(NI) simulations (Figure 2.7). There is another region, approximately equal to $\{(\lambda_x, \lambda_y) : 0.1 < \lambda_x < 0.28,\ 0.05 < \lambda_y < 0.15\}$, in which the system converges to plane in (x, y)-space with $x = y$. There are also isolated points in which the system converges to finite limit cycles, as in the HB(I) system. As in previous cases, EGC had numerical instabilities at these coupling parameters, whilst the CCM and NLGC returned NaN values. The information-theoretic indices were broadly in agreement with each other and with EGC, PI and $\mathrm{SI}^{(1)}$. In contrast, NLGC was negative almost everywhere (and remained so in a repeat analysis with an increased number of RBF kernels), while both $\mathrm{SI}^{(2)}$ and CCM were largely non-negative. However, the regions with the most extreme values occurred in different places in each of these three indices.

Comparison between approaches In Figure 2.8, I assessed the degree to which there was agreement between the causal influence indices, by computing the correlation between them as the coupling parameters λ were varied. For LP, there was almost universal agreement in the $Y \to X$ direction. This was not the case in the $X \to Y$ direction. In this instance, since the causal influence value should have been approximately zero for all λ values (but with some random variability around zero), the off-diagonal correlations should also be about zero. However, as noted previously, the estimated causal influence

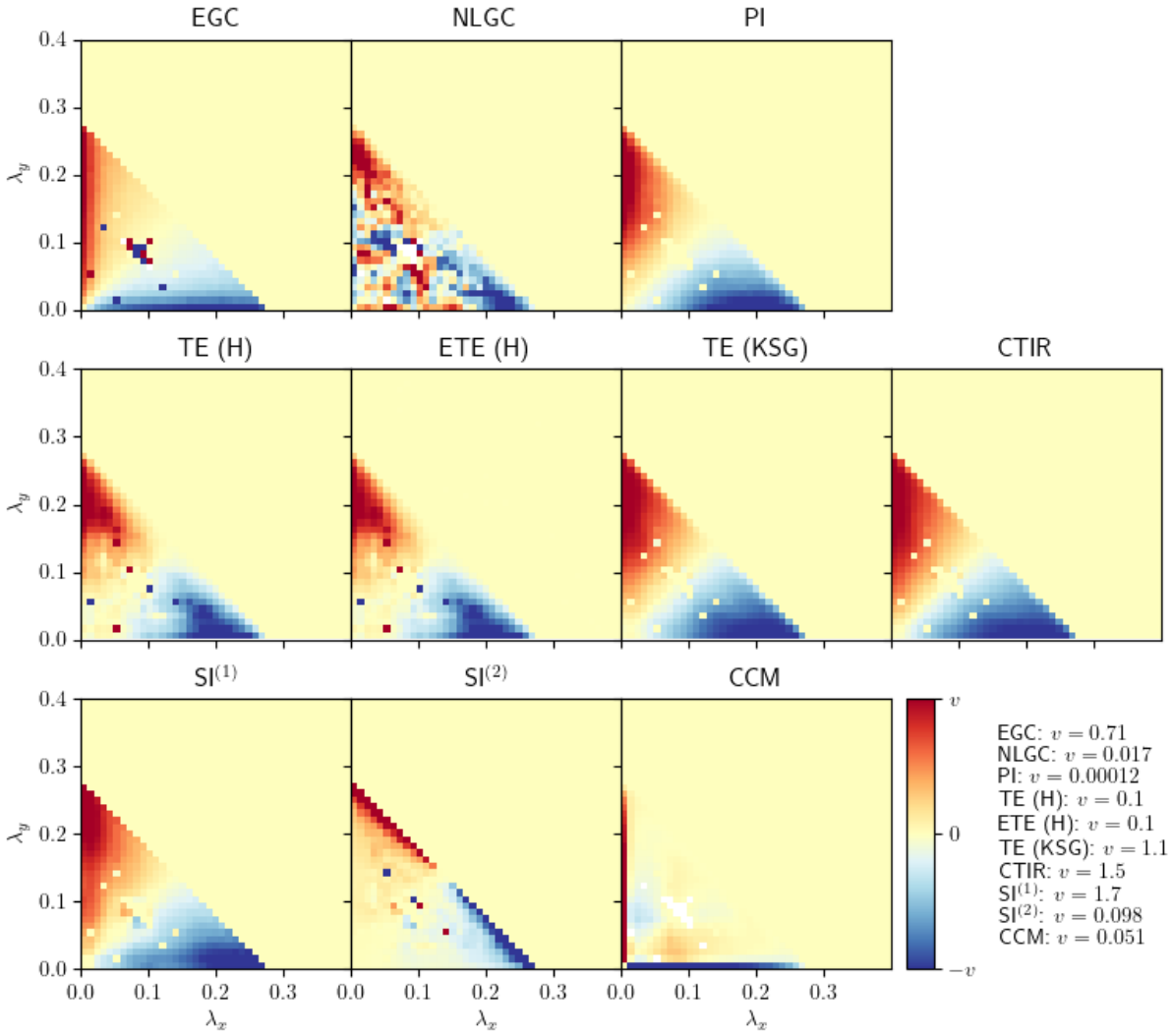

Figure 2.6: Identical bidirectional coupled Hénon map (HB(I)) simulation results. This shows the net directed index $r_{X \to Y} = (i_{X \to Y} - i_{Y \to X})$ averaged across 10 independent simulations, for bidirectional $X \leftrightarrow Y$ coupling and time-series length $T = 10^4$. Limits v were a function of data percentiles, i.e. $v = \max(-p_1, p_{99})$, where p_i was the i^{th} percentile.

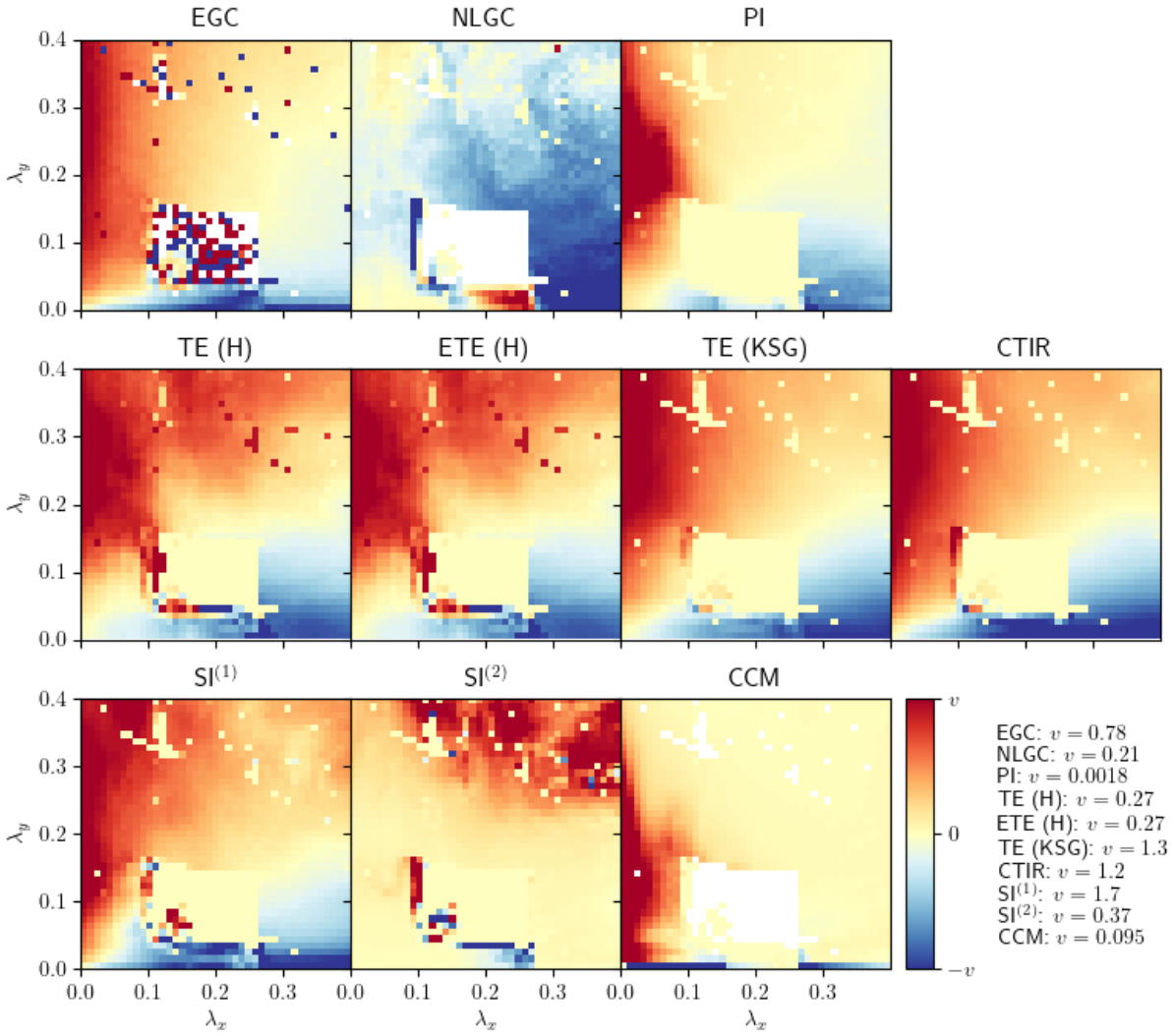

Figure 2.7: Non-identical bidirectional coupled Hénon map (HB(NI)) simulation results. This shows the net directed index $r_{X\to Y} = (i_{X\to Y} - i_{Y\to X})$ averaged across 10 independent simulations, for bidirectional $X \leftrightarrow Y$ coupling and time-series length $T = 10^4$. Limits v were a function of data percentiles, i.e. $v = \max(-p_5, p_{95})$, where p_i was the i^{th} percentile.

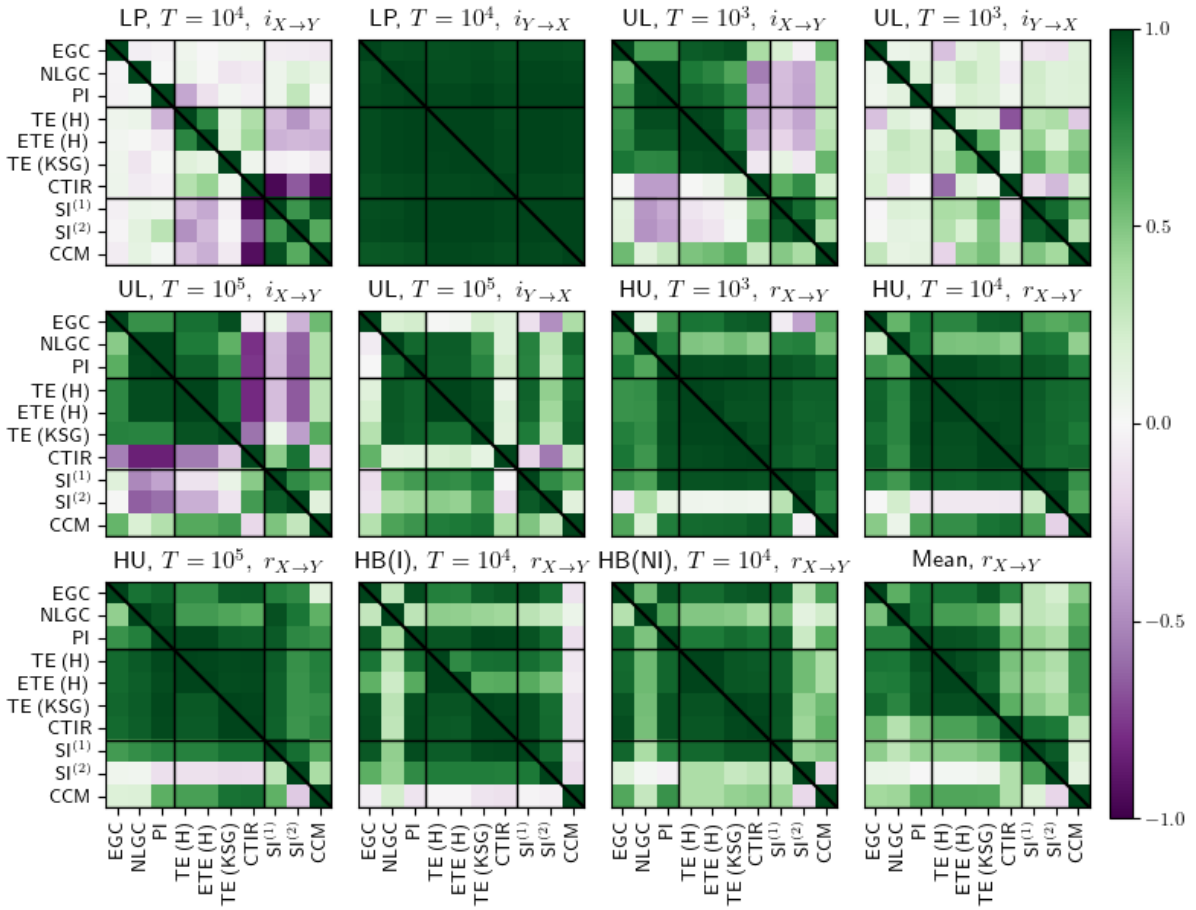

Figure 2.8: Correlations between each of the causal influence indices for all simulations: linear process (LP), Ulam lattice (UL) and unidirectional coupled Hénon maps (HU), bidirectional coupled Hénon maps (HB(I) and HB(NI)). For UL and HU, simulations were repeated with multiple time-series lengths T. In each subplot, the lower left half below the diagonal shows the Pearson correlation between pairs of indices (across all runs and values of λ) and the upper right half shows the rank-based Spearman correlation. Both directed indices $i_{X\to Y}$ and $i_{Y\to X}$ were calculated for LP and UL simulations but only the net directed indices $r_{X\to Y}$ for HU and HB. The final bottom right subfigure contains the mean correlation in $r_{X\to Y}$ across all simulations, weighting each previous subfigure equally.

erroneously increased with increasing λ for some indices and decreased with increasing λ for others, which led to positive and negative correlations between indices. For UL, there were mixed results in the $X \to Y$ direction in terms of correlations, even though there was clear causal structure in the system. In particular, CTIR and SI both decreased in value as the coupling strength was increased. In the $Y \to X$ direction, there was generally positive correlation between indices, despite a negligible causal relationship in this direction. This was likely due a very slight peak in value for most indices at approximately $\lambda = 0.5$, the reasons for which were unclear. As noted, there were signficant correlations between methods for the coupled Hénon maps, with some exceptions (such as CCM in HB(I)).

Comparison to the original publication. I was able to reproduce the findings in [47] in most cases, but there were also some minor differences between their results and mine. I followed the implementation in [47] as closely as possible and it was generally unclear why these differences occurred. In particular, I sometimes found results of a similar profile in λ but different magnitude. I observed this for most notably for the following: EGC and LP, PI and all simulated systems, $\mathrm{SI}^{(1)}$ and UL, TE and HU. My results were largely comparably in magnitude for HB maps, though I handled numerical outliers differently in visualisation.

One possible explanation for these discrepancies was an occasional lack of detail in [47], e.g. no mention of data standardisation. The results I have reported did not involve any preprocessing and when I repeated experiments (not shown) with some preprocessing, this did not rectify these differences. In one notable difference between their results and mine, they found a region in λ-space (namely $\{(\lambda_x, \lambda_y) : \lambda_x > 0.1,\ \lambda_y > 0.1,\ \lambda_y + \lambda_x < 0.35\}$) in which they identified general synchronisation between X and Y that resulted in numerical instabilities and a difficulty in estimating the causal influence indices, but I did not observe this behaviour.

I made only one deliberate change in implementation, which was to use k-means clustering to select RBF centres in NLGC, rather than fuzzy c-means clustering. The rationale behind this change was that k-means gave similar or improved results with a much reduced computational cost. Lungarella *et al.* noted that NLGC was numerically unstable for 'small' T and computationally expensive for 'large' T, which I suggest may be partly due to their use of fuzzy c-means giving 'poor quality' clustering, perhaps because of early-stopping criteria in a computationally inefficient algorithm. I also found that the performance of NLGC in HB simulations improved significantly with a different set of NLGC hyperparameters, e.g. $P = 50$ instead of $P = 10$ (not shown).

Computational costs. One consideration that may be relevant to identifying a suitable method is a trade-off between robust performance and computational efficiency. This is particularly true if an algorithm needs be run repeatedly over a large number of variables or if there is a large quantity of data. In practice, the most significant factor in this trade-off was the time-series length T. Table A.2 shows the time taken to estimate the causal influence indices for each simulated system. TE (H)/ETE (H) was the fastest in almost all cases, followed by CCM, EGC and TE (KSG).

2.2.2 Sensitivity analysis

Next, I investigated the sensitivity of all causal influence indices to several types of data modifications that mirrored issues often arising in real-world data. I chose to illustrate the effects of these data transformations using UL with $T = 10^3$, and to naïvely keep all

hyperparameters the same, even when this was clearly ill-advised (such as when the data was scaled). This was in part to highlight that users should consider data preprocessing or hyperparameter selection before estimation, because failure to do so can impact results and interpretation. In Table 2.3, I summarised how the means and standard deviations of the directed indices $r_{X\to Y}$ (averaged over all λ values) were impacted by these data modifications.

Data availability. Many of the indices remained consistent with increasing time-series length T, but had smaller variance (Figure 2.4). The exceptions to this were TE (KSG) and CTIR, which had large increases in the value as T increased. This is something that should at least be acknowledged in any implementation of these two indices. These results reinforced similar observations in HU maps (Figure 2.5). It remained unclear whether the values would converge as the time-series length continued to increase, or whether both are unbounded as $T\to\infty$, but some initial empirical evidence did not support the former (not shown). Though $r_{X\to Y}$ values from both transfer entropy methods were highly correlated, they should in theory be equal, since they are both estimates of the same quantity. It is difficult to reconcile the different magnitudes, particularly as I previously observed underestimation in TE (H) for LP simulations.

In the later work with physiological signals from intensive care, I initially considered even shorter lengths than previously investigated in this section, e.g. 2 hours of minute-by-minute data or $T = 120$. Up until this point, the analysis of causal influence indices was for time-series of length at least $T = 10^3$. Therefore, I decided to revisit this analysis with $T = 100$, for both the linear process and Ulam lattice experiments (Figure A.4). The results for the Ulam lattice were generally noisier for $T = 100$ than for $T = 10^3$ but were still clear. The results for the linear process were worse for $T = 100$, with huge standard deviations across multiple independent simulations. As a result, I decided to increase the length of time-series in the anaylsis of physiological signals, and instead focused on 24hrs of minute-by-minute data.

Data scaling and standardisation. The second set of tests was split into three. First, I standardised both series by the sample means and standard deviations, and then I separately scaled each (un-standardised) series by a factor of 10 (Figure A.2). For the Ulam lattice system, sample means for both X and Y were typically between 0.4 and 0.7 and standard deviations were both approximately equal to 1.2 (except when the system converged to a two-cycle or fixed point). Several methods are invariant under linear scaling or shifting of the original time-series, including cross mapped indices. Information-theoretic measures are also invariant in theory, but the KSG algorithm is not, since it is based on k-nearest neighbour distances. EGC relies on a neighbourhood size parameter, and mismatched scaling of the time-series without suitably adjusting this parameter can result

Method	Baseline	Data size	Data scaling			Rounding			Missingness		Gaussian noise		
	$T = 10^3$	$T = 10^5$	Stand.	$\times 10$	$\times 10$	1 d.p.	1 d.p.	2 d.p.	10%	20%	$\sigma_a^2 = 0.1$	$\sigma_a^2 = 1$	$\sigma_a^2 = 1$
Variable(s)			X, Y	X	Y	X	Y	X, Y	X, Y	X, Y	X, Y	X	Y
Mean	$\frac{1}{n_\lambda}\sum_\lambda \mu_\lambda$	$\sum_\lambda(\mu_\lambda - \mu_\lambda^{(a)})/\sum_\lambda \lvert\mu_\lambda\rvert$											
EGC	0.840	-0.064	0.036	0.207	-0.071	0.031	0.112	0.004	-0.027	-0.040	0.533	0.981	0.950
NLGC	0.610	0.013	0.299	2.905	-101.849	0.000	-0.003	0.001	-0.008	-0.020	0.031	0.740	-0.007
PI	0.380	0.002	0.293	1.576	-98.727	0.950	-0.951	-0.016	0.001	0.005	0.011	0.617	0.019
TE (H)	0.675	-0.158	0.000	0.000	0.000	0.011	0.030	0.000	0.071	0.159	0.026	0.786	0.731
ETE (H)	0.674	-0.158	0.000	0.000	0.000	0.014	0.026	0.000	0.075	0.155	0.026	0.748	0.774
TE (KSG)	1.509	-1.348	0.000	0.269	0.609	0.095	-0.134	-0.029	0.111	0.216	0.306	0.841	0.863
CTIR	0.462	-1.226	0.000	0.128	0.713	0.299	-0.355	-0.026	0.083	0.161	0.273	0.826	0.848
$SI^{(1)}$	0.001	0.015	0.000	0.000	0.000	0.549	-0.556	0.005	-0.013	0.018	-0.007	-0.061	0.066
$SI^{(2)}$	0.000	0.029	0.000	0.000	0.000	-3.105	3.121	-0.001	-0.037	0.017	0.000	-8.256	7.736
CCM	0.001	0.031	0.000	0.000	0.000	-1.249	1.289	-0.005	-0.009	-0.025	0.013	0.151	-0.075
Std.	$\frac{1}{n_\lambda}\sum_\lambda \sigma_\lambda$	$\sum_\lambda \sigma_\lambda^{(a)}/\sum_\lambda \sigma_\lambda$											
EGC	0.021	0.660	1.004	1.425	0.691	0.959	0.946	0.970	1.023	1.033	1.025	0.473	0.598
NLGC	0.023	0.089	0.608	64.469	126.734	1.000	0.954	0.994	1.353	1.906	1.023	1.345	2.325
PI	0.032	0.094	0.681	69.340	66.364	0.918	1.051	1.001	1.214	1.511	1.007	2.041	2.120
TE (H)	0.019	0.085	1.000	1.000	1.000	1.004	0.994	1.013	1.313	1.633	1.112	1.015	0.920
ETE (H)	0.019	0.083	1.000	1.000	1.000	0.992	0.994	1.009	1.296	1.634	1.109	0.947	0.854
TE (KSG)	0.025	0.120	1.071	0.869	1.026	1.232	1.320	1.042	1.197	1.430	1.028	1.017	0.962
CTIR	0.014	0.110	1.016	0.898	0.920	1.159	1.209	1.042	1.076	1.258	0.958	0.942	0.872
$SI^{(1)}$	0.029	0.097	1.000	1.000	1.000	1.363	1.355	1.053	1.084	1.219	0.895	0.705	0.693
$SI^{(2)}$	0.000	0.000	1.000	1.000	1.000	1.394	1.399	1.074	1.666	2.630	0.968	2.334	2.018
CCM	0.047	0.103	1.000	1.000	1.000	1.115	1.105	0.740	1.090	1.250	1.010	0.944	0.959

Table 2.3: Summary of all results from tests involving the effects of data size/availability, data scaling, rounding or precision error, missingness and Gaussian noise. I used UL with $T = 10^3$ as a baseline and calculated the mean μ_λ and standard deviation σ_λ of the net directed $r_{X\to Y}$ at each λ, over 10 independent experimental runs. I then computed the average means $1/n_\lambda \sum_\lambda \mu_\lambda$ and standard deviations $1/n_\lambda \sum_\lambda \sigma_\lambda$, over all λ, excluding only λ values where the system converged to a finite limit cycle. For each modification test, I repeated these calculations for $1/n_\lambda \sum_\lambda \mu_\lambda{}^{(a)}$, and $1/n_\lambda \sum_\lambda \sigma_\lambda{}^{(a)}$, and computed their deviation from the baseline averaged mean (difference in value, divided by absolute value of the baseline) and averaged standard deviation (ratio). If a modified simulated system returned the same values as the baseline, then these deviations should equal zero and one respectively. For each modification test, I highlighted the causal influence index that was closest to these values. For the increased data size, the averaged standard deviation should instead be much reduced, so I highlighted the value closest to 0 instead.

in an insufficient number of points available for the locally linear autoregression. This was observed when either series was scaled by 10. The net directed index for both NLGC and PI had vastly inflated magnitude when Y was scaled by 10. I recommended in [90] that both series should be independently standardised or normalised before causal influence estimation.

Rounding and precision error. Real-world data often comes from measuring instruments with a fixed measurement precision or are reported with some rounding error. I performed three tests to investigate rounding error, first rounding each series separately to 1 decimal place and then simulteneously rounding both to 2 decimal places (Figure A.2). TE, EGC and NLGC had similar performance to the baseline in all cases, whilst CCM suffered the most. There were some practical implementation issues that relate to rounding error, particularly edge-cases in nearest neighbour approaches. Theoretical work motivating estimation algorithms often makes assumptions about these edge-cases or neglects them entirely. For example, one typical assumption is that the nearest neighbour distances are unique, which may not hold if data is rounded. A solution to this, as suggested in [69], is to add low-amplitude random noise to the data before estimating causal relationships, but the impact of this on the bias and variance of causal influence estimation is still unclear.

Missingness. In two experiments, all causal influence indices appeared robust to random missingness of 10% and 20% (Figure A.3). While some implementations of causal influence indices, e.g. [91, 92], are unable to handle missing data (or NaN values) there is no theoretical reason that any of the causal influence indices should be unable to estimate the value when there is some amount of missing data, since the univariate embedding vectors should simply be discarded for both series if either one contains NaN values. Missing data has some similarity to reduced length time-series and there is some level of missingness that ought to invalidate results and interpretation, but it is not currently clear what that is. There is one important subtlety between missingness and data quantity, which is that the embedding step must be performed before excluding NaN values and only afterwards should any embedding vectors containing NaNs be discarded. For example, if $X = (1, 3, \text{NaN}, 0, 3, 2)$, the embedding vectors with $m = 1$ are $((1, 3), (3, \text{NaN}), (\text{NaN}, 0), (0, 3), (3, 2))$ and $\boldsymbol{X} = ((1, 3), (0, 3), (3, 2))$. If a window of $m = 1$ either side of the missing data was removed prior to embedding, then the embedding vectors $(1, 3)$ and $(0, 3)$ would be incorrectly excluded. Alternatively, if the missing data was initially discarded, truncating the series to $X = (1, 3, 0, 3, 2)$, then an extra embedding vector $\boldsymbol{x}_t = (3, 0)$ would be erroneously included. This additional embedding vector would only be valid under imputation assumptions.

Noisy data. In the earlier LP simulations, Gaussian noise was an intrinsic component of the system and theoretical expression showed that this impacted the value of TE only through the ratio of variances σ_x/σ_y. However, this noise is internal to the simulation process and does not arise in observation. In the UL tests, I added additional Gaussian noise after the simulation was completed. The inclusion of this 'measurement noise' does not alter the state of the system or the causal influence between variables, but it may obscure the causal structure. In the first of these tests, in which I added small variance Gaussian noise to both variables ($\sigma_a = 0.1$), the amplitude of the noise was an order of magnitude less than the values of the time-series, and the inclusion of this noise had only a small effect for all indices (Figure A.3). In the latter tests, I added noise with much higher variance to each time-series in turn (with $\sigma_a = 1$) and the effect of this was more pronounced. NLGC performed best and appeared very resilient to noise added to Y (effect variable), though it dropped slightly in value when noise was added to X (cause variable).

2.2.3 Discussion and recommendations

In-depth comparative studies of this kind are relatively rare in the literature, particularly methods for describing a mathematical concept that does not have a consistent mathematical definition, e.g. estimating causal influence in time-series. Even without a universal definition, causality has huge importance in how we can model, predict and exploit real-world applications from many scientific disciplines. Understanding asymmetric causal influence in bivariate systems is an important step towards providing insight into causal interactions between components in complex temporal networks.

Most causal influence indices have strengths and weaknesses, and there did not appear to be one method whose all-round performance exceeded all others. Granger causality and transfer entropy have long been regarded as the leading methods for bivariate or low-dimensional multivariate systems and these have had wide applications [33, 34, 39]. Transfer entropy has the distinct advantage that it is built upon a well-established and universal framework of information-theoretic principles (i.e. Shannon entropy). It performed solidly throughout, though there was some tension between algorithms for estimating TE. I showed that a fixed partition approach using histogram binning (TE (H)) was biased in the simplest model, despite this having better computationally efficiency and more consistency as time-series length varied. The KSG algorithm appears to be a better option, unless data is extremely scarce. However, there are some unanswered concerns about TE (KSG), particularly that it sometimes increased in magnitude as the time-series length was increased. CTIR did not seem to offer any obvious advantage to compensate for a much higher computational cost. Standard linear Granger causality is widely favoured but has restrictive assumptions and is ill-suited to complex nonlinear problems. Of the two nonlinear extensions to Granger causality, EGC was preferred in [47].

Some of the computational challenges and numerical instability that they experienced with NLGC may have been a result of their choice of a fuzzy c-means for RBF kernels, and alternate parameter choices appeared to resolve some of their concerns. I found that NLGC was one of the most robust methods to rounding error, missing data and Gaussian noise. Lungarella *et. al* rightly noted that "if the rank of the data is small, kernel based methods tend to overfit" [47], but I did not observe any issues with this in simulations. Predictability improvement (PI) likewise performed solidly, and had a slight advantage among regression-based indices in that it was less reliant on hyperparameter choices. Finally, cross mapped indices had mixed results in the quantitative review, but dynamical systems theory offers different insights into causal inference, particularly in high signal-to-noise settings, so should not be readily dismissed. Convergent cross mapping is a more recent and popular method, and this offered a broad improvement on the similarity indices (SI), which did not consistently identify the strength or direction of causal influence. However, CCM also did not always manage to determine the correct direction of causal influence in the simulations.

In Section 2.2.2, I highlighted the importance of some preprocessing steps to avoiding algorithmic issues, in particular data standardisation. This is also important to allow comparison of results relative to other data, since there is no clear interpretation of the absolute value for any causal influence index. Rounding error gave rise to practical issues within the implementation of several of the algorithms, particularly for indices that require k-nearest neighbour computations. Noisy data led to the largest changes in absolute value for most methods, particularly for observation noise in the causal variable. As noted in previous studies, "noise in real-world data is ubiquitous, [and] the inclusion of noise in model investigations has been largely ignored" [78]. However, provided the noise had small variance relative to the magnitude of the time-series values, all methods performed adequately.

On the basis of this work, I concluded that the strongest choices for quantifying bivariate causal relationships were, in my view, transfer entropy using the KSG algorithm, followed by nonlinear Granger causality and predictability improvement. A more complete approach would be to estimate the causal influence using multiple indices, which may strengthen conclusions if they are largely in agreement. Furthermore, new proposed methodologies should include discussion of the various real-world data issues that I have identified.

2.3 Information, intensive care and Covid-19

The motivation for the review of causal influence indices was an application to physiological time-series from ICU. In particular, I sought to test whether there was evidence of Covid-19

brainstem dysregulation, via decreased interaction between physiological subsystems. I decided to use measures from information theory to describe the interactions between three of the main physiological time-series variables (temperature, heart rate and mean arterial blood pressure). Alongside transfer entropy (Equation 2.7), I also estimated the entropy (Equation 2.4) and mutual information (Equation 2.6), in order to describe the information content of these physiological time-series more completely.

2.3.1 Dutch Data Warehouse for Covid-19

The Covid-19 pandemic placed huge burdens on ICUs, with high mortality rates and more demand for ICU beds than availability. The urgency for scientific discovery around Covid-19 necessitated large-scale multi-centre ICU data sharing, especially with uncertainty around the variation in clinical practice between centres and with more frequent patient transfers between centres. Aided by experience with previous large scale single-centre ICU databases, Amsterdam UMC led a data sharing collaboration in the Netherlands to create the Dutch Data Warehouse for Covid-19 (DDW) [16]. This contained clinical data from 3463 patients and 25 hospitals, mapped to a unified pre-defined vocabulary of 942 clinical parameters. This data totalled over 200 million datapoints, and included demographics, outcomes, medications, laboratory data and routine physiological measurements. Overall, patients in ICU had a median age of 64, median ICU length of stay of 7.1 days and 24.4% ICU mortality. Within the entire dataset, there were between 6.5 and 8 million measurements for each of the three physiological time-series variables (T, HR, ABP), averaging at around 2000 measurements per patient. All three variables were present within the database at minute-by-minute intervals. Arterial blood pressure is often recorded at the bedside at higher frequencies, but it was only available here as three summary values (systolic, mean and diastolic), averaged by minute. I used the mean ABP in this chapter and in Chapter 3.

Since the set of hospitals contributing to DDW used different electronic health record systems (Epic, HiX and MetaVision) and recorded data at different levels of granularity, not all of the patients in the full dataset had consistent simultaneous minute-by-minute measurements of all three variables. I used a simple thresholding to remove extreme values outside of pre-defined ranges for each variable. The minimum and maximum allowed values were 50mmHg and 140mmHg for ABP, 33°C and 42°C for temperature, and 50bpm and 150bpm for HR. Having removed these outliers, I identified a subset of 136 patients who were in ICU for at least 2 days, with at least 1440 measurements in each variable and at least 90% of these occurring within 1 minute of the previous measurement (i.e. at least 24hrs of minute-by-minute time-series data for each variable). The patients in this smaller Covid-19 cohort came from 4 hospitals and had a maximum length of stay of 55.5 days. 34 patients in this cohort died in ICU with median length of stay 14.1 days, while 102

patients were discharged with median length of stay 13.6 days. In my analysis, I restricted to looking at a maximum of 15 days after ICU admission. Within this window, 19 patients died in ICU, 55 were discharged and 62 were still in ICU at the end of the 15 day period.

2.3.2 Hypothesis of dysregulation of the brain stem

One of the most widely-known physiological regulation models is homeostasis, which means 'stability through invariance' (etymologically, 'similar' + 'standing still') [93–95]. In this model, systems in the body (e.g. chemical, biophysical or physiological) are maintained by regulatory mechanisms that apply negative feedback loops to reduce perturbations away from narrow, fixed homeostatic ranges. Sometimes, a homeostatic range is necessary for function, but it can also be a compromise between cost (i.e. energy requirements) and utility. There are many examples of feedback mechanisms within the human body, e.g. thermoregulation [96, 97], arterial blood pressure [98] and fluid balance [99].

An alternative regulation model is allostasis, or 'stability through variation' (etymologically, 'different' + 'standing still') [100–102]. The central idea of allostasis is that an efficient regulatory system involves constant natural variation (usually within some 'safe' variable range), which in turn allows flexible adaptation to changes in the availability of resources. Examples of this include insulin levels [103] and arterial blood pressure [104]. Compared to homeostasis, allostasis can be viewed almost as a paradigm switch from a reactive regulation model to a feedforward predictive regulation model, anticipating the demands on internal biological systems based on the predicted resource abundance or scarcity. In physiological systems, this variability can be observed in continually varying physiological time-series variables (e.g. heart rate, arterial blood pressure).

The hypothalamus and the brainstem play critical roles in regulating cardiovascular and respiratory function, and in thermoregulation. The brainstem also conveys all information from the brain to the rest of the body and vice versa, and has neural function relating to alertness, awareness and consciousness. Brainstem damage is very serious, and can lead to irreversible loss of capacity for consciousness (i.e. 'brainstem death') [105]. Anecdotal evidence from clinicians treating severely-ill Covid-19 patients, which included abnormal brainstem reflex and unresponsiveness to sensory stimuli, suggested that brainstem dysfunction was a symptom of Covid-19. This resulted in a clinical hypothesis of brainstem dysfunction in patients with severe illness induced by respiratory viruses, which has since been supported in multiple studies [24, 25]. Since the brainstem controls some physiological regulation, brainstem dysregulation may lead to reduced interactions between physiological systems. The idea that severe disease disrupts interactions between different systems within the body (i.e. 'decomplexification') follows from research during the 1980s and 1990s, which focused on measures of heart-rate variability as a proxy for the interaction between heart and brain [106–108]. In fact, several measures of system

complexity, including approximate entropy [109], were developed around this time because of their application to cardiology. Causal influence between time-series can therefore provide insights into these interactions, though this has not yet been done in the context of severe respiratory viral disease. The fundamental idea in allostasis is that there is some natural variation, but this variation should remain largely consistent over time. As such, there are also parallels between this regulation model and the concepts from information theory, as both can be viewed in terms of 'expected surprise' and 'average uncertainty'. This meant that causal inference and information theory provided a natural framework for investigating the brainstem dysfunction hypothesis under the allostasis model and through the lens of physiological time-series.

2.3.3 Information theory and physiological time-series

For the Covid-19 cohort defined in Section 2.3.1, I rounded each variable to a uniform precision, in order to ensure consistency between patients (to the nearest integer for HR and ABP, and to 1 decimal place for temperature). I then added infinitesimally-small uniform noise to each variable (i.e. between -5×10^{-9} and 5×10^{-9}) to ensure distances between embedding vectors during causal influence estimation were unique, and standardised each variable separately by its mean and standard deviation. I estimated the entropy individually for each variable (T, HR, ABP), and the mutual information and (bidirectional) transfer entropy for each pair of variables (T and HR, T and ABP, HR and ABP). I used the KSG algorithm for estimation (Equations 2.8, 2.9 and 2.10), with hyperparameters $m = 1$, $\tau = 1$ and $k = 4$.

For each information-theoretic measure, these calculations were for a time window of length 24hrs, containing up to 1440 measurements. I repeated this at 6hr increments from ICU admission until discharge, death or the end of the 15^{th} day after admission. By shifting the 24hr window and estimating each information-theoretic measure at 6hr increments over a maximum of 15 days, I created multivariate trajectories of information-theoretic measures (which I refer to as 'information trajectories') for each patient during their ICU stay. These consisted of 12 values (3 entropy values, 3 mutual information values and 6 transfer entropy values) at every successive 6hr timestamp, with a maximum of 60 multivariate observations per patient. My implementation of the KSG algorithm was able to handle missing data, but I discarded windows in which there was a significant amount of missingness in the physiological variables. For the differential entropy estimation, I required there to be at least 200 valid measurements in the time-series. For mutual information and transfer entropy, I required there to be at least 200 simultaneous valid measurements in both variables. In practice, this meant that within each 24hr window, there were 1343 raw measurements on average, rather than 1440.

In the remainder of this chapter, I sought to illustrate what the information-theoretic

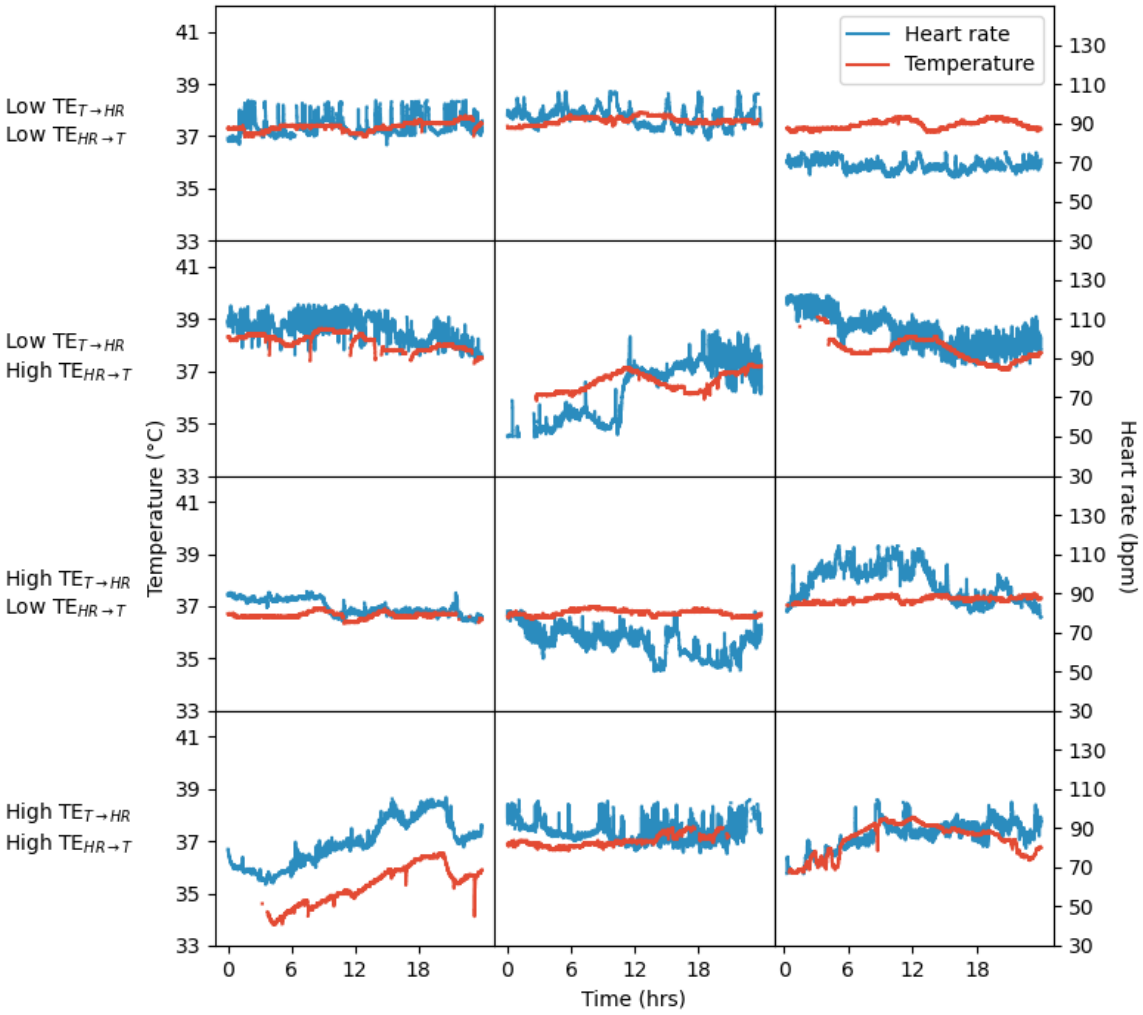

Figure 2.9: Example physiological time-series with extreme transfer entropy values. Each row was randomly selected from among the 24hr windows which had TE values in the lowest and highest deciles (in both transfer from HR to T and from T to HR).

values and trajectories looked like. I returned to the information trajectories later in the thesis, in the second half of Chapter 3. Firstly, I identified 24hr windows with extreme transfer entropy values between temperature and heart rate, i.e. the highest or lowest deciles. In Figure 2.9, I showed 24hr windows of the raw data corresponding to these. Both transfer entropy values were in the lowest decile in the top row, and in the highest decile in the bottom row. In the middle row, the transfer entropy value in one direction was in the lowest decile and in the highest decile in the other direction. Each row in this figure contained three randomly selected 24hr observations that met these conditions. In some examples, an increase in one physiological variable appeared to coincide with a later increase or decrease in the other variable (suggesting some feedback response), but generally the situation was slightly more complicated than this, particularly since 'expected surprise' is not straightforward to visualise, and minute-to-minute variability is not easy to read from this figure. One thing that was apparent from Figure 2.9 was

that there was much less variability on small timescales in temperature than in heart rate, which in turn corresponded to much lower entropy values (Figure 2.11).

In Figure 2.10, I showed individual information trajectories for four randomly selected patients from the cohort. Three of these patients remained in ICU for the full 15 days, while the other was discharged after 208hrs. I summarised the information trajectories of all 136 patients in the cohort were summarised in Figure 2.11. This figure contains heatmaps, where counts are the number of patients in discretised value ranges (with 50 bins), at each 6hr timestamp. This was useful for visualisation at the cohort-level, but masked individual variability at the patient-level. The general picture from the examples in these figures were that (i) entropy of ABP and HR tended to increase during ICU stay, (ii) entropy of temperature was much lower and more stable within each patient, (iii) mutual information tended to decrease over time, with the largest values between HR and temperature, (iv) transfer entropy from ABP or HR to temperature was lower than in the other direction (and between ABP and HR in either direction), (v) in general, transfer entropy had high within-patient variance but was reasonably constant on the cohort-level. The lower entropy of the temperature time-series was likely to have also contributed to the reduced transfer entropy from ABP to HR, since the 'expected surprise' in each entropy component of Equation 2.7 will be reduced (though exactly how is unclear, since these entropy components do not have consistent sign). One other interesting observation from Figure 2.10 is that the patient who was discharged after 208hrs had significantly higher HR entropy values than almost all other patients (though smaller mutual information), and experienced sudden changes in entropy values for both HR and T in the 24hrs preceding ICU discharge. A possible conclusion here is that this could have represented a short period of clinical deterioration, yet there was no clear evidence in the DDW data to suggest that this occurred (and the fact that the patient was discharged from ICU indicated an improvement in the patient state). Identifying and explaining dramatic shifts in the information trajectories may help to identify rapid changes in clinical prognosis and to help our understanding of the underlying physiological processes, but this is a direction for future research. Initial interpretation from Figures 2.9-2.11 may be useful but does not provide clear and objective evidence, without more detailed analysis. Therefore, in order to investigate these trajectories further, I returned to this problem in the next chapter, using flexible time-series modelling and Bayesian model evidence estimation.

Figure 2.10: Example information trajectories for four patients. These patients were randomly chosen from the Covid-19 cohort that I defined in the main text. Each patient is identified in all subplots by separate colours (pink, blue, green, orange).

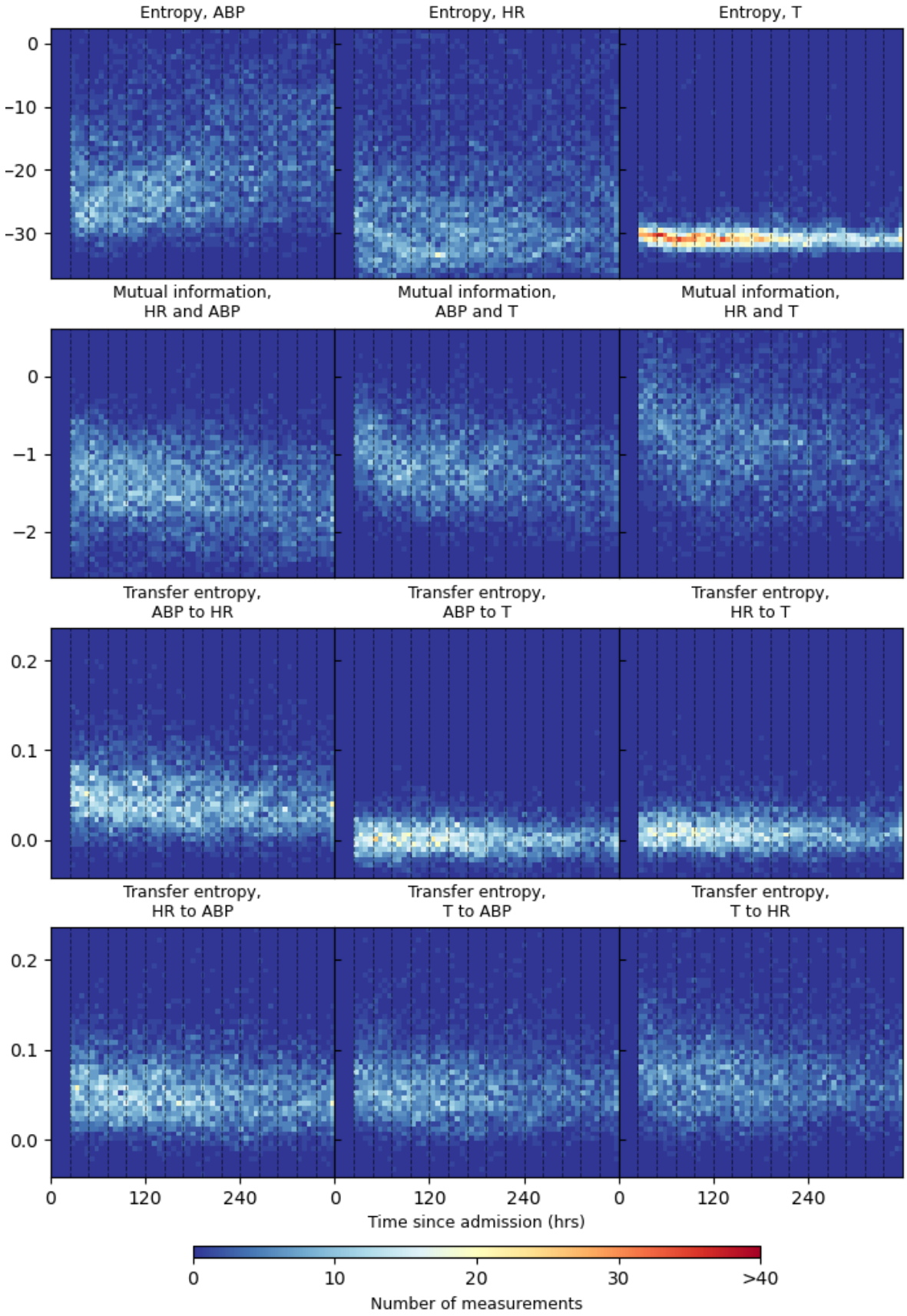

Figure 2.11: Cohort-wide summary of information trajectories. These heatmaps show histogram-binned counts of information-theoretic values (with 50 bins) for all 136 patients within the Covid-19 cohort, at every 6hr timestamp.

CHAPTER 3

Multilevel modelling of time-series and integrated likelihoods

The focus of this chapter was an analysis of frequentist and Bayesian methods for modelling the trajectories of time-series data, in particular data with hierarchical structure. This was as a follow up to the previous chapter, and concluded my work on bivariate causal influence in time-series and brainstem dysfunction in ICU patients with Covid-19. I estimated information flow between temperature and heart rate for a cohort of ICU patients from the Dutch Data Warehouse for Covid-19 (DDW), at regular intervals over the first 15 days in ICU in Chapter 2. The next steps were to compare the information trajectories between this cohort and a control group of mechanically-ventilated ICU patients experiencing similar respiratory distress but without viral infection, i.e. patients in ICU with sepsis and respiratory dysfunction. This full dataset had a multilevel group structure, with groupings at the patient level and hospital level, in which data observations are more closely related to other observations in the same group than to observations in other groups. I used multilevel (mixed effects) linear models to model information trajectories for each cohort separately, and for both cohorts combined. For each information-theoretic measure, I defined a statistical hypothesis that the former (split cohort) model described the data better than the latter (combined cohort) model. If true, this could provide further evidence that the Covid-19 ICU patients had impaired physiological regulation, compared to patients who had a similar severity of respiratory illness.

One of the main Bayesian approaches for quantifying the support for one model over another is to calculate the Bayes factor, which can be estimated using importance sampling or Markov chain Monte Carlo (MCMC) techniques. This is challenging in high-dimensional or hierarchical settings. There are established methods for flexibly modelling data with hierarchical structure, but I was initially frustrated in my attempts to estimate the Bayes factor using MCMC. Instead, I decided to approach the problem from a different angle,

which involved estimating the model evidence for each model, since the Bayes factor is the ratio of the model evidence. As a result, I developed and expanded upon an approach that uses semi-conjugate priors and integrated likelihoods as an intermediate step for this, including a novel extension of the integrated likelihood approach to hierarchical data with multiple-level group structure. The remaining integration in the model evidence can be performed using the previously mentioned sampling techniques, but crucially this part can be performed over a much lower dimension, typically only a few single-valued parameters. Multilevel models have been used widely in applications such as phylogenetics, education and healthcare [110, 111]. My approach here is relevant to any multilevel linear model under weak prior assumptions, and so has wider applicability than the original ICU time-series modelling task. I published my contribution to this integrated likelihood approach as a methodology paper in *PLOS One* [22], separately from the application to Covid-19 ICU data.

The second half of this chapter used this approach to evaluate and conclude the research questions from Chapter 2. In addition to information-theoretic measures of directed causal influence between temperature, heart rate and arterial blood pressure, I also modelled and compared trajectories of the raw data of inflammatory marker variables, C-reactive protein (CRP) level and white blood cell (WBC) count, in order to create a more complete picture of the differences between the two ICU cohorts. In the course of this, I published an open-source implementation of the Sepsis-3 criteria for AmsterdamUMCdb, in the journal *Gigabyte* [112]. I have not yet submitted the work on information trajectories for Covid-19 and sepsis ICU patients for publication but intend to do so.

3.1 Introduction

Multilevel models are a generalisation of linear models to settings in which the model parameters (e.g. regression coefficients) are stratified by groups within the population [113]. For example, individuals in the population may belong to a small number of groups or clusters, and data may be available on both the individual level and the group level. Simple linear models without multilevel structure are generally inferior in situations where the data has hierarchical structure, as it neglects information intrinsically contained within the group structure. In contrast, multilevel models explicitly model at each level of granularity.

The most general multilevel setting in this chapter is a nested three-level hierarchy, with first-level observations (clinical measurements) nested by second-level group (patient pseudo-identifier) nested by third-level group (hospital identifier). The goal of the chapter was to determine whether there were statistically significant differences in a number of variables between two mutually-exclusive patient cohorts (Covid-19 and sepsis). If there was no difference between the cohorts, then they can be considered as both belonging

to the same three-level combined dataset. Otherwise, the cohorts could be viewed as an fourth-level grouping within a merged dataset (i.e. with some shared model coefficients and some depending explicitly on the cohort) or as two completely different datasets (no shared model coefficients or a different model structure entirely).

Given there is a wide variety of modelling structures that can describe multilevel data, an important question is how we might identify an 'optimal' model from multiple competing candidate models. There is not an established methodological answer to this, particularly as the answer may be context-dependent. There are many criteria than can be used to compare the suitability of competing candidate models. In the frequentist setting, this includes Akaike information criterion (AIC) [114], false-discovery rate [115] and likelihood ratio tests [116, 117]. The standard Bayesian approach is to estimate the ratio of the model evidence for each model, where the model evidence is the full likelihood integrated over all parameters in the model with respect to their priors. One advantage of using a Bayesian approach for model comparison is that it implicitly penalises model complexity, as increasing the number of parameters of the model increases the dimension of the parameter space and the integral. However, direct computation of the model evidence is not possible in most cases, except for simple models with fully conjugate priors.

3.1.1 Key contributions

There are two major elements to this chapter, first extending the Bayesian integrated likelihood approach to multilevel models, and then the application of both Bayesian and frequentist approaches to model selection with time-series data from ICU. The key contributions were as follows:

1. I derived integrated likelihoods in multilevel models with semi-conjugate priors, up to a three-level hierarchical data structure. The integrated likelihood calculation is well-established for a simple linear model and for a simple two-level linear model [118], though the latter is generally only presented in simplified matrix algebra form without explicit dependence on variance parameters. To my knowledge, integrated likelihood computations for a general two-level model and a three-level model, which are described in Equations 3.4 and 3.5, have not previously been discussed or published in the literature.
2. The model evidence is usually estimated using sampling methods, such as importance sampling or Monte Carlo integration. I showed that model evidence estimates were more robust and consistent when using the integrated likelihood (sampling over variance parameters) in place of the full likelihood (sampling over all parameters). Using simulated time-series datasets generated from various multilevel models, I

showed that the 'true' model could be identified in each case using the integrated likelihood approach but not using the full likelihood approach.

3. I modelled the trajectories of information-theoretic variables (entropy, mutual information and transfer entropy) for physiological time-series (temperature, heart rate and arterial blood pressure) and of clinical variables (CRP and WBC) for patient cohorts in ICU with Covid-19 or respiratory sepsis. I showed there were significant differences between cohorts in the information trajectories. These findings were agnostic to the basis expansion used to define the function of time and to the statistical paradigm (Bayesian or frequentist). Finally, I contextualised these information-theoretic results in terms of the clinical presentation of patients in each cohort.

3.1.2 Mathematical definition and notation

The basic premise in this chapter involved modelling a variable y as a non-linear smooth function of time t, i.e. $\mathbb{E}[y] = f(t)$, for data $\mathcal{D}$ containing datapoints (y_i, t_i) (ignoring for the moment any hierarchical data structure). This is usually achieved by approximating the unknown non-linear function $f(t)$ as a finite weighted sum of known basis functions $f_j(t)$, i.e. $f(t) \approx \sum_{j=1}^{J} \beta_j f_j(t)$. Given pre-defined basis functions $f_j(t)$, this converts the problem to a regular generalised linear model $\mathbb{E}[y_i] = \beta^T x_i$, where $x_i = (f_1(t_i), \ldots, f_J(t_i))$ are new covariates under the basis expansion of $f(t)$. Under this expansion, the datapoints in $\mathcal{D}$ are (y_i, x_i). Even though this describes a non-linear function of t, it describes a linear model because of linearity in the coefficient β. I assume all models are general linear models, i.e. with normal distribution and identity link. Both frequentist and Bayesian approaches use some form of penalised regression (either explicitly or implicitly) to fit the model. I described the model structure for each setting, e.g. the basis functions, later in Section 3.1.3. In this chapter, I use the notation $\mathbb{1}\{A\}$ to denote the indicator function for some event or condition A.

Data structure. I denote (strictly) hierarchical structure using index notation, with the number of subscript indices equal to the number of levels in this multilevel structure. In a three-level hierarchy, the i^{th} observation in the j^{th} second-level group in the k^{th} third-level group is (y_{ijk}, x_{ijk}), where y_{ijk} is the response variable and x_{ijk} are the covariates. I denote a dataset as $\mathcal{D}$, the most general of which contains (y_{ijk}, x_{ijk}) for $i = 1, \ldots, m_{jk},\ j = 1, \ldots, n_k$ and $k = 1, \ldots, K$. There are m_{jk} and m_k observations in second-level group j and third-level group k respectively (with $m_k = \sum_{j=1}^{n_k} m_{jk}$), and n_k second-level groups in group k. In total, there are K third-level groups, n second-level groups ($n = \sum_{k=1}^{K} n_k$) and m first-level observations ($m = \sum_{k=1}^{K} \sum_{j=1}^{n_k} m_{jk} = \sum_{k=1}^{K} m_k$). The independent variables x_{ijk} are vector-valued, with dimension d. Unless otherwise stated, this includes an intercept

term (i.e. the first element of x_{ijk} has value 1 for all indices). I assume the data is standardised (centred and scaled) prior to any modelling, standardising each dependent and independent variable separately.

Subscripts. At this point, I include a brief note about subscript indices, as there is potential for the subscript notation to become confusing within a variety of hierarchical structures. I have tried to make this notation as consistent as possible. Firstly, when the hierarchical data structure has fewer levels (or a model structure does not require higher-level groups), the corresponding index can be dropped for easier readability. For example, with no hierarchical structure, this dataset $\mathcal{D}$ contains (y_i, x_i) for $i = 1, \ldots, m$. Secondly, in the above, the subscripts ijk are nested. Results in Section 3.2 and their derivations in Appendix B.2 often involve summation (or products) over combinations of these subscripts. Including limits in these sums makes the algebra bulky and difficult to read (particularly in Appendix B.2), so these were left implicit. Summation (or product) over k is from 1 to K, summation over j for a fixed k is from 1 to n_k, and summation over i is from 1 to m_{jk} for fixed k and j. Equivalently, $\sum_k$ means $\sum_{k=1}^{K}$, $\sum_j$ means $\sum_{j=1}^{n_k}$, $\sum_i$ means $\sum_{i=1}^{m_{jk}}$, $\sum_{i,j}$ means $\sum_{j=1}^{n_k}\sum_{i=1}^{m_{jk}}$, etc. Where necessary, I used l or p in place of j in summation and q in place of i in summation, to avoid repeating subscripts.

Models. I denote multilevel linear models as $\mathcal{M}_{ln}$, where l is the number of levels and n an additional index to distinguish between models of the same level. I denote the model parameters as $\theta \in \Theta$, e.g. $\theta = (\beta^T, \sigma^2)^T$. In the Bayesian setting, models are not completely specified unless accompanied by prior distributions for model parameters θ, i.e. $\theta \sim \mathbb{P}_\theta$. I did not explicitly include any priors until Section 3.2, but they are understood to be implicitly accompanied by suitably defined priors when viewed from a Bayesian setting. The simplest model is a linear model with no hierarchical structure:

$$\mathcal{M}_{11}: \; y_i = \beta^T x_i + \epsilon_i, \; \epsilon_i \sim N(0, \sigma^2) \tag{3.1}$$

The idea behind a multilevel model framework is that each level is modelled separately in a cascading sequence, with possible covariates at the different group-levels. For instance, in a two-level hierarchical structure, if there are individual-level covariates x_{ij} and group-level covariates $\tilde{x}_j$, then:

$$\begin{aligned} \mathcal{M}_{21}: \; y_{ij} &= \beta_1^T x_{ij} + u_j + \epsilon_{ij}, \; \epsilon_{ij} \sim N(0, \sigma_y^2) \\ u_j &= \beta_2^T \tilde{x}_j + \eta_j, \; \eta_j \sim N(0, \sigma_\eta^2) \end{aligned}$$

In practice, any group-level covariates are absorbed directly into x_{ij} and do not change value between same-group observations. This model can be then rewritten as:

$$\mathcal{M}_{21}: \; y_{ij} = \beta^T x_{ij} + \eta_j + \epsilon_{ij}, \; \epsilon_{ij} \sim N(0, \sigma_y^2), \; \eta_j \sim N(0, \sigma_\eta^2) \tag{3.2}$$

Instead of directly modelling the unobserved group-level response u_j, the term η_j can be viewed as group-level deviation from the 'population average' (in the terminology of mixed effects models, η_j is a random effect). The model parameters in this instance are $\theta = (\beta^T, \sigma_y^2, \sigma_\eta^2)$. This multilevel linear model can also be rewritten as a single-level linear model with correlated errors [119]:

$$\mathcal{M}_{21}: \; y \sim N(x\beta, V), \; V = \sigma_y^2 I + \sigma_\eta^2 M M^T \tag{3.3}$$

In this case, y is a m-dimensional vector $(y_{11}, \ldots, y_{m_n n})$, x is a $m \times d$ matrix defined similarly, where M is an $m \times n$ matrix indicating group-membership, $M_{ij} = \mathbb{1}\{i \in \{1 + \sum_{k=1}^{j-1} m_k, \ldots, \sum_{k=1}^{j} m_k\}\}$. In some cases, a d'-dimensional subset of covariates x_{ij}, which I denote z_{ij}, have regression coefficients that vary by second-level group:

$$\mathcal{M}_{22}: \; y_{ij} = \beta^T x_{ij} + \eta_j^T z_{ij} + \epsilon_{ij}, \; \epsilon_{ij} \sim N(0, \sigma_y^2), \; \eta_j \sim N(0, \Sigma_\eta(\phi)) \tag{3.4}$$

The covariance matrix Σ_η for d'-dimensional coefficient η_j is symmetric positive-definite and parameterised through ϕ, e.g. $\Sigma_\eta(\phi) = \phi I$ (though the independence assumption in this example is restrictive). Finally, a three-level hierarchical model, with parameters $\theta = (\beta^T, \sigma_y^2, \sigma_\eta^2, \sigma_\zeta^2)$, is:

$$\begin{aligned} \mathcal{M}_{31}: \; & y_{ijk} = \beta^T x_{ijk} + \eta_{jk} + \zeta_k + \epsilon_{ijk}, \\ & \epsilon_{ijk} \sim N(0, \sigma_y^2), \; \eta_{jk} \sim N(0, \sigma_\eta^2), \; \zeta_k \sim N(0, \sigma_\zeta^2) \end{aligned} \tag{3.5}$$

For a linear model $\mathcal{M}_{11}$, I denote the following:

- Likelihood of $\mathcal{D}$ at θ: $p(\mathcal{D}|\mathcal{M}_{11}, \theta)$ (Bayesian) or $\mathcal{L}(\theta|\mathcal{M}_{11}, \mathcal{D})$ (frequentist)
- Prior distribution function for θ: $p(\theta|\mathcal{M}_{11})$
- Posterior distribution function for θ: $p(\theta|\mathcal{M}_{11}, \mathcal{D}) \propto p(\mathcal{D}|\mathcal{M}_{11}, \theta)p(\theta|\mathcal{M}_{11})$
- Integrated likelihood, over β: $p(\mathcal{D}|\mathcal{M}_{11}, \sigma^2) = \int_{\mathbb{R}^d} p(\mathcal{D}|\mathcal{M}_{11}, \beta, \sigma^2)p(\beta|\mathcal{M}_{11}, \sigma^2)d\beta$
- Model evidence (marginal likelihood): $p(\mathcal{D}|\mathcal{M}_{11}) = \int_\Theta p(\mathcal{D}|\mathcal{M}_{11}, \theta)p(\theta|\mathcal{M}_{11})d\theta$

The likelihood is central to both Bayesian (as a function of θ) and frequentist (as a function of $\mathcal{D}$) paradigms. Unless explicitly stated otherwise, I assume independence of priors, so the prior for θ is the product of individual priors e.g. $p(\theta|\mathcal{M}_{11}) = p(\beta|\mathcal{M}_{11})p(\sigma^2|\mathcal{M}_{11})$. For higher-level models, the likelihood and prior are the same as above, but the model

evidence and integrated likelihood depend on the multilevel structure of the model, since group-level deviation terms (η and ζ) are additional nuisance parameters that must be also integrated out. For example, for $\mathcal{M}_{21}$:

- Integrated likelihood, over β and η:
  $p(\mathcal{D}|\mathcal{M}_{21}, \sigma_y^2, \sigma_\eta^2) = \int_{\mathbb{R}^{d+n}} p(\mathcal{D}|\mathcal{M}_{21}, \beta, \eta, \sigma_y^2, \sigma_\eta^2) p(\eta|\mathcal{M}_{21}, \sigma_\eta^2) p(\beta|\mathcal{M}_{21}) d\beta d\eta$
- Model evidence (marginal likelihood):
  $p(\mathcal{D}|\mathcal{M}_{21}) = \int_{\Theta \times \mathbb{R}^n} p(\mathcal{D}|\mathcal{M}_{21}, \theta, \eta) p(\eta|\mathcal{M}_{21}, \theta) p(\theta|\mathcal{M}_{21}) d\theta d\eta$

Similarly, for $\mathcal{M}_{31}$:

- Integrated likelihood, over β and η: $p(\mathcal{D}|\mathcal{M}_{31}, \sigma_y^2, \sigma_\eta^2, \sigma_\zeta^2) =$
  $\int_{\mathbb{R}^{d+n+K}} p(\mathcal{D}|\mathcal{M}_{31}, \beta, \eta, \zeta, \sigma_y^2, \sigma_\eta^2) p(\zeta|\mathcal{M}_{31}, \sigma_\zeta^2) p(\eta|\mathcal{M}_{31}, \sigma_\eta^2) p(\beta|\mathcal{M}_{31}) d\beta d\eta d\zeta$
- Model evidence (marginal likelihood):
  $p(\mathcal{D}|\mathcal{M}_{31}) = \int_{\Theta \times \mathbb{R}^{n+K}} p(\mathcal{D}|\mathcal{M}_{31}, \theta, \eta, \zeta) p(\zeta|\mathcal{M}_{31}, \theta) p(\eta|\mathcal{M}_{31}, \theta) p(\theta|\mathcal{M}_{31}) d\theta d\eta d\zeta$

3.1.3 Overview and related work

Frequentist model comparison. One of most common model selection tools is the Akaike information criterion [114], a measure of the goodness of fit. This is defined as the following, where k is the number of unconstrained parameters:

$$\text{AIC} = 2k - 2\text{max}_{\theta \in \Theta} \log \mathcal{L}(\theta|\mathcal{M}, \mathcal{D})$$

Candidate models can be ranked by their AIC to evaluate the relative suitability of each. AIC is rooted in information theory, as an asymptotic estimate of the relative amount of information lost when describing the true data generation process by the statistical model, i.e. it is an asymptotic Kullback-Liebler divergence. A more complicated model will generally improve the goodness of fit but risks overfitting to the data. AIC penalises model complexity to mitigate against this.

Models can be compared more explicitly in a frequentist setting using statistical hypothesis tests, e.g. a likelihood-ratio test for imposing some constraints on the full parameter space Θ. However, the exact distribution of a likelihood ratio under competing hypotheses is often difficult to compute. One example relevant to multilevel modelling involves generalised additive mixed models.

Generalised additive mixed models. The frequentist multivariate extension to generalised linear models for the smooth non-linear function $f(\cdot)$ are generalised additive models (GAMs) [120], where $f(\cdot)$ may be a function of multiple variables (rather than a function of t). In a GAM, the multivariate $f(\cdot)$ is represented by a superposition of smooth

univariate functions, which are each approximated in turn by a finite sum of basis functions. The rationale for this expansion is based a modified version of the Kolmogorov-Arnold representation [121]. Generalised additive mixed models (GAMMs) further extend the GAM model with the inclusion of higher-level group effects (random effects). In GAMs and GAMMs, smoothing basis functions are used alongside penalised least-squares regression to prevent too much 'wiggliness' (i.e. overfitting) in the function $f(\cdot)$. These smoothing bases are usually spline functions, e.g. weighted radial basis functions or piece-wise polynomial with smoothness enforced at fixed changepoint knots, $\tau_1, \ldots, \tau_{d+1}$.

Later in Section 3.3, I used a cyclic cubic spline basis for the generalised linear mixed model (a univariate GAMM), i.e. $f(t)$ is a cubic polynomial on every interval $[\tau_n, \tau_{n+1}]$ for $n = 1, \ldots, d$. It is more convenient to describe the maximum penalised likelihood using matrix algebra rather than subscript notation. For the general second-level model $\mathcal{M}_{22}$ (Equation 3.3), penalised regression minimises the following:

$$\mathcal{L}(\theta|\mathcal{M}_{22}, \mathcal{D}) + \lambda \int_{\tau_1}^{\tau_d} f''(t)^2 dt = -\frac{1}{2}|V| - \frac{1}{2}(y - x\beta)V^{-1}(y - x\beta) + \lambda\beta^T S\beta$$

The basis functions $f_j(t)$ and the matrix S in the above are described fully in Appendix B.1, while the correlated-error covariance V (Equation 3.3) depends on the variance parameters σ_y^2 and σ_η^2. The maximum penalised likelihood estimate (MPLE) for β is:

$$\hat{\beta} = (x^T\hat{V}^{-1}x + \hat{\lambda}S)^{-1}x^T\hat{V}^{-1}y = \hat{P}y$$

Then, for an m-dimensional function $c(t)$ with i^{th} component equal to $\sum_{j=1}^{d} f_j(t)\hat{P}_{ji}$, the MPLE estimate for $f(t)$ is $\hat{f}(t) = \sum_{j=1}^{d} f_j(t)(\hat{P}y)_j = \hat{c}(t)^T y$. Imposing a penalty term (like the term $\beta^T S\beta$) is in some sense equivalent to assigning prior beliefs about the model characteristics. In this case, if λ, σ_y^2 and σ_η^2 were known, then if β has prior $p(\beta|\mathcal{M}_{22}) \sim N(0, S^{-1}\sigma_y^2/\lambda)$, it has posterior $p(\beta|\mathcal{M}_{22}, \mathcal{D}) \sim N(\hat{\beta}, (x^T V^{-1}x + \lambda S)^{-1}\sigma_y^2)$. This posterior can be used to generate credible regions for inference using the GAMM model.

The goal was to compare non-linear functions $f(t)$ and $g(t)$ describing time-series trajectories for two cohorts $\mathcal{D}_1 = (y^{(1)}, t^{(1)})$ and $\mathcal{D}_2 = (y^{(2)}, t^{(2)})$. Assuming each cohort has different model parameters β_l and V_l for $l = 1, 2$, then the joint correlated-errors single-level model is $y^{(l)} \sim N(x^{(l)}\beta_l, V_l)$. Following [122, 123], a frequentist hypothesis test for model comparison between cohorts and a measure of the difference between $f(t)$ and $g(t)$ are:

$$H_0 \colon f(t) = g(t),\ H_1 \colon f(t) \neq g(t) \qquad \Delta[f(\cdot), g(\cdot)] = \int_{\tau_1}^{\tau_{d+1}} (f(t) - g(t))^2 dt \qquad (3.6)$$

The test statistic for this hypothesis test is:

$$\Delta_{\text{obs}} = \Delta[\hat{f}, \hat{g}] = \int_{\tau_1}^{\tau_{d+1}} (y^{(0)})^T \hat{c}_0(t)\hat{c}_0(t)^T y^{(0)} dt = (y^{(0)})^T \hat{C} y^{(0)}$$

$$\hat{C} = \int_{\tau_1}^{\tau_{d+1}} \hat{c}_0(t)\hat{c}_0(t)^T dt,\ y^{(0)} = \begin{pmatrix} y^{(1)} \\ y^{(2)} \end{pmatrix},\ \hat{c}_0(t) = \begin{pmatrix} \hat{c}_1(t) \\ -\hat{c}_2(t) \end{pmatrix},\ \hat{V}_0 = \begin{pmatrix} \hat{V}_1 & 0 \\ 0 & \hat{V}_2 \end{pmatrix}$$

The null hypothesis is rejected if $\Delta[\hat{f}, \hat{g}] > 0$. This has an approximate p-value for the observed test statistic Δ_{obs} is $p(\chi_1^2 > \Delta_{\text{obs}}/\text{tr}(\hat{C}\hat{V}_0)^2)$ [122], where tr is the matrix trace.

This approach can be further generalised to include additional covariates unrelated to the function $f(t)$, e.g. a covariate z with coefficient α. Then, for $\mathcal{M}_{23} : y \sim N(x\beta + z\alpha, V)$,

$$\mathcal{L}(\theta|\mathcal{M}_{23}, \mathcal{D}) + \lambda \int_{\tau_1}^{\tau_d} f''(t)^2 dt = -\frac{1}{2}|V| - \frac{1}{2}(y - x\beta - z\alpha)V^{-1}(y - x\beta - z\alpha) + \lambda\beta^T S\beta$$

$$\hat{\beta} = (x^T\hat{W}x + \hat{\lambda}S)^{-1}x^T\hat{W}^{-1}y = Py,\ W = V^{-1} - V^{-1}z(z^TV^{-1}z)^{-1}z^TV^{-1}$$

Given this MPLE, the remaining definitions and hypothesis test remain the same as the above.

Modular time-series models. A more Bayesian approach to this problem allows more flexibility in the model structure, while model coefficients are already implicitly penalised through their prior distributions. In particular, the function $f(t)$ does not need to be smooth or continuous in this framework. In [124], the authors use a modular model with growth and seasonality components as an alternative forecasting model to an autoregressive integrated moving average (ARIMA) model. I used a similar basis expansion in the Bayesian sections of this chapter. This included a piece-wise linear term and a truncated Fourier series term:

$$f(t) = \lambda + \sum_{n=1}^{d_1} \delta_n (t - s_n)\, \mathbb{1}\{t > s_n\} + \sum_{n=1}^{d_2} \left(a_n \cos\frac{2\pi nt}{P} + b_n \sin\frac{2\pi nt}{P} \right) \tag{3.7}$$

The piece-wise linear component has fixed changepoints s_n for $n = 1, \ldots, d_1$, at which the function is continuous but not smooth, and the Fourier series has fixed periodicity P. With $s_1 \leq \min_i t_i$, the baseline gradient is δ_1, while δ_n are gradient adjustments for $n = 2, \ldots, d_1$. The generalised linear model for this is $\mathbb{E}[y_{ijk}] = \beta^T x_{ijk}$, with:

$$\beta^T = (\lambda, \delta^T, a^T, b^T),\ x_{ijk} = \left(1, (t_{ijk} - s_1)\, \mathbb{1}\{t_{ijk} - s_1\}, \ldots, \cos\frac{2\pi t_{ijk}}{P}, \ldots, \sin\frac{2\pi t_{ijk}}{P}, \ldots\right)$$

The model coefficient β includes the intercept adjustment λ, gradient terms δ_n and the Fourier coefficients a_n and b_n. Consequently, the dimension of x_{ijk} and β is $d = 1 + d_1 + 2d_2$.

Autoregressive models and other alternatives. In the above paragraphs, I described non-linear regression two approaches. These model the data as a direct function of time t, so each group shares the same response variable trajectory with i.i.d. errors, up to a group-level effects. There are alternative approaches to this. The most common are autoregressive models, which model the response variable at time t as a function of the response variable at previous times e.g. at time $t-1$. Hidden Markov models and simple recurrent neural networks leverage time-series data similarly. As with the non-linear regression model, these types of models can be augmented with multilevel effects, e.g. [125–127]. It was not feasible to investigate all of these models within this thesis, so I focused on the models previously introduced. I discuss this choice again in Chapter 6.

Bayes factor. The standard Bayesian approach to model selection is to calculate the Bayes factor, defined as the ratio of the model evidence for each model. For two competing models, $\mathcal{M}_a$ and $\mathcal{M}_b$, the Bayes factor is $\mathrm{BF}_{ab} = p(\mathcal{D}|\mathcal{M}_a)/p(\mathcal{D}|\mathcal{M}_b)$, with $\mathrm{BF}_{ba} = 1/\mathrm{BF}_{ab}$. The value of the Bayes factor indicates the strength of evidence for one model over the other, and interpretation is provided via tables proposed by Jeffreys [128] or Kass [129]. A Bayes factor $\mathrm{BF}_{ab} > 1$ suggests that the data supports model $\mathcal{M}_a$ more strongly than model $\mathcal{M}_b$. Using the model evidence guards against model overfitting, since it involves integration with respect to priors over the entire parameter space Θ, and each additional parameter increases the dimension of this parameter space. Competing models may have different subsets of independent variables or different prior beliefs associated with model parameters, but the data $\mathcal{D}$ must remain fixed and include the same m individual observations.

Markov chain Monte Carlo methods. The model evidence for multilevel linear models are analytically intractable, except for single-level models with a fully conjugate normal-inverse-gamma prior (e.g. [118]). There are several approaches for estimating the Bayes factor, including approximating the integral as a sum (e.g. importance sampling [130] or sequential Monte Carlo [131]), numerical optimisation methods [132, 133], and jointly estimating posterior probabilities of candidate models using Monte Carlo methods.

In the latter, each model is identified by a new subscript index. Hierarchical MCMC sampling alternates between two sampling steps, first across the model indices and then for model parameters of the current chosen model. This sampling scheme can accommodate multiple models, $\mathcal{M}_j$, $j = 1, \ldots, J$, and requires both the prior probability $p(\mathcal{M}_j)$ for choosing individual models and the prior distributions $p(\theta_j|\mathcal{M}_j)$ for parameters of each model. The relative acceptance frequencies in MCMC chain for the model index provides an approximation to the posterior probability $p(\mathcal{M}_j|\mathcal{D})$ for each model. This bypasses the need to calculate the model evidence, since the Bayes factor between any pair of models is estimated as $\mathrm{BF}_{ab} = p(\mathcal{M}_a|\mathcal{D})/p(\mathcal{M}_b|\mathcal{D}) \times p(\mathcal{M}_b)/p(\mathcal{M}_a)$. However, the challenge in this

approach is to ensure sufficient mixing in the MCMC chain for the model index, since if MCMC spends too long repeatedly choosing and exploring only one of the models, the posterior probability is biased by the resulting extreme autocorrelation. There are several variants of this ABC framework, including reversible-jump MCMC [134] and product-space MCMC [135]. The former is difficult to implement because it has a non-constant parameter space with variable dimension.

Product-space MCMC defines a distribution over the product of all model indices and their parameters, i.e. at each step it samples $(\mathcal{M}_j, \theta_{1:J}) \in \{\mathcal{M}_{11}, \ldots, \mathcal{M}_J\} \times \prod_j \Theta_j$, where $\mathcal{M}_j$ is any generic model rather than a linear model with j levels and $\theta_{1:J} = (\theta_1^T, \ldots, \theta_J^T)$. In the two-step sampling process, the algorithm alternates between two steps, first selecting one candidate model and then sampling the parameters of that particular model in the usual manner (e.g. using Metropolis-Hastings or Hamiltonian Monte Carlo). Therefore, the n^{th} iteration of the MCMC involves sampling steps $p(\mathcal{M}_j^{(n+1)}|\mathcal{D}, \theta^{(n)})$ and $p(\theta^{(n+1)}|\mathcal{M}_j^{(n+1)}, \mathcal{D})$. The likelihood of selecting model $\mathcal{M}_j$ depends only on its own parameters θ_j, i.e. $p(\mathcal{D}|\mathcal{M}_j, \theta_{1:J}) = p(\mathcal{D}|\mathcal{M}_j, \theta_j)$. The remaining parameters $\theta_{-j} = (\theta_1^T, \ldots, \theta_{j-1}^T, \theta_{j+1}^T, \ldots \theta_J^T)$ need to specified as well, but can be chosen arbitrarily without affecting the marginals of $\mathcal{M}_j$ and θ_j. A standard assumption is that the parameters θ_{-j} are independent of each other and of θ_j, i.e. $p(\theta_{-j}|\mathcal{M}_j, \theta_j) = \prod_{k \neq j} p(\theta_k|\mathcal{M}_j)$. The full posterior for $(\mathcal{M}_j, \theta_{1:J})$ is $p(\mathcal{M}_j, \theta_{1:J}|\mathcal{D}) \propto p(\mathcal{D}|\mathcal{M}_j, \theta_j)p(\theta_j|\mathcal{M}_j)p(\mathcal{M}_j) \prod_{k \neq j} p(\theta_k|\mathcal{M}_j)$. The product-space MCMC approach is very inefficient if each model does not have a reasonable chance of selection in the first of the two sampling steps. This is likely to be the case if the distribution functions $p(\theta_k|\mathcal{M}_j)$ are chosen poorly and are implausible under $\mathcal{M}_k$. This motivates a construct called pseudo-priors, which set the distribution of unused parameters θ_k to an approximate of their posterior under $\mathcal{M}_k$, i.e. $p(\theta_k|\mathcal{M}_j) \approx p(\theta_k|\mathcal{D}, \mathcal{M}_k)$. In theory, pseudo-priors reduce the first sampling step for $\mathcal{M}_j$ to an approximation of its true model posterior probability, i.e. $p(\mathcal{M}_j|\mathcal{D}, \theta) \approx p(\mathcal{M}_j|\mathcal{D}) = \int_{\Theta_j} p(\mathcal{M}_j, \theta_j|\mathcal{D}) d\theta_j$. Typically the prior over models will be a discrete uniform distribution with $p(\mathcal{M}_j) = 1/J$ and the Bayes factor reduces to the proportion of MCMC iterations in which each model is selected.

Instead of estimating the Bayes factor using a hierarchy of models, a fully Bayesian hierarchical scheme could assign shared priors to any cohort-specific parameters. For example, suppose the model parameters for two cohorts are β_1 and β_2, with $\beta_l \sim N(\mu, \Sigma),\ l = 1, 2$. This is routinely the case throughout this chapter, where the prior mean is typically fixed as $\mu = 0$. Instead, by placing a further prior on μ, we can test the assumption that the priors for β_1 and β_2 are correlated, i.e. whether the credible interval for μ under its posterior includes zero or not. This has several advantages. Similar to the product-space MCMC, it is a joint model for cohort-specific parameters, but it avoids issues that arise from mixing discrete and continuous sampling. Furthermore, the posteriors of these

cohort-specific parameters could be used to quantify whether there exists shared structure between cohorts, instead of using the model evidence for this purpose. This may be preferable in high-dimensional settings, when likelihoods are often sub-optimal. This offers an attractive alternative to the approaches in this chapter. However, in the interest of time, this was not pursued further here.

Importance sampling and sequential Monte Carlo (SMC). An alternative approach is a direct Monte Carlo estimate of the model evidence using importance sampling. In a naïve Monte Carlo integration (later used in Section 5.2.2), an integral of a function $g(\phi)$, with respect to distribution $\mathbb{P}$ with pdf $p(\phi)$, is estimated as the sample mean of a large number of samples drawn from that distribution:

$$\mathbb{E}[g(\phi)] = \int_{\Phi} g(\phi)p(\phi)d\phi \approx \frac{1}{N}\sum_{i=1}^{N} g(\phi_i),\ \phi_i \sim \mathbb{P} \tag{3.8}$$

The difficulty in this scheme is that it is often challenging to sample from $\mathbb{P}$. It is usually easier to evaluate an unnormalised $\tilde{p}(\phi) = c_p p(\phi)$ than to evaluate $p(\phi)$, because the normalising constant c_p is difficult to calculate. This means that this result holds for unnormalised kernels as well as fully-specified distributions, e.g. if the distribution in question is the unnormalised posterior $p(\mathcal{D}|\mathcal{M},\theta)p(\theta)$.

Importance sampling circumnavigates sampling issues by specifying a proposal distribution $\mathbb{P}_q$ that has pdf $q(\phi)$, is an approximation to $\mathbb{P}$, and is easy to sample from and easy to evaluate. Then:

$$\mathbb{E}[g(\phi)] = \int_{\Theta} g(\phi)\frac{p(\phi)}{q(\phi)}q(\phi)d\phi \approx \frac{1}{N}\sum_{i=1}^{N}\frac{p(\phi_i)}{q(\phi_i)}g(\phi_i) = \sum_{i=1}^{N} w_i g(\phi_i),\ \phi_i \sim \mathbb{P}_q \tag{3.9}$$

$$w_i = \frac{\tilde{p}(\phi_i)/\tilde{q}(\phi_i)}{\sum_{j=1}^{N}\tilde{p}(\phi_j)/\tilde{q}(\phi_j)},\ \sum_{i=1}^{N} w_i = 1$$

Importance sampling is asymptotically unbiased but may have large variance if an insufficient number of samples are taken, particularly if the tails of proposal and target distributions are not well aligned. Sequential Monte Carlo (SMC) extends and improves this by alternating between Metropolis-Hastings sampling and importance sampling. SMC is also referred to as a particle filter, because of theoretical foundations in mean-field particle methods from fluid dynamics, where the pair $\{\theta_i, w_i\}$ describes the coordinates and weight of a particle. Particle filters are widely used as hidden Markov models in modelling the state space evolution of dynamical systems. In the context of marginal likelihood estimation, the sequential element of SMC is artificially introduced by an annealing process from the prior for θ to its posterior. A temperature parameter γ controls

the annealing, with $p_n(\theta|\mathcal{M},\mathcal{D}) \propto p(\mathcal{D}|\mathcal{M},\theta)^{\gamma_n} p(\theta|\mathcal{M})$ for $0 = \gamma_0 < \ldots < \gamma_n < \gamma_{n+1}$, until $\gamma \geq 1$. The n^{th} stage has N particle-weight pairs $\{\theta_i^{(n)}, w_i^{(n)}\}$ for $i = 1, \ldots, N$. In the first stage, particles $\theta_i^{(0)}$ are drawn from the known prior distribution, since $p_0(\theta|\mathcal{M},\mathcal{D}) \propto p(\theta|\mathcal{M})$. The corresponding (unnormalised) weights are used to estimate the likelihood, i.e. $w_i^{(0)} = \tilde{w}_i^{(0)} / \sum_j \tilde{w}_j^{(0)}$ with $\tilde{w}_i^{(0)} = p(\mathcal{D}|\mathcal{M},\theta^{(0)})$. At the n^{th} stage, with temperature γ increased, the particles are 'evolved' by running independent Metropolis-Hastings chains initiated at $\theta_i^{(n-1)}$. This step uses the target distribution $p_n(\theta|\mathcal{M},\mathcal{D})$ and a tractable proposal distribution that depends on previous stages, e.g. a multivariate normal distribution with mean and covariance matching the posterior from the previous iteration. The importance weights are then updated by multiplying with the likelihood of the particle at the current stage and are then re-normalised. When the annealing process is complete, the distribution of θ from the final stage is the posterior distribution. Finally, the model evidence is estimated as the mean of the unnormalised weights: $p(\mathcal{D}|\mathcal{M}) \approx 1/N \sum_{i=1}^{N} \tilde{w}_i$.

Thermodynamic integration is a similar approach that can be used for Bayes factor estimation. It also sequentially tempers the likelihood between 0 and 1. An estimate of the log model evidence comes from path sampling via Metropolis-Hastings, alongside numerical integration over different temperatures [136, 137]. As with sequential Monte Carlo, this approach could be adapted to use the integrated likelihoods derived in this chapter, but this has not been performed in this thesis.

3.2 Model evidence and integrated likelihoods

The idea in this chapter was to to compare time-series trajectory models for two cohort datasets, $\mathcal{D}_1$ and $\mathcal{D}_2$, which contained datapoints $(y_{ijk}^{(1)}, t_{ijk}^{(1)})$ and $(y_{ijk}^{(2)}, t_{ijk}^{(2)})$ respectively, where y_{ijk} were standardised by their mean and standard deviation and t_{ijk} were normalised to the interval $[0, 1]$. The number of individuals and groups in each cohort does not need to the same, so the subscripts may have different ranges for $\mathcal{D}_1$ and $\mathcal{D}_2$. I first focused on the frequentist approach, where a framework to do this had already been established (Equation 3.6). I also wanted to validate and strengthen my findings using a more flexible Bayesian framework (Equation 3.7). As such, I needed to define two competing models and estimate the Bayes factor. To be able to perform the Bayesian model comparison, both models have to be defined for the same data (with t_{ijk} now replaced by covariates x_{ijk}, as defined by Equation 3.7). To enable this, I defined the full combined dataset $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2$ as the datapoints (y_{ijk}, x_{ijk}) from both cohort datasets. As this section (Section 3.2 describes only Bayesian methodology, I explicitly stated priors for each model, with an unspecified joint distribution $\mathbb{P}_\theta$ defining the most general prior for θ.

In the baseline model, denoted $\mathcal{M}_b$, both cohorts were assumed to be from the same underlying data generation process, i.e. there was no significant difference between the

cohorts, so all model parameters are shared across cohorts. For a three-level hierarchical structure, $\mathcal{M}_b$ is of the form:

$$\begin{aligned}\mathcal{M}_{31}: \; & y_{ijk} = \beta^T x_{ijk} + \eta_{jk} + \zeta_k + \epsilon_{ijk} \\ & \epsilon_{ijk} \sim N(0, \sigma_y^2), \; \eta_{jk} \sim N(0, \sigma_\eta^2), \; \zeta_k \sim N(0, \sigma_\zeta^2) \\ & \theta = (\beta^T, \sigma_y^2, \sigma_\eta^2, \sigma_\zeta^2)^T \sim \mathbb{P}_\theta\end{aligned} \tag{3.10}$$

In the alternate model, denoted $\mathcal{M}_a$, the cohorts were fundamentally distinct from each other and came from different underlying data generation processes, i.e. there was a significant difference between the cohorts. In this case, there were twice as many model parameters, since each set of parameters depends explicitly on the cohort label. Therefore, $\mathcal{M}_a$ is of the form:

$$\begin{aligned}\mathcal{M}_{32}: \; & y_{ijk}^{(l)} = \beta_l^T x_{ijk}^{(l)} + \eta_{jk}^{(l)} + \zeta_k^{(l)} + \epsilon_{ijk}^{(l)}, \quad l = 1, 2 \\ & \epsilon_{ijk}^{(l)} \sim N(0, \sigma_{y,l}^2), \; \eta_{jk}^{(l)} \sim N(0, \sigma_{\eta,l}^2), \; \zeta_k^{(l)} \sim N(0, \sigma_{\zeta,l}^2) \\ & \theta_l = (\beta_l^T, \sigma_{y,l}^2, \sigma_{\eta,l}^2, \sigma_{\zeta,l}^2)^T \sim \mathbb{P}_{\theta_l}, \; \theta = (\theta_1^T, \theta_2^T)^T\end{aligned} \tag{3.11}$$

As a reminder, the strength of evidence for one model over the other is given by the value of $\text{BF}_{ab} = p(\mathcal{D}|\mathcal{M}_a)/p(\mathcal{D}|\mathcal{M}_b)$. Assuming $\mathcal{D}_1$ and $\mathcal{D}_2$ are completely distinct (i.e. no shared or overlapping group structure), then each cohort dataset can be modelled separately using alternate models of the form $\mathcal{M}_{31}$. Denoting these $\mathcal{M}_a^{(1)}$ and $\mathcal{M}_a^{(2)}$, the model evidence for $\mathcal{M}_a$ is then $p(\mathcal{D}|\mathcal{M}_a) = p(\mathcal{D}_1|\mathcal{M}_a^{(1)})p(\mathcal{D}_2|\mathcal{M}_a^{(2)})$.

Difficulties with MCMC sampling. My first attempts to estimate the Bayes factor focused around product-space MCMC. This proved challenging, because alternating between discrete Gibbs sampling (on the set of models) and continuous Hamiltonian Monte Carlo sampling (over the model parameters) can be very fragile, with strongly autocorrelated chains and extremely low acceptance rates for new proposed states. It proved difficult to get any meaningful results using this method, so I realised that I needed a more carefully considered sampling scheme.

One of the biggest challenges with estimating the model evidence using SMC sampling in high-dimensional parameter spaces. While there are asymptotic results for the validity of SMC, in practice the estimates suffer when there are a large number of regression coefficients β or a multilevel group structure. Multilevel models are particularly hard to sample, because sampling of group-level deviance terms (η and ζ) is dependent on sampled variance parameters (σ_η^2 and σ_ζ^2), and the non-constant variance means that it is difficult to achieve a stationary distribution for these deviance terms. One solution to these issues is to impose specific distributional forms on the priors. In particular, Gaussian

priors for β are semi-conjugate in a multilevel linear model with normally-distributed errors and identity link. This means that all non-variance parameters, including the (potentially high-dimensional) regression coefficients and group-level deviance terms, are treated as 'nuisance' parameters and can be analytically integrated out of the full model likelihood. This reduces the full likelihood on all model parameters to a partially-integrated likelihood on only variance parameters. Instead of sampling over all model parameters, the model evidence can be estimated using SMC sampling over the remaining parameters, so this converts the SMC requirements from a high-dimensional sampling scheme to a low-dimensional sampling scheme (though with a more complicated likelihood function in place of a composite of simple likelihood functions). In theory, both model evidence estimates are identical, but I showed later in Section 3.2.3 that sampling over the lower-dimensional parameter space yields improved results, reduces the bias and variance in estimates, and typically improves computational efficiency.

In the rest of this section, I provided results of integrated likelihoods for models $\mathcal{M}_{11}$ (Equation 3.1), $\mathcal{M}_{21}$ (Equation 3.2), $\mathcal{M}_{22}$ (Equation 3.4) and $\mathcal{M}_{31}$ (Equation 3.5). The latter two of these are new results, so I have included full derivations for these two in Appendix B.2. In practice, it is more convenient to work with log (integrated) likelihoods for computational reasons.

3.2.1 Linear models and fully conjugate priors

Linear model. First, I considered a linear model $\mathcal{M}_{11}$ (Equation 3.1) with independent priors for β and σ^2, where the prior for β was Gaussian with mean μ and covariance Σ, i.e. $p(\theta|\mathcal{M}_{11}) = p(\beta|\mathcal{M}_{11})p(\sigma^2|\mathcal{M}_{11})$, $p(\beta|\mathcal{M}_{11}) \sim N(\mu, \Sigma)$. At this stage, it was not necessary to specify a prior for σ^2, though later I used an inverse-gamma prior. As a reminder, $\mathcal{M}_{11}$ has no hierarchical structure, so group-level indices can be dropped and the datapoints are (y_i, x_i) for $i = 1, \ldots, m$. The integrated likelihood for this is:

$$p(\mathcal{D}|\mathcal{M}_{11}, \sigma^2) = \int_{\mathbb{R}^d} \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\beta-\mu)^T\Sigma^{-1}(\beta-\mu)\right) \times \prod_i \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left(-\frac{1}{2\sigma^2}(y_i - \beta^T x_i)^2\right) d\beta$$

Rearranging the integrand gives posterior mean and covariance (conditional on σ^2):

$$\hat{\Sigma}^{-1} = \Sigma^{-1} + \frac{1}{\sigma^2}\sum_i x_i x_i^T, \quad \hat{\mu} = \hat{\Sigma}\left(\Sigma^{-1}\mu + \frac{1}{\sigma^2}\sum_i x_i y_i\right)$$

Integrating out β and taking the logarithm, the log integrated likelihood is:

$$\log p(\mathcal{D}|\mathcal{M}_{11},\sigma^2) = -\frac{1}{2}\left(\log|\hat{\Sigma}^{-1}| + \log|\Sigma| + n\log(2\pi\sigma^2) + \mu^T\Sigma^{-1}\mu + \frac{1}{\sigma^2}\sum_i y_i^2 - \hat{\mu}^T\hat{\Sigma}^{-1}\hat{\mu}\right) \tag{3.12}$$

Fully-conjugate normal-inverse-gamma prior. For a linear model with fully conjugate normal-inverse-gamma (NIG) prior, the model evidence can be derived in full. I denoted this $\mathcal{M}_{12}$ to distinguish it from the above linear model. In the NIG prior, $\beta, \sigma^2 \sim NIG(a, b, \mu, \Sigma)$, the prior covariance for β is a function of the unknown variance parameter σ^2, i.e. $\sigma^2 \sim IG(a, b)$, $\beta|\sigma^2 \sim N(\mu, \sigma^2\Sigma)$. For this model, the joint posterior distribution is also NIG, i.e. $\beta, \sigma^2|\mathcal{D} \sim NIG(\hat{a}, \hat{b}, \hat{\mu}, \hat{\Sigma})$, with $\hat{a} = n/2 + a$, and:

$$\hat{\Sigma}^{-1} = \Sigma^{-1} + \sum_i x_i x_i^T, \quad \hat{\mu} = \hat{\Sigma}\left(\Sigma^{-1}\mu + \sum_i x_i y_i\right)$$
$$\hat{b} = b + \frac{1}{2}\left(\sum_i y_i^2 - (\sum_i x_i y_i)^T(\Sigma^{-1} + \sum_i x_i x_i^T)^{-1}(\sum_i x_i y_i)\right)$$

The log integrated likelihood and log model evidence are then:

$$\log p(\mathcal{D}|\mathcal{M}_{12},\sigma^2) = -\frac{1}{2}\left(\log|\hat{\Sigma}^{-1}| + \log|\Sigma| + n\log(2\pi\sigma^2) + \frac{1}{\sigma^2}\mu^T\Sigma^{-1}\mu + \frac{1}{\sigma^2}\sum_i y_i^2 - \frac{1}{\sigma^2}\hat{\mu}^T\hat{\Sigma}^{-1}\hat{\mu}\right)$$

$$\log p(\mathcal{D}|\mathcal{M}_{12}) = -\frac{1}{2}\left(\log|\hat{\Sigma}^{-1}| + \log|\Sigma| + n\log(2\pi) - 2a\log b + 2\hat{a}\log(\hat{b}) - 2\log\Gamma(\hat{a}) + 2\log\Gamma(a)\right) \tag{3.13}$$

The fully conjugate NIG prior is convenient in that it allows an analytic expression for the model evidence, but it also imposes a restrictive prior relationship between β and σ^2, and is generally not a useful or realistic assumption in practice [118, 138]. Furthermore, the conjugacy of the NIG prior does not extend to multilevel models.

3.2.2 Multilevel models

Two-level model. Next, I considered the basic multilevel model $\mathcal{M}_{21}$ (Equation 3.2), with a Gaussian prior for β, i.e. $p(\beta|\mathcal{M}_{21}) \sim N(\mu, \Sigma)$, independent of the priors for both variance parameters. For data $\mathcal{D}$ with datapoints (y_{ij}, x_{ij}), $i = 1, \ldots, m_j$ and $j = 1, \ldots, n$,

the integrated likelihood is:

$$p(\mathcal{D}|\mathcal{M}_{21}, \sigma_y^2, \sigma_\eta^2) = \int_{\mathbb{R}^d} \int_{\mathbb{R}^n} \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\beta-\mu)^T\Sigma^{-1}(\beta-\mu)\right) \times \prod_j \frac{1}{(2\pi\sigma_\eta^2)^{1/2}} \exp\left(-\frac{\eta_j^2}{2\sigma_\eta^2}\right) \times \prod_{i,j} \frac{1}{(2\pi\sigma_y^2)^{1/2}} \exp\left(-\frac{(y_{ij}-\beta^T x_{ij}-\eta_j)^2}{2\sigma_y^2}\right) d\eta d\beta$$

Following similar algebra as for the single-level linear model, the log integrated likelihood becomes:

$$\log p(\mathcal{D}|\mathcal{M}_{21}, \sigma_y^2, \sigma_\eta^2) = -\frac{1}{2}\Bigg(\log|\hat{\Sigma}^{-1}| + \log|\Sigma| + m\log(2\pi\sigma_y^2) + \sum_j \log\left(\frac{\sigma_y^2 + m_j\sigma_\eta^2}{\sigma_y^2}\right) + \mu^T\Sigma^{-1}\mu + \frac{1}{\sigma_y^2}\sum_{i,j} y_{ij}^2 - \frac{1}{\sigma_y^2}\sum_j\left(\frac{\sigma_\eta^2}{\sigma_y^2+m_j\sigma_\eta^2}(\sum_i y_{ij})^2\right) - \hat{\mu}^T\hat{\Sigma}^{-1}\hat{\mu}\Bigg) \tag{3.14}$$

where the posterior mean $\hat{\mu}$ and posterior covariance $\hat{\Sigma}$ are functions of σ_y^2 and σ_η^2:

$$\hat{\Sigma}^{-1} = \Sigma^{-1} + \frac{1}{\sigma_y^2}\sum_{i,j} x_{ij}x_{ij}^T - \frac{1}{\sigma_y^2}\sum_j\left(\frac{\sigma_\eta^2}{\sigma_y^2+m_j\sigma_\eta^2}(\sum_i x_{ij})(\sum_q x_{qj}^T)\right)$$

$$\hat{\mu} = \hat{\Sigma}\left(\Sigma^{-1}\mu + \frac{1}{\sigma_y^2}\sum_{i,j} x_{ij}y_{ij} - \frac{1}{\sigma_y^2}\sum_j\left(\frac{\sigma_\eta^2}{\sigma_y^2+m_j\sigma_\eta^2}(\sum_i y_{ij})(\sum_q x_{qj})\right)\right)$$

Two-level model with group-varying coefficients. The more general two-level case of model $\mathcal{M}_{22}$ (Equation 3.4) has group-independent and group-varying coefficients for covariates x_{ij} and z_{ij} respectively. The integrated likelihood for this is:

$$p(\mathcal{D}|\mathcal{M}_{22}, \sigma_y^2, \phi) = \int_{\mathbb{R}^d} \int_{\mathbb{R}^{n\times d'}} \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\beta-\mu)^T\Sigma^{-1}(\beta-\mu)\right) \times \prod_j \frac{1}{(2\pi)^{d'/2}|\Sigma_\eta(\phi)|^{1/2}} \exp\left(-\frac{1}{2}\eta_j^T\Sigma_\eta^{-1}(\phi)\eta_j\right) \times \prod_{i,j} \frac{1}{(2\pi\sigma_y^2)^{1/2}} \exp\left(-\frac{(y_{ij}-\beta^T x_{ij}-\eta_j^T z_{ij})^2}{2\sigma_y^2}\right) d\eta d\beta$$

In this case, the posterior means $\hat{\Sigma}_{\eta,j}$ and posterior covariances $\hat{\mu}_{\eta,j}$ for group-level deviance terms η_j are different for each group. As usual, posterior means and covariances

are conditional on the model variance parameters σ_y^2 and ϕ:

$$\hat{\Sigma}_{\eta,j}^{-1} = \Sigma_\eta^{-1} + \frac{1}{\sigma_y^2}\sum_i z_{ij} z_{ij}^T, \quad \hat{\mu}_{\eta,j} = \hat{\Sigma}_{\eta,j}\left(\frac{1}{\sigma_y^2}\sum_i z_{ij}(y_{ij} - \beta^T x_{ij})\right)$$
$$\hat{\Sigma}^{-1} = \Sigma^{-1} + \frac{1}{\sigma_y^2}\sum_{i,j} x_{ij}x_{ij}^T - \frac{1}{\sigma_y^4}\sum_j \left((\sum_i x_{ij} z_{ij}^T)\hat{\Sigma}_{\eta,j}(\sum_q z_{qj} x_{qj}^T)\right)$$
$$\hat{\mu} = \hat{\Sigma}\left(\Sigma^{-1}\mu + \frac{1}{\sigma_y^2}\sum_{i,j} x_{ij} y_{ij} - \frac{1}{\sigma_y^4}\sum_j \left((\sum_i x_{ij} z_{ij}^T)\hat{\Sigma}_{\eta,j}(\sum_q z_{qj} y_{qj})\right)\right)$$

The log integrated likelihood in this instance is:

$$\begin{aligned}\log p(\mathcal{D}|\mathcal{M}_{22}, \sigma_y^2, \phi) = -\frac{1}{2}\Bigg(&\log|\hat{\Sigma}^{-1}| + \log|\Sigma| + m\log(2\pi\sigma_y^2) + n\log|\Sigma_\eta| \\ &+ \sum_j \log|\hat{\Sigma}_{\eta,j}^{-1}| + \mu^T\Sigma^{-1}\mu + \frac{1}{\sigma_y^2}\sum_{i,j} y_{ij}^2 \\ &- \frac{1}{\sigma_y^4}\sum_j \left((\sum_i z_{ij}^T y_{ij})\hat{\Sigma}_{\eta,j}(\sum_q z_{qj} y_{qj})\right) - \hat{\mu}^T\hat{\Sigma}^{-1}\hat{\mu}\Bigg)\end{aligned} \tag{3.15}$$

Three-level hierarchical structure. Finally, in a three-level hierarchical structure with the usual $N(\mu, \Sigma)$ prior for β, the integrated likelihood for model $\mathcal{M}_{31}$ is:

$$\begin{aligned}p(\mathcal{D}|\mathcal{M}_{31}, \sigma_y^2, \sigma_\eta^2, \sigma_\zeta^2) = \int_{\mathbb{R}^{d+m+K}} &\frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}}\exp\left(-\frac{1}{2}(\beta-\mu)^T\Sigma^{-1}(\beta-\mu)\right) \times \\ &\prod_k \frac{1}{(2\pi\sigma_\zeta^2)^{1/2}}\exp\left(-\frac{\zeta_k^2}{2\sigma_\zeta^2}\right)\prod_{j,k}\frac{1}{(2\pi\sigma_\eta^2)^{1/2}}\exp\left(-\frac{\eta_{jk}^2}{2\sigma_\eta^2}\right) \times \\ &\prod_{i,j,k}\frac{1}{(2\pi\sigma_y^2)^{1/2}}\exp\left(-\frac{(y_{ijk} - \beta^T x_{ijk} - \eta_{jk} - \zeta_k)^2}{2\sigma_y^2}\right) d\zeta d\eta d\beta\end{aligned}$$

The log integrated likelihood is:

$$\begin{aligned}\log p(\mathcal{D}|\mathcal{M}_{31},\sigma_y^2,\sigma_\eta^2,\sigma_\zeta^2) = \frac{1}{2}\Bigg(&- \log|\hat{\Sigma}^{-1}| - \log|\Sigma| - m\log(2\pi\sigma_y^2) \\ &+ \sum_k \log\left(\frac{\sigma_y^2}{\sigma_y^2+\sigma_\zeta^2\sum_p(\sigma_y^2 m_{pk}/(\sigma_y^2+\sigma_\eta^2 m_{pk}))}\right) \\ &+ \sum_{j,k}\log\left(\frac{\sigma_y^2}{\sigma_y^2+\sigma_\eta^2 m_{jk}}\right) - \mu^T\Sigma^{-1}\mu + \hat{\mu}^T\hat{\Sigma}^{-1}\hat{\mu} \\ &- \frac{1}{\sigma_y^2}\sum_{i,j,k} y_{ijk}^2 + \frac{1}{\sigma_y^2}\sum_{j,k}\frac{\sigma_\eta^2}{\sigma_y^2+\sigma_\eta^2 m_{jk}}(\sum_i y_{ijk})^2 \\ &+ \frac{1}{\sigma_y^2}\sum_k \frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2\sum_l(\sigma_y^2 m_{lk}/(\sigma_y^2+\sigma_\eta^2 m_{lk}))}\times \\ &\qquad (\sum_j \frac{\sigma_y^2}{\sigma_y^2+\sigma_\eta^2 m_{jk}}\sum_i y_{ijk})^2\Bigg)\end{aligned} \tag{3.16}$$

where the posterior mean and covariance for β (conditional upon σ_y^2, σ_η^2 and σ_ζ^2) are:

$$\begin{aligned}\hat{\Sigma} = \Bigg(&\Sigma^{-1} + \frac{1}{\sigma_y^2}\sum_{i,j,k} x_{ijk}x_{ijk}^T - \frac{1}{\sigma_y^2}\sum_{j,k}\left(\frac{\sigma_\eta^2}{\sigma_y^2+\sigma_\eta^2 m_{jk}}(\sum_i x_{ijk})(\sum_q x_{qjk}^T)\right) \\ &-\frac{1}{\sigma_y^2}\sum_k \frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2\sum_l(\sigma_\eta^2 m_{lk}/(\sigma_y^2+\sigma_\eta^2 m_{lk}))}\times \\ &\qquad (\sum_p \frac{\sigma_y^2}{\sigma_y^2+\sigma_\eta^2 m_{pk}}\sum_q x_{qpk})(\sum_j \frac{\sigma_y^2}{\sigma_y^2+\sigma_\eta^2 m_{jk}}\sum_i x_{ijk}^T)\Bigg)^{-1}\end{aligned}$$

$$\begin{aligned}\hat{\mu} = \hat{\Sigma}\Bigg(&\Sigma^{-1}\mu + \frac{1}{\sigma_y^2}\sum_{i,j,k} x_{ijk}y_{ijk} - \frac{1}{\sigma_y^2}\sum_{j,k}\left(\frac{\sigma_\eta^2}{\sigma_y^2+\sigma_\eta^2 m_{jk}}(\sum_i x_{ijk})(\sum_q y_{qjk})\right) \\ &-\frac{1}{\sigma_y^2}\sum_k \frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2\sum_l(\sigma_\eta^2 m_{lk}/(\sigma_y^2+\sigma_\eta^2 m_{lk}))}\times \\ &\qquad (\sum_p \frac{\sigma_y^2}{\sigma_y^2+\sigma_\eta^2 m_{pk}}\sum_q x_{qpk})(\sum_j \frac{\sigma_y^2}{\sigma_y^2+\sigma_\eta^2 m_{jk}}\sum_i y_{ijk})\Bigg)\end{aligned}$$

I did not consider this three-level model in my paper on this topic [22], but I included the derivation in Appendix B.2.

3.2.3 Simulation study example

In [22], I illustrated that using the integrated likelihood with SMC sampling yielded improved model evidence estimates, via simulated datasets from the modular time-series framework (Equation 3.7) corresponding to models $\mathcal{M}_{11}$, $\mathcal{M}_{12}$, $\mathcal{M}_{21}$ and $\mathcal{M}_{22}$. For all four

models, I generated a simulated dataset where each model was the 'true' model in turn, i.e. $\mathcal{D}_{11}$, $\mathcal{D}_{12}$, $\mathcal{D}_{21}$ and $\mathcal{D}_{22}$. I then estimated the model evidence using SMC sampling for all four models on all four datasets, first with the integrated likelihood (sampling variance parameters) and with the full likelihood (sampling all model parameters).

In this simulation study, the data and models had at most a two-level structure. To generate the datasets, I first simulated multilevel group structure and timepoints t_{ij}. The timepoints then defined covariates x_{ij}. I defined a (centred) subset of the covariates as z_{ij}, the group-varying coefficients in $\mathcal{M}_{22}$. The group structure and covariates were shared across all datasets, even when the underlying structure was not explicitly used. For each dataset, I sampled 'true' model parameters (including 'true' Gaussian noise), which were then regarded as fixed. I denoted the true model parameters with the following notation changes: $\beta \to b$, $\sigma^2 \to s^2$, $\eta_j \to h_j$ and $\epsilon_{ij} \to e_{ij}$. I computed the outcome variable y_{ij} as defined by the corresponding model. Together, the multilevel structure, covariates and outcome variable formed the four datasets that corresponded to each model.

In all datasets, I fixed the number of groups as $n = 15$ and the number of individual datapoints as $m = 1000$. To assign multilevel group membership (with unequal group sizes), I sampled the indices $j \in \{1, \dots, n\}$ m times with replacement, with probability p_j from a Dirichlet distribution with parameter $\alpha = (2, \dots, n+1)$. The probabilities p_j satisfied $\sum_j p_j = 1$ and $\mathbb{E}[p_j] = (j+1)/n^2$. For all datasets, the simulated data also included:

$$
\begin{aligned}
&t_{ij} \sim U[0,1] \qquad (3.17)\\
&d_1 = 5, \quad s = (0, 0.2, 0.4, 0.6, 0.8), \quad P = 1, \quad d_2 = 20, \quad d = 46\\
&x_{ij}^T = (1, t_{ij}, (t_{ij} - 0.2)\,\mathbb{1}\{t_{ij} > 0.2\}, \dots, \cos(2\pi t_{ij}), \dots, \sin(2\pi t_{ij}), \dots \sin(6\pi t_{ij}))\\
&z_{ij}^T = (1, t_{ij} - 0.5, (t_{ij} - 0.4)\,\mathbb{1}\{t_{ij} > 0.4\} - 0.18, (t_{ij} - 0.8)\,\mathbb{1}\{t_{ij} > 0.8\} - 0.02)
\end{aligned}
$$

The constants added to z_{ij} were such that $\mathbb{E}[z_{ij}^T] = (1, 0, 0, 0)$. I specified a covariance hyperparameter S for simulating 'true' coefficients b from a multivariate Gaussian distribution:

$$
S_1 = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0\\
0 & 4 & -3 & -1 & 0 & 0\\
0 & -3 & 5 & -4 & 2 & 0\\
0 & -1 & -4 & 10 & -4 & 0\\
0 & 0 & 2 & -4 & 5 & 2\\
0 & 0 & 0 & 0 & 2 & 6
\end{pmatrix}, \; \lambda = 0.001, \; S = \begin{pmatrix} S_1 & 0 \\ 0 & \lambda I \end{pmatrix}
$$

The elements of S were chosen to encourage a reasonable amount of 'wiggliness' in $f(t_{ij})$, while also ensuring S was $d \times d$ positive-definite. In particular, for $k = 2, \dots, 5$, the 'true'

parameter b_{k+1} only specified a non-smooth change in gradient at the changepoint s_k, so the covariance structure of S_1 was chosen to encourage anti-correlated gradients in successive changepoint intervals. Similarly, the Fourier element weight λ was small to ensure the Fourier terms of Equation 3.7 did not dominate the piece-wise linear component.

Simulated datasets. All four simulated datasets are shown in Figure 3.1 and are available in the repository at [139]. These were defined as follows:

$$\mathcal{D}_{11}: \; b \sim N(0, S), \quad s^2 \sim IG(3, 0.4), \quad e_{ij} \sim N(0, s^2)$$
$$y_{ij}^{(11)} = b^T x_{ij} + e_{ij}, \quad \mathcal{D}_{11} = (y_{ij}^{(11)}, x_{ij}, z_{ij})$$

$$\mathcal{D}_{12}: \; s^2 \sim IG(3, 0.4), \quad \gamma = 5 = 1/\,\mathbb{E}[s^2], \quad b|s^2 \sim N(0, \gamma s^2 S), \quad e_{ij} \sim N(0, s^2)$$
$$y_{ij}^{(12)} = b^T x_{ij} + e_{ij}, \quad \mathcal{D}_{12} = (y_{ij}^{(12)}, x_{ij}, z_{ij})$$

$$\mathcal{D}_{21}: \; b \sim N(0, S), \quad s_y^2 \sim IG(3, 0.3), \quad s_h^2 \sim IG(3, 0.1), \quad e_{ij} \sim N(0, s_y^2), \quad h_j \sim N(0, s_h^2)$$
$$y_{ij}^{(21)} = b^T x_{ij} + h_j + e_{ij}, \quad \mathcal{D}_{21} = (y_{ij}^{(21)}, x_{ij}, z_{ij})$$

$$\mathcal{D}_{22}: \; b \sim N(0, S), \quad s_y^2 \sim IG(3, 0.3), \quad s_{h,1}^2, s_{h,2}^2, s_{h,3}^2, s_{h,4}^2 \sim IG(3, 0.1), \quad \rho = 0.2$$
$$e_{ij} \sim N(0, s_y^2), \quad h_j \sim N(0, S_h), \quad S_h = \begin{pmatrix} s_{h,1}^2 & 0 & 0 & 0 \\ 0 & s_{h,2}^2 & \rho s_{h,2} s_{h,3} & 0 \\ 0 & \rho s_{h,2} s_{h,3} & s_{h,3}^2 & \rho s_{h,3} s_{h,4} \\ 0 & 0 & \rho s_{h,3} s_{h,4} & s_{h,4}^2 \end{pmatrix}$$
$$y_{ij}^{(22)} = b^T x_{ij} + h_j^T z_{ij} + e_{ij}, \quad \mathcal{D}_{22} = (y_{ij}^{(22)}, x_{ij}, z_{ij})$$

In $\mathcal{D}_{12}$, I added the factor γ so that $\mathbb{E}[\beta] = 0$, $\mathrm{Cov}(\beta) = S$ under the joint NIG prior for b and s^2. In each dataset, the outcome variable y_{ij} should have similar expected value and variance, because the expected value of $IG(a, b)$ is $b/(a-1)$ and because η_j and ϵ_{ij} are independent. As a result, $\mathbb{E}_{t,\theta}[\mathbb{E}[y_{ij}|t_{ij}, \theta]] = 0$ and $\mathbb{E}_{t,\theta}[\mathrm{var}(y_{ij} - b^T x_{ij}|t_{ij}, \theta)] = 0.2$ in each case. This is important because it meant the choice of prior hyperparameters should not unduly influence the model evidence (at least, relative to the model structure), when all models were tested against a given dataset.

Models. I then specified priors for each model that were similar to the distributions from which the 'true' coefficients were drawn. In place of the covariance matrix S, the prior for β had covariance Σ, which shared the diagonal terms of S but was zero elsewhere:

$$\Sigma_1 = \mathrm{diag}(1, 4, 5, 10, 5, 6), \;\; \lambda = 0.001, \;\; \Sigma = \begin{pmatrix} \Sigma_1 & 0 \\ 0 & \lambda I \end{pmatrix}$$

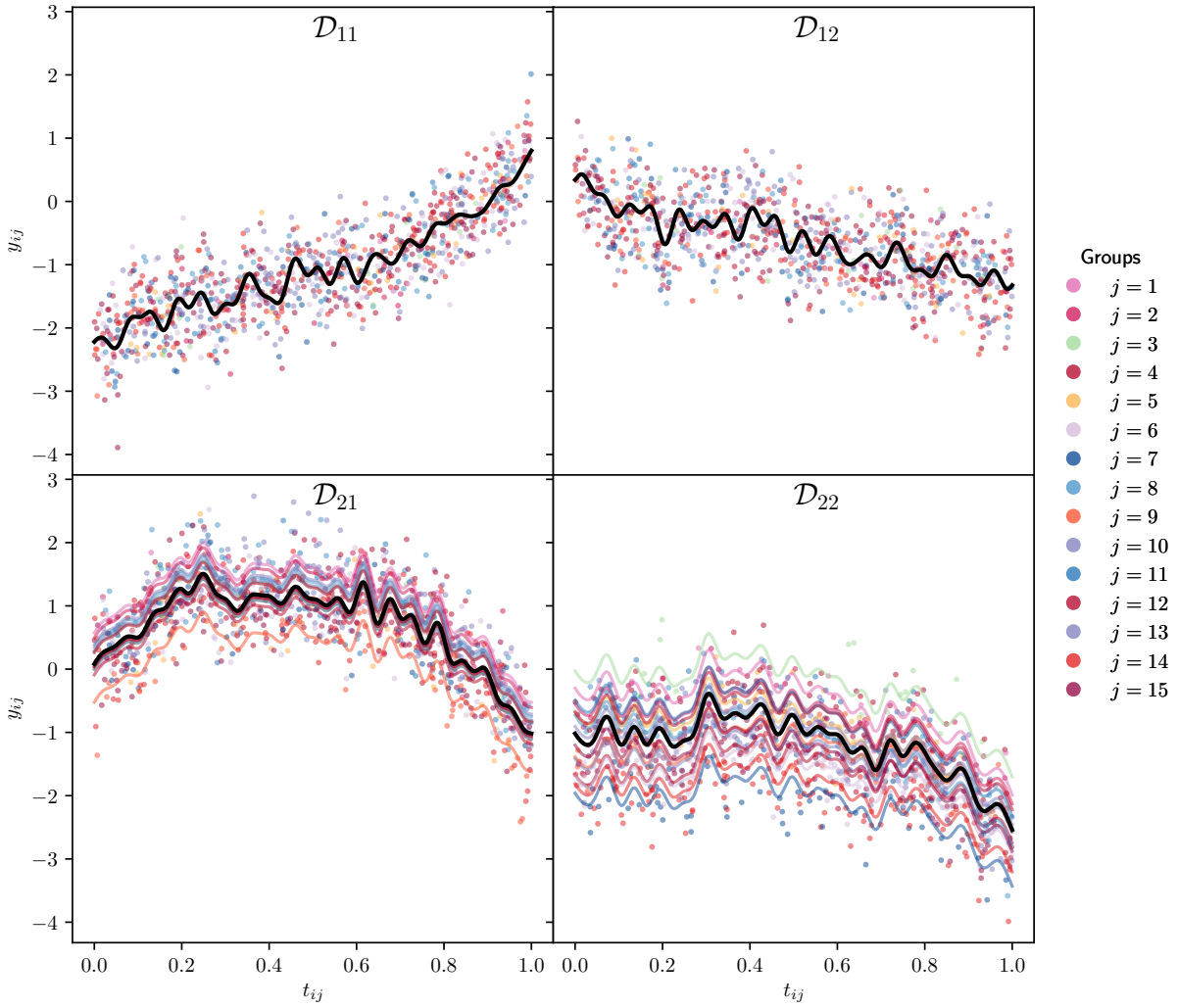

Figure 3.1: Simulated datasets, $\mathcal{D}_{11}, \mathcal{D}_{12}, \mathcal{D}_{21}, \mathcal{D}_{22}$. Each dot represented a single datapoint. The covariate function $x = (f_1(t), \ldots, f_J(t))$ was a deterministic multi-valued function of t. I included the line $b^T x$ for all values of $t \in [0, 1]$ in each subfigure. For $\mathcal{D}_{21}$, I also included the lines $b^T x + h_j$ for $j = 1, \ldots, n = 15$. For $\mathcal{D}_{22}$, I included the lines $b^T x + h_j^T z$, where z was a subset of x as defined in Equation 3.17.

The models were then as follows:

$$\begin{aligned}
\mathcal{M}_{11}: \; & y_i = \beta^T x_i + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2) \\
& \beta \sim N(0, \Sigma), \quad \sigma^2 \sim IG(3, 0.4)
\end{aligned}$$

$$\begin{aligned}
\mathcal{M}_{12}: \; & y_i = \beta^T x_i + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2) \\
& \sigma^2 \sim IG(3, 0.4), \quad \gamma = 5, \quad \beta | \sigma^2 \sim N(0, \gamma \sigma^2 \Sigma)
\end{aligned}$$

$$\begin{aligned}
\mathcal{M}_{21}: \; & y_{ij} = \beta^T x_{ij} + \eta_j + \epsilon_{ij}, \quad \epsilon_{ij} \sim N(0, \sigma_y^2), \quad \eta_j \sim N(0, \sigma_\eta^2) \\
& \beta \sim N(0, \Sigma), \quad \sigma_y^2 \sim IG(3, 0.4), \quad \sigma_\eta^2 \sim IG(3, 0.1)
\end{aligned}$$

$$\begin{aligned}
\mathcal{M}_{22}: \; & y_{ij} = \beta^T x_{ij} + \eta_j^T z_{ij} + \epsilon_{ij}, \quad \epsilon_{ij} \sim N(0, \sigma_y^2), \quad \eta_j \sim N(0, \Sigma_\eta) \\
& \beta \sim N(0, \Sigma), \quad \sigma_y^2 \sim IG(3, 0.3), \quad \sigma_{\eta,1}^2, \sigma_{\eta,2}^2, \sigma_{\eta,3}^2, \sigma_{\eta,4} \sim IG(3, 0.1), \quad \rho = 0.2 \\
& \Sigma_\eta = \begin{pmatrix} \sigma_{\eta,1}^2 & 0 & 0 & 0 \\ 0 & \sigma_{\eta,2}^2 & \rho\sigma_{\eta,2}\sigma_{\eta,3} & 0 \\ 0 & \rho\sigma_{\eta,2}\sigma_{\eta,3} & \sigma_{\eta,3}^2 & \rho\sigma_{\eta,3}\sigma_{\eta,4} \\ 0 & 0 & \rho\sigma_{\eta,3}\sigma_{\eta,4} & \sigma_{\eta,4}^2 \end{pmatrix}
\end{aligned}$$

Results. For each dataset, we would expect the model with 'true' structure to have the largest model evidence. Table 3.1 shows the performance of all four models on all datasets. SMC sampling with the integrated likelihood correctly identified the 'true' model for each dataset, and resulted in decreased uncertainty in model evidence estimates. In contrast, SMC sampling with the full likelihood misspecified the correct model in two of the four cases. For model $\mathcal{M}_{12}$, the model evidence could be directly calculated, so the bias of each SMC sampling approach can be evaluated. Model evidence estimates were more accurate using the integrated likelihood approach than using the full likelihood approach. The bias was not possible to directly evaluate for the other models, where there was no exact solution. One approach for estimating the bias here is to increase the number of particles in the SMC sampling, as the SMC estimate of the model evidence should converge as the number of particles tends to infinity. However, the main problem with this is that the computational cost gets prohibitively expensive, and unfortunately it was not possible to complete the computations for all models. However, some initial results for model $\mathcal{M}_{12}$ showed a small reduction in the bias when the number of particles was doubled from 8 to 16. In every instance, the computational cost of sampling variance parameters with the integrated likelihood was smaller than the cost of sampling all parameters with the full likelihood. Finally, in [22], I used the Mahalabonis distance to compare posterior distributions for coefficients β against the 'true' coefficients b for every pair of dataset and model. I have omitted the details of this here but, in every case, the 'true' coefficient was

closer to the posterior distribution when using the integrated likelihood approach than when using the full likelihood approach.

	SMC with integrated likelihood (sampling variance parameters)				
	$\log p(\mathcal{D}_{11}\|\mathcal{M})$	$\log p(\mathcal{D}_{21}\|\mathcal{M})$	$\log p(\mathcal{D}_{22}\|\mathcal{M})$	$\log p(\mathcal{D}_{12}\|\mathcal{M})$	Time (s)
$\mathcal{M}_{11}$	-633.08 (0.03)	-684.87 (0.05)	-753.53 (0.05)	-908.24 (0.07)	6.0
$\mathcal{M}_{12}$	-633.10 (0.05)	-684.87 (0.03)	-753.50 (0.04)	-909.22 (0.03)	2.3
$\mathcal{M}_{21}$	-642.08 (0.04)	-694.88 (0.06)	-681.06 (0.02)	-518.24 (0.04)	8.7
$\mathcal{M}_{22}$	-644.66 (0.05)	-697.33 (0.06)	-683.43 (0.03)	-508.96 (0.06)	37.3
	SMC with full likelihood (sampling all parameters)				
	$\log p(\mathcal{D}_{11}\|\mathcal{M})$	$\log p(\mathcal{D}_{12}\|\mathcal{M})$	$\log p(\mathcal{D}_{21}\|\mathcal{M})$	$\log p(\mathcal{D}_{22}\|\mathcal{M})$	Time, s
$\mathcal{M}_{11}$	-633.34 (0.15)	-685.00 (0.22)	-753.79 (0.18)	-908.29 (0.22)	13.8
$\mathcal{M}_{12}$	-632.95 (0.26)	-684.63 (0.21)	-753.56 (0.19)	-909.44 (0.18)	45.8
$\mathcal{M}_{21}$	-643.23 (0.24)	-695.64 (0.26)	-681.37 (0.32)	-526.05 (2.69)	19.1
$\mathcal{M}_{22}$	-647.68 (1.07)	-700.39 (1.05)	-686.42 (0.99)	-532.39 (6.75)	63.2
	Fully analytical solution				
	$\log p(\mathcal{D}_{11}\|\mathcal{M})$	$\log p(\mathcal{D}_{12}\|\mathcal{M})$	$\log p(\mathcal{D}_{21}\|\mathcal{M})$	$\log p(\mathcal{D}_{22}\|\mathcal{M})$	
$\mathcal{M}_{12}$	-633.08	-684.87	-753.47	-909.23	

Table 3.1: Comparison of model evidence for each simulated dataset and model. This was computed using sequential Monte Carlo (SMC), with the integrated likelihood and with the full likelihood. The results shown are the mean and standard deviation over 8 random initialisations in SMC sampling. For both approaches and for each dataset, the model with the strongest evidence was highlighted. For each model, I also reported the time taken to complete SMC sampling, averaged over different datasets and initialisations.

3.3 Comparing trajectories between cohorts

As previously stated, the original goal was to compare the trajectories between two cohorts of patients from ICU. The first cohort of patients were those admitted to ICU with respiratory viral infection from Covid-19, as described earlier in Section 2.3.1. In Section 2.3, I calculated information-theoretic measures associated with key physiological variables for this cohort, but it was difficult to draw any conclusions about this, firstly without modelling these trajectories as a function of time and secondly without similar knowledge of information trajectories in more 'routine' ICU patients. Heterogeneity in patient conditions within ICU meant that it was difficult to define a suitable baseline cohort, so instead I chose a more directly comparable cohort of ICU patients who were admitted with sepsis and severe respiratory infection.

3.3.1 Amsterdam University Medical Centers database

The Amsterdam University Medical Centers database (AmsterdamUMCdb) is a recently-published freely-accessible ICU database [15], and the first large-scale European ICU database released in compliance with both European and US data regulations. This database contains close to 1 billion datapoints from 20,109 critically ill patients admitted to Amsterdam UMC between 2003 and 2016, and consists of patients admitted to both ICU and to a 'medium care unit' (MCU). The median length of stay in AmsterdamUMCdb was 26hrs, with ICU mortality of 9.9%. Age was categorised by decade, with the median age between 60 and 69. Within a dictionary of 9031 clinical parameters, there were approximately 8 million temperature measurements, 33 million ABP measurements and 38 million HR measurements, averaging 360, 1450 and 1600 measurements per patient respectively. The database was de-anonymised through risk-based patient de-identification, and contains demographics, routinely-measured physiological vitals, administered drugs, laboratory results, diagnoses and procedures. AmsterdamUMCdb has already been the focus of several multidisciplinary research events, including datathons hosted by the European Society of Intensive Care Medicine (ESICM) [140].

Sepsis is a leading cause of mortality in ICU and is caused by a dysregulated biochemical, physiological and immune response to infection resulting in multiple organ dysfunction. It is generally difficult to quantify its incidence and mortality rate in ICU, as it is a heterogeneous syndrome characterised by wide-ranging infectious agents, infection site, treatment history and host response. Sepsis can be acquired in the community, in hospital or in ICU [141], and therefore it may not always be the primary reason for ICU admission. As such, there was no consistent sepsis diagnosis among the severity scores and diagnoses included in AmsterdamUMCdb. As a result, I developed an open-source implementation of the Sepsis-3 criteria for AmsterdamUMCdb [112]. The Sepsis-3 criteria involves a disease severity score, Sequential Organ Failure Assessment (SOFA), which grades a set of six physiological systems between 0 and 4. For the purposes of this chapter, the Sepsis-3 criteria was only required to identify a subset of patients with a sepsis diagnosis and respiratory failure. A patient has SOFA respiration score >2 if they have poor oxygenation (hypoxemia) and ventilatory support. I have described Sepsis-3 and SOFA in more detail later in Section 5.4, as the clinical Sepsis-3 definition was more relevant to my thesis in that chapter.

Using the Sepsis-3 criteria, I defined an initial cohort of 1761 patients with sepsis at ICU admission, ICU length of stay >48hrs and either (i) a SOFA respiration score of >2, (ii) a medical specialty admission relating to lung disease or respiratory-related surgery, or (iii) a clinical diagnosis of pneumonia or respiratory failure. As before, the workflow for building the full time-series dataset included estimation of information-theoretic measures between pairs of physiological variables recorded at 1 minute intervals over 24hr windows.

I required at least 24hrs of continuous monitoring of each variable (temperature and heart rate) with measurements recorded at 1 minute intervals. I followed the same preprocessing steps as for the Covid-19 cohort, removing outlier values in the data and excluding patients who (i) did not have at least 24hrs of data or (ii) for whom $<90\%$ of measurement intervals were 1 minute or less. Under these criteria, I identified a subset of 176 patients who were admitted to ICU with sepsis and respiratory failure, and who had sufficient quantity of data. Of these, 54 died in ICU with median length of stay 10.0 days, and 122 were discharged with median length of stay 10.7 days. Restricting to a maximum of 15 days after ICU admission, 42 patients died in ICU, 74 were discharged and 60 were still in ICU at the end of the 15 days.

For each patient in the cohort, I added precision-level uniform noise to each variable, then estimated the entropy, mutual information and (bidirectional) transfer entropy between temperature, HR and ABP across 24hr windows, which contained up to 1440 minute-by-minute measurements. I repeated the 24hr information estimation at uniformly-spaced 6hr increments, from ICU admission until discharge or death. I used the KSG algorithm for estimation (Equations 2.8, 2.9 and 2.10). I discarded any 24hr window that did not contain at least 200 multivariate measurements, as the KSG algorithm performed poorly when there were too few datapoints. In addition to the information trajectories, I also modelled the trajectories of lab result values for two markers of inflammation, C-reactive protein level and WBC count. These were both recorded more infrequently, typically with one measurement per day, so I used the raw data values in this instance. Unlike the information-theoretic measures, the timepoints for these data were unique to individual patients and irregularly spaced.

3.3.2 Cohort datasets

I modelled each inflammatory marker or information-theoretic measure separately, as univariate functions of time. For each dependent variable, I defined a trio of datasets, the Covid-19 cohort $\mathcal{D}_1$, the sepsis cohort $\mathcal{D}_2$ and a combined cohort with all datapoints $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2$. To avoid over-complicating the notation further, I used y_{ijk} (and the datasets $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}$) to interchangeably represent just one inflammatory marker or information-theoretic measure at any given point during the rest of this chapter. This should be clear in context. The number of measurements, patients and hospitals in all datasets (i.e. for each variable) is summarised in Table 3.2. For each variable, I modelled time-series trajectories using both Bayesian and frequentist frameworks, which had different basis functions, model fitting and model comparison approaches. The Bayesian approach was described in Section 3.2. In the baseline model (Equation 3.10), both disease cohorts (sepsis and Covid-19) belonged to a single unified dataset and all model parameters were shared, i.e. I modelled the combined cohort. In the alternate model (Equation 3.11), each

	Cohorts		
	Covid-19, $\mathcal{D}_1$	Sepsis, $\mathcal{D}_2$	Combined, $\mathcal{D}$
Number of patients, n	136	172	312
Number of hospitals, K	4	1	5
Number of measurements, m			
Entropy, T	4197	3882	8079
Entropy, HR	5836	6762	12598
Entropy, ABP	5795	6646	12441
MI, T and HR	4194	3874	8068
MI, T and ABP	4189	3857	8046
MI, HR and ABP	5794	6627	12421
CRP	1682	1825	3507
Leukocytes	1780	2083	3871

Table 3.2: Dataset sizes for information-theoretic measures and inflammatory markers. For each pair of physiological time-series, the number of transfer entropy measurements (in each direction) was the same as the number of mutual information measurements. For information-theoretic measures, there was a maximum of 57 measurements per patient. For inflammatory markers, there was a maximum of 24 measurements per patient.

disease cohort was modelled independently (under the same basis functions). I estimated the (log) model evidence for both baseline and alternate models using SMC with integrated likelihoods (Equations 3.14 and 3.16), and compared the baseline and alternate models using the Bayes factor. Further details about this (including model priors) is given in Appendix B.3. The frequentist approach used GAMMs, along with the hypothesis test defined in Equation 3.6. In this paradigm, the sepsis and Covid-19 cohorts were modelled independently and a test statistic involved the integrated squared difference between their time-series trajectory functions (Equation 3.6). Under a null hypothesis of no difference between cohorts, this integral is equal to zero. I included further details relating to this model setup in Appendix B.3. If there was a significant difference in the trajectories between cohorts, then the alternate model should be favoured in the Bayesian paradigm and the null hypothesis should be rejected at some appropriate α-level in the frequentist paradigm. As there were explicit differences in the model setups between Bayesian and frequentist paradigms, there was no reason to expect full agreement between these two approaches, though less similarity between cohort should make it more likely that both rejected the baseline.

I evaluated the model evidence for each dataset, first including only second-level patient groups (i.e. patients had multiple measurements, and each patient was modelled with a separate intercept term) and then repeating this with both second-level patient groups and third-level hospital groups (apart from the sepsis cohort, which did not contain any third-level groups). These results are detailed in Tables 3.3 and 3.4. In the two-level datasets, there was decisive evidence in favour of the alternate model in the Bayesian approach for

every information-theoretic measure and inflammatory marker, but not universal rejection of the frequentist null hypothesis at the level $\alpha = 0.05$. In the three-level model, the alternate model was favoured and the null hypothesis rejected for most variables, but this was also not consistent. In particular, moving from two-level models to three-level models altered some results. For mutual information between temperature and HR, the addition of third-level hospital groups changed the Bayes factor from decisive in favour of the alternate model to substantial in favour of the baseline model. This suggested that there was enough group-level variation between hospitals to explain the differences in value between cohorts as a hospital-level deviation rather than as a fundamental cohort-wide difference. Similar effects were observed in the (frequentist) p-values for transfer entropies. Finally, I evaluated the model evidence for three-level models for models without time-dependence, i.e. with the function $f(t)$ replaced by a single intercept term. In Table B.1, I compared this to results from Table 3.4, to test whether the trajectories were indeed time-varying or whether they were fully independent of time. There was decisive evidence in favour of the full time-varying model in almost every case, except for transfer entropy from ABP to T and transfer entropy from HR to T. Only once was there evidence in favour the intercept-only model (in the combined cohort, for transfer entropy from ABP to T). These results suggested that there were time-dependent relationships in the cohort-average information trajectories.

Figures 3.2 and 3.4 show Bayesian model fits for the information trajectories. These model fits use posterior means for model parameters, adjusted by group-level intercept deviations. In these figures, I provided highest density intervals (HDIs) for the time-series trajectory function $f(t)$ alone (via the posterior for regression coefficients β), and for $f(t)$ with the group-level deviations (via the posteriors for β and deviations η and ζ). HDIs are a type of credible interval. The latter HDI was vastly wider than the former, which was often tight and could not be easily distinguished from the model fit itself. The HDI for $f(t)$ without group-level effects was also consistently wider in a three-level model (Covid-19 cohort and combined cohort) than for a two-level model (sepsis cohort). This was unsurprising, because the results in Section 3.2 show that the posterior covariance for β in a three-level model should theoretically be larger than for a two-level model (in the three-level model, the inverse of the posterior covariance contains an additional negative term and the posterior covariance has larger determinant). Figures 3.3 and 3.5 show frequentist GAMM model fits. In these figures, I used bootstrap resampling to estimate confidence intervals for $f(t)$ alone, in which both patient groups and individual measurements were resampled multiple times before repeated model fitting. To estimate HDIs for $f(t)$ and the group-level deviations combined, I computed the lower and upper HDI limits of the group-level intercept deviations and added these to the previous lower and upper confidence interval bounds respectively. One point to emphasise in these figures

Two-level models (with group-level patient terms)

	SMC with integrated likelihoods					GAMM	
	Baseline model	Alternate model					
Variable	$\log p(\mathcal{D}\|\mathcal{M}_b)$	$\log p(\mathcal{D}\|\mathcal{M}_a)$	$\log p(\mathcal{D}_1\|\mathcal{M}_a^{(1)})$	$\log p(\mathcal{D}_2\|\mathcal{M}_a^{(2)})$	$\log_{10} \mathrm{BF}_{ab}$	Δ_{obs}	p-value
Entropy, T	-8761.13 (0.02)	-8719.24 (0.05)	-4748.93 (0.03)	-3970.31 (0.03)	18.19	0.029	0.048
Entropy, HR	-13334.46 (0.03)	-13222.52 (0.06)	-5627.25 (0.04)	-7595.27 (0.05)	48.62	0.072	<0.001
Entropy, ABP	-14378.94 (0.02)	-14264.09 (0.05)	-6299.60 (0.03)	-7964.49 (0.02)	49.88	0.127	<0.001
Mutual information, HR and T	-9443.50 (0.03)	-9418.46 (0.06)	-4938.32 (0.03)	-4480.15 (0.03)	10.87	0.298	<0.001
Mutual information, ABP and T	-9938.40 (0.03)	-9926.57 (0.04)	-5214.25 (0.03)	-4712.32 (0.02)	5.14	0.142	<0.001
Mutual information, ABP and HR	-14007.91 (0.02)	-13907.95 (0.05)	-6628.24 (0.04)	-7279.71 (0.04)	43.42	0.276	<0.001
Transfer entropy, HR to T	-10111.38 (0.03)	-10102.05 (0.03)	-5332.84 (0.03)	-4769.21 (0.03)	4.05	0.061	<0.001
Transfer entropy, T to HR	-10248.87 (0.04)	-10221.43 (0.04)	-5119.66 (0.03)	-5101.77 (0.02)	11.92	0.044	0.005
Transfer entropy, ABP to T	-10020.79 (0.04)	-10003.26 (0.05)	-5006.30 (0.04)	-4996.96 (0.04)	7.61	0.005	0.310
Transfer entropy, T to ABP	-10235.45 (0.03)	-10230.12 (0.05)	-5362.35 (0.04)	-4867.77 (0.03)	2.31	0.033	0.017
Transfer entropy, ABP to HR	-16320.68 (0.02)	-16316.05 (0.04)	-7369.94 (0.03)	-8946.10 (0.04)	2.01	0.004	0.292
Transfer entropy, HR to ABP	-16146.54 (0.03)	-16121.08 (0.05)	-7763.12 (0.03)	-8357.96 (0.03)	11.06	0.078	<0.001
C-reactive protein (mg/l)	-3999.41 (0.04)	-3993.24 (0.06)	-1813.29 (0.02)	-2179.95 (0.04)	2.68	0.029	0.078
Leukocytes (10^9/l)	-4155.36 (0.03)	-3718.17 (0.04)	-1049.96 (0.03)	-2668.20 (0.04)	189.87	0.197	<0.001

Table 3.3: Bayes factors and frequentist p-values for model comparisons of all information trajectories and inflammatory marker trajectories. This involved two-level models, with second-level patient groups but without third-level hospital groups (compared to three-level models in Table 3.4). The Bayes factors were calculated using log model evidence estimates from sequential Monte Carlo (SMC) with integrated likelihoods (Equations 3.10, 3.11, 3.14, 3.16). These are colour-coded according to the interpretation table of Jeffrey [128], with decisive in favour of $\mathcal{M}_a$ (greater than 2). The p-values are from the GAMM hypothesis test (Equation 3.6) and are colour-coded with $p > 0.1$, $p < 0.1$, $p < 0.05$ and $p < 0.01$.

Three-level models (with group-level patient and hospital terms)

	SMC with integrated likelihoods					GAMM	
	Baseline model	Alternate model					
Variable	$\log p(\mathcal{D}\|\mathcal{M}_b)$	$\log p(\mathcal{D}\|\mathcal{M}_a)$	$\log p(\mathcal{D}_1\|\mathcal{M}_a^{(1)})$	$\log p(\mathcal{D}_2\|\mathcal{M}_a^{(2)})$	$\log_{10} \mathrm{BF}_{ab}$	Δ_{obs}	p-value
Entropy, T	-8761.78 (0.02)	-8715.44 (0.05)	-4747.09 (0.03)	-3968.35 (0.04)	20.13	0.115	0.004
Entropy, HR	-13325.86 (0.03)	-13221.09 (0.04)	-5626.12 (0.02)	-7594.97 (0.03)	45.51	0.117	<0.001
Entropy, ABP	-14372.46 (0.04)	-14264.73 (0.04)	-6300.23 (0.02)	-7964.50 (0.04)	46.79	0.380	<0.001
Mutual information, HR and T	-9412.71 (0.04)	-9414.92 (0.05)	-4933.39 (0.03)	-4481.53 (0.03)	-0.96	0.522	<0.001
Mutual information, ABP and T	-9931.20 (0.03)	-9928.21 (0.04)	-5215.88 (0.03)	-4712.34 (0.04)	1.30	0.477	<0.001
Mutual information, ABP and HR	-13979.65 (0.03)	-13907.62 (0.06)	-6627.92 (0.03)	-7279.70 (0.05)	31.28	0.521	<0.001
Transfer entropy, HR to T	-10104.68 (0.03)	-10102.14 (0.04)	-5333.33 (0.04)	-4768.81 (0.03)	1.10	0.012	0.298
Transfer entropy, T to HR	-10245.52 (0.02)	-10216.67 (0.04)	-5116.88 (0.03)	-5099.79 (0.02)	12.53	0.030	0.098
Transfer entropy, ABP to T	-10025.14 (0.02)	-10005.87 (0.03)	-5008.92 (0.02)	-4996.96 (0.02)	8.37	0.003	0.592
Transfer entropy, T to ABP	-10238.43 (0.03)	-10232.86 (0.03)	-5365.08 (0.03)	-4867.77 (0.04)	2.42	0.105	0.003
Transfer entropy, ABP to HR	-16323.07 (0.03)	-16317.10 (0.05)	-7371.00 (0.04)	-8946.10 (0.03)	2.59	0.071	0.002
Transfer entropy, HR to ABP	-16136.40 (0.04)	-16123.60 (0.06)	-7765.66 (0.03)	-8357.94 (0.04)	5.56	0.224	<0.001
C-reactive protein (mg/l)	-3988.38 (0.04)	-3980.23 (0.03)	-1803.79 (0.04)	-2176.44 (0.02)	3.54	0.127	0.007
Leukocytes (10^9/l)	-4144.67 (0.03)	-3714.77 (0.04)	-1047.09 (0.03)	-2667.68 (0.03)	186.70	0.096	0.002

Table 3.4: Bayes factors and frequentist p-values for model comparisons of all information trajectories and inflammatory marker trajectories. This involved three-level models, with second-level patient groups and third-level hospital groups (compared to two-level models in Table 3.4). The Bayes factors were calculated using log model evidence estimates from sequential Monte Carlo (SMC) with integrated likelihoods (Equations 3.10, 3.11, 3.14, 3.16). These are colour-coded according to the interpretation table of Jeffrey [128], including substantial in favour of $\mathcal{M}_b$ ($\log_{10} \mathrm{BF}_{ab}$ between -1 and -0.5), strong in favour of $\mathcal{M}_a$ (between 1 and 1.5) and decisive in favour of $\mathcal{M}_a$ (greater than 2). The p-values are from the GAMM hypothesis test (Equation 3.6) and are colour-coded with $p > 0.1$, $p < 0.1$ and $p < 0.01$.

is that the choice of cyclic GAMM splines enforced matching boundary conditions, i.e. $f(0) = f(t_{\max})$, which seemed to result in superficially suboptimal model fits (e.g. increases in mutual information towards the end of the time interval). Finally, the trajectories for the lab measurement inflammatory markers (C-reactive protein and leukocytes) are shown in Figure 3.6, for both paradigms together. I discuss quantitative results from these figures one by one in the following paragraphs, before providing interpretation in Section 3.3.5.

Figure 3.2 shows entropy and mutual information trajectories. As noted in the previous chapter (Figure 2.11), the trajectory models illustrated several cohort-wide patterns, including: (i) HR and ABP entropy gradually increased over time, (ii) temperature entropy was much lower than HR and ABP entropy and appeared to remain largely constant, (iii) mutual information decreased significantly over time, (iv) transfer entropies had small fluctuations but overall decreased a small amount, (v) transfer entropies from ABP to T and from HR to T were smaller than other transfer entropies and did not vary as much. In addition, there were several new insights from the comparison between sepsis cohort and Covid-19 cohort. Figure 3.2 clearly shows that the mutual information trajectories were lower in the sepsis cohort than in the Covid-19 cohort, but Tables 3.3 and Tables 3.4 suggested that there was sufficient between-hospital variability that this could be explained as a hospital-level effect. In the combined cohort, the HDIs were very wide, indicating that the deviance intercept terms (at both patient and hospital level) had large variance in this model. In contrast, the HDIs in the combined cohort were surprisingly narrow in the GAMM framework (Figure 3.3), perhaps suggesting overconfidence in the frequentist model fits (or that the variability was mostly contained in the independent measurement-level error terms ϵ_{ijk}). In the case of mutual information between ABP and T, the differences between cohorts could not be explained as a between-hospital deviation, and model evidence estimation suggested that both cohorts were best modelled separately. While the Bayes factor favoured the separate-cohort alternate model, the frequentist GAMM p-value was not significant at $\alpha = 0.05$ for transfer entropy from ABP to T (though this was reversed in the two-level model without a hospital-level deviation term).

Finally, I also modelled the trajectories of C-reactive protein and leukocytes, both inflammatory markers (Figure 3.6). In both cases, the CRP trajectory decreased over time, though with a sharp peak in the sepsis cohort within the first 48hrs. The white blood cell count (leukocytes) was much lower in Covid-19 patients than in sepsis patients and the Bayes factor favoured the separate-cohort alternate model, suggesting that patients with Covid-19 perhaps had a decreased inflammatory response to infection. However, the HDI for the leukocytes trajectory was wider in the sepsis cohort, suggesting much more between-patient variability. The frequentist p-value was largely in agreement, except for the two-level CRP model. However, the CRP model p-value was significant when the model included third-level hospital groups. This may have been due to model misspecification

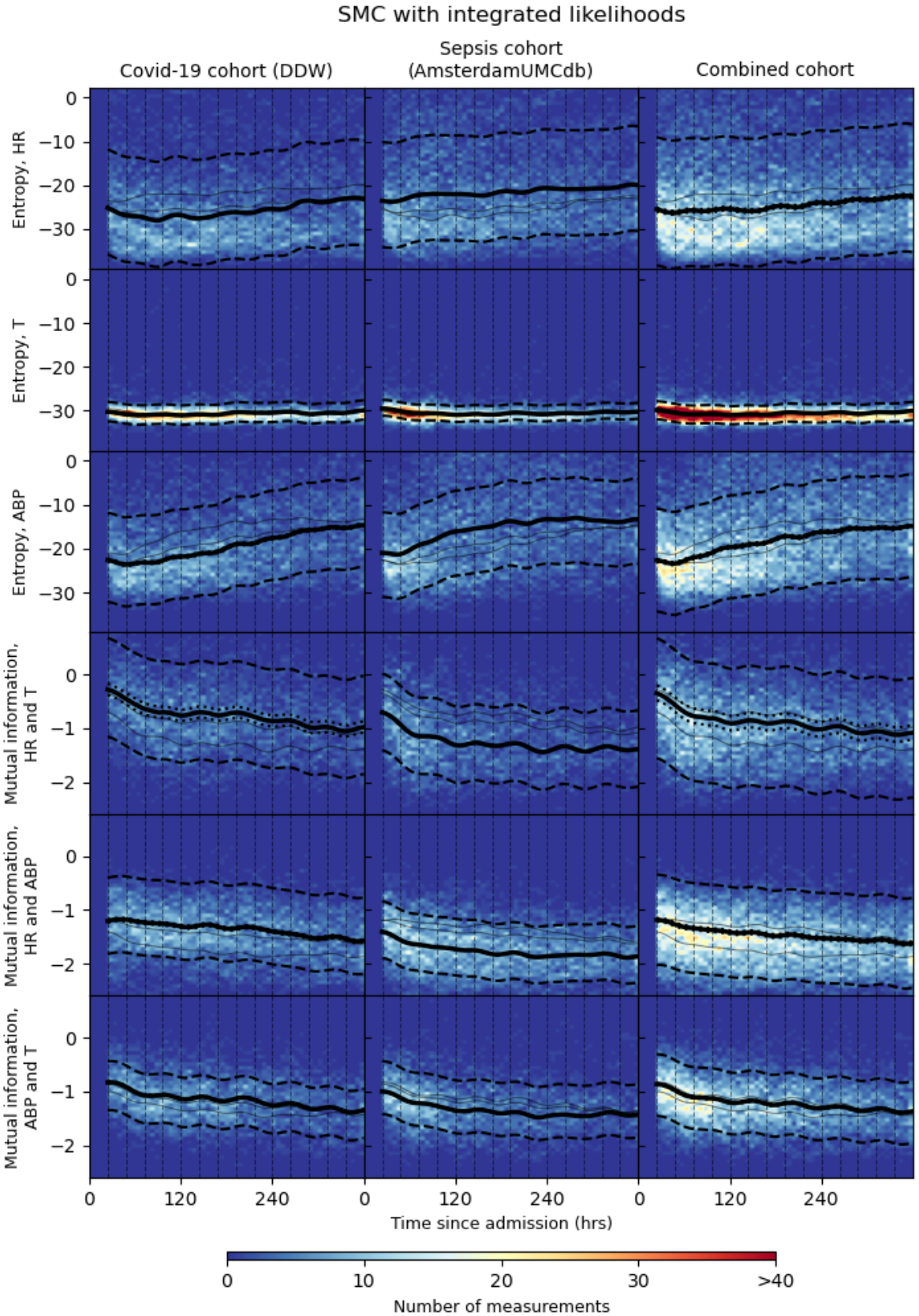

Figure 3.2: Bayesian model fits for information trajectories of entropy and mutual information estimates, using SMC with integrated likelihoods. The heatmaps show counts of each variable across all patients in the cohort, with 50 histogram bins. The (three-level) model fits (−) are the posterior mean of the function $f(t)$, adjusted by the sum of group-level deviations. The dotted lines (··) are the highest density interval (HDI) of $f(t)$ only, and dashed lines (--) are the HDI of $f(t)$ and group-level deviations combined.

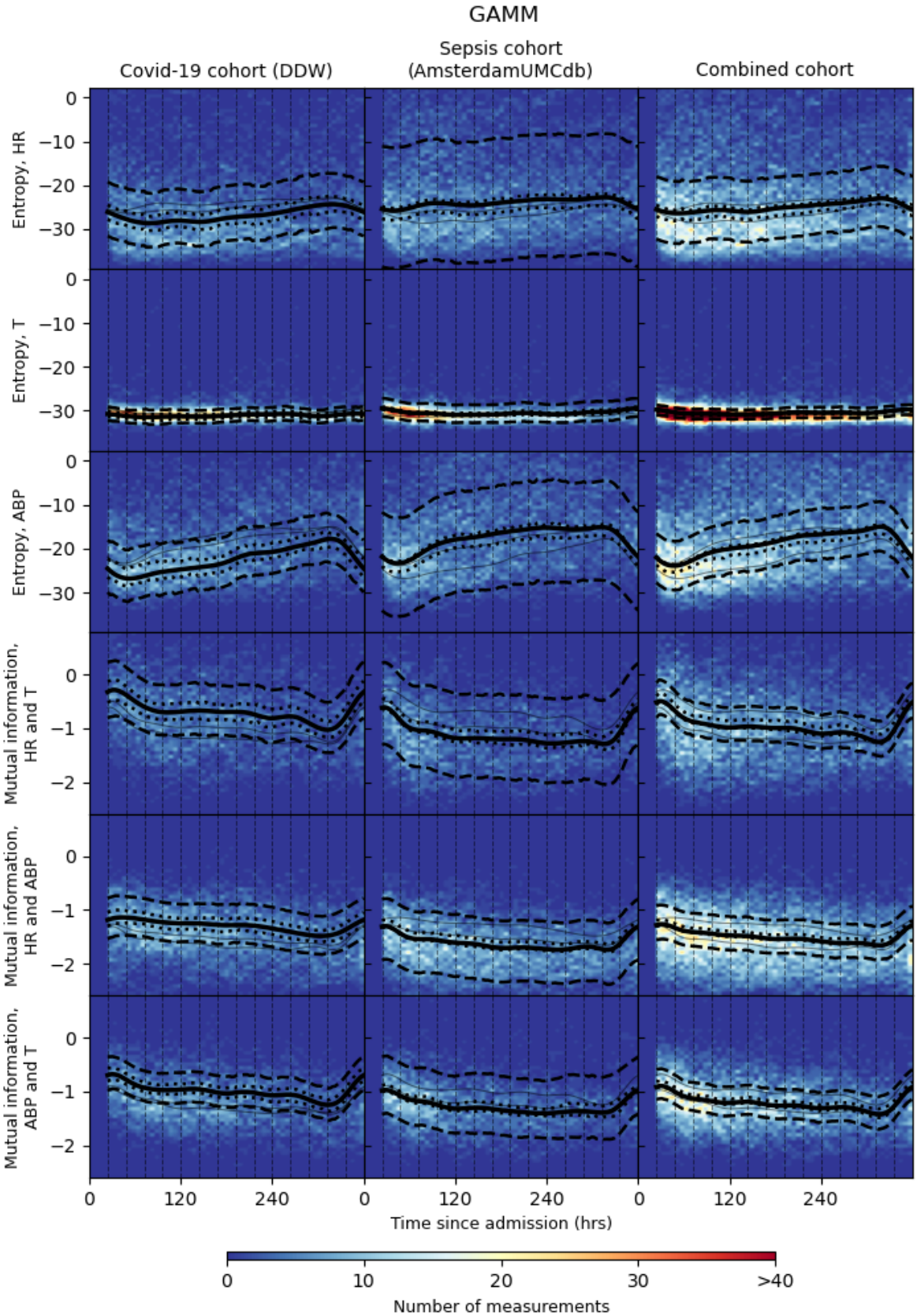

Figure 3.3: Frequentist model fits for information trajectories of entropy and mutual information estimates, using GAMMs with cyclic splines. The heatmaps show counts of each variable across all patients in the cohort, with 50 histogram bins. The (three-level) model fits (−) are the maximum likelihood estimate of the function $f(t)$. The dotted lines (··) are the highest density interval (HDI) of $f(t)$ only, and the dashed lines (--) are the HDI of $f(t)$ and group-level deviations combined.

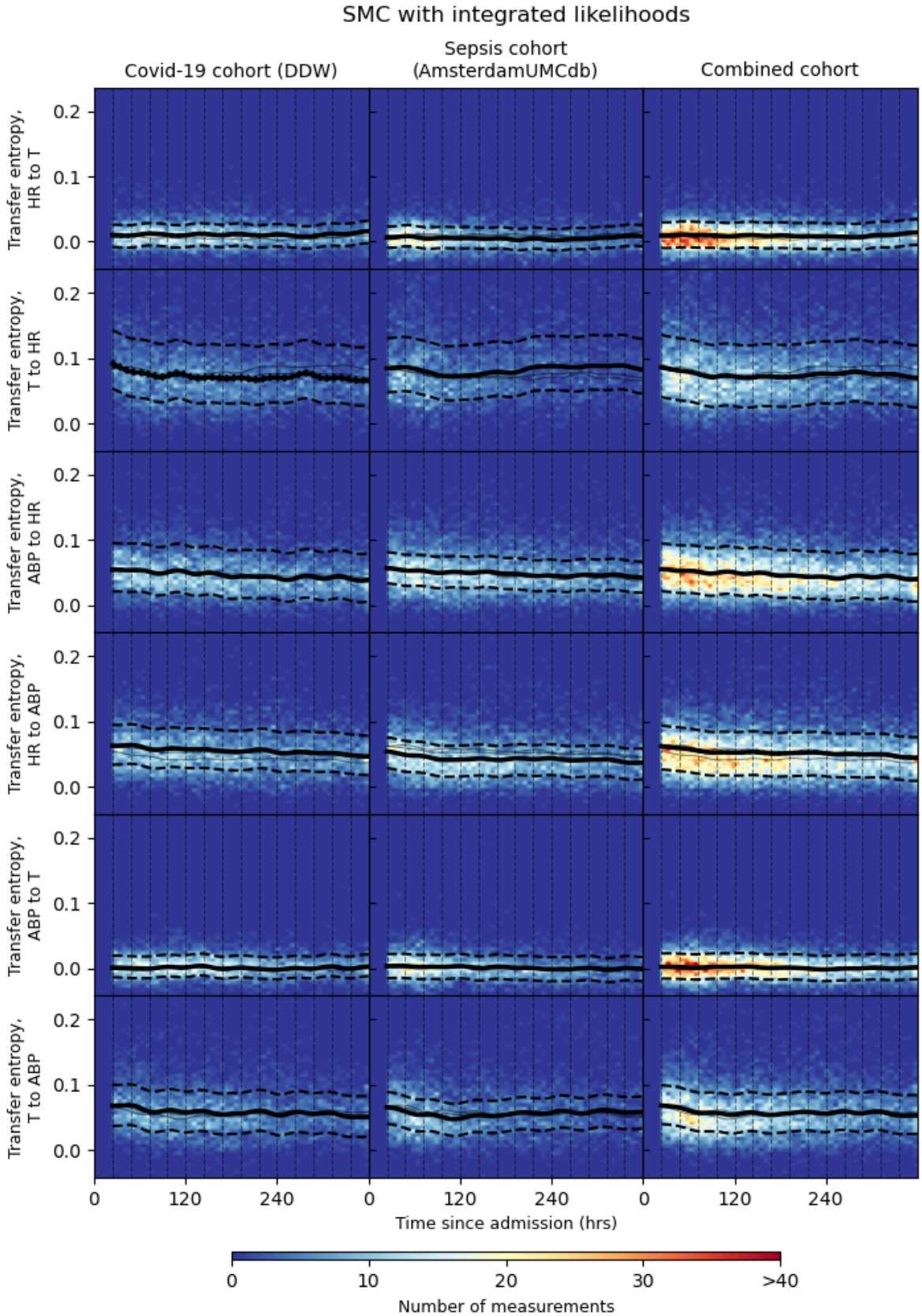

Figure 3.4: Bayesian model fits for information trajectories of transfer entropy estimates, using SMC with integrated likelihoods. The heatmaps show counts of each variable across all patients in the cohort, with 50 histogram bins. The (three-level) model fits (−) are the posterior mean of the function $f(t)$, adjusted by the sum of group-level deviations. The dotted lines (··) are the highest density interval (HDI) of $f(t)$ only, and dashed lines (--) are the HDI of $f(t)$ and group-level deviations combined.

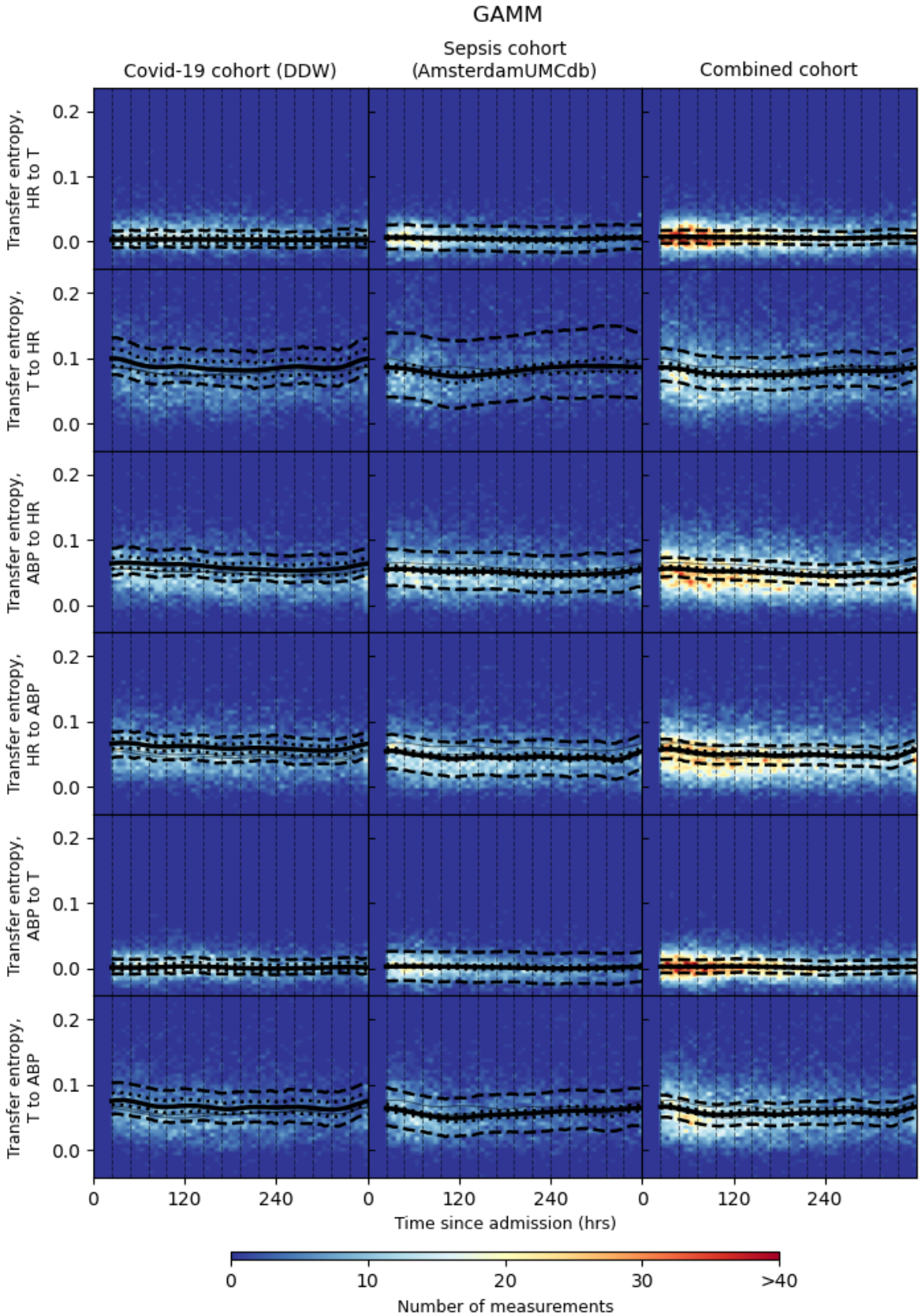

Figure 3.5: Frequentist model fits for information trajectories of transfer entropy estimates, using GAMMs with cyclic splines. The heatmaps show counts of each variable across all patients in the cohort, with 50 histogram bins. The (three-level) model fits (−) are the maximum likelihood estimate of the function $f(t)$. The dotted lines (··) are the highest density interval (HDI) of $f(t)$ only, and the dashed lines (--) are the HDI of $f(t)$ and group-level deviations combined.

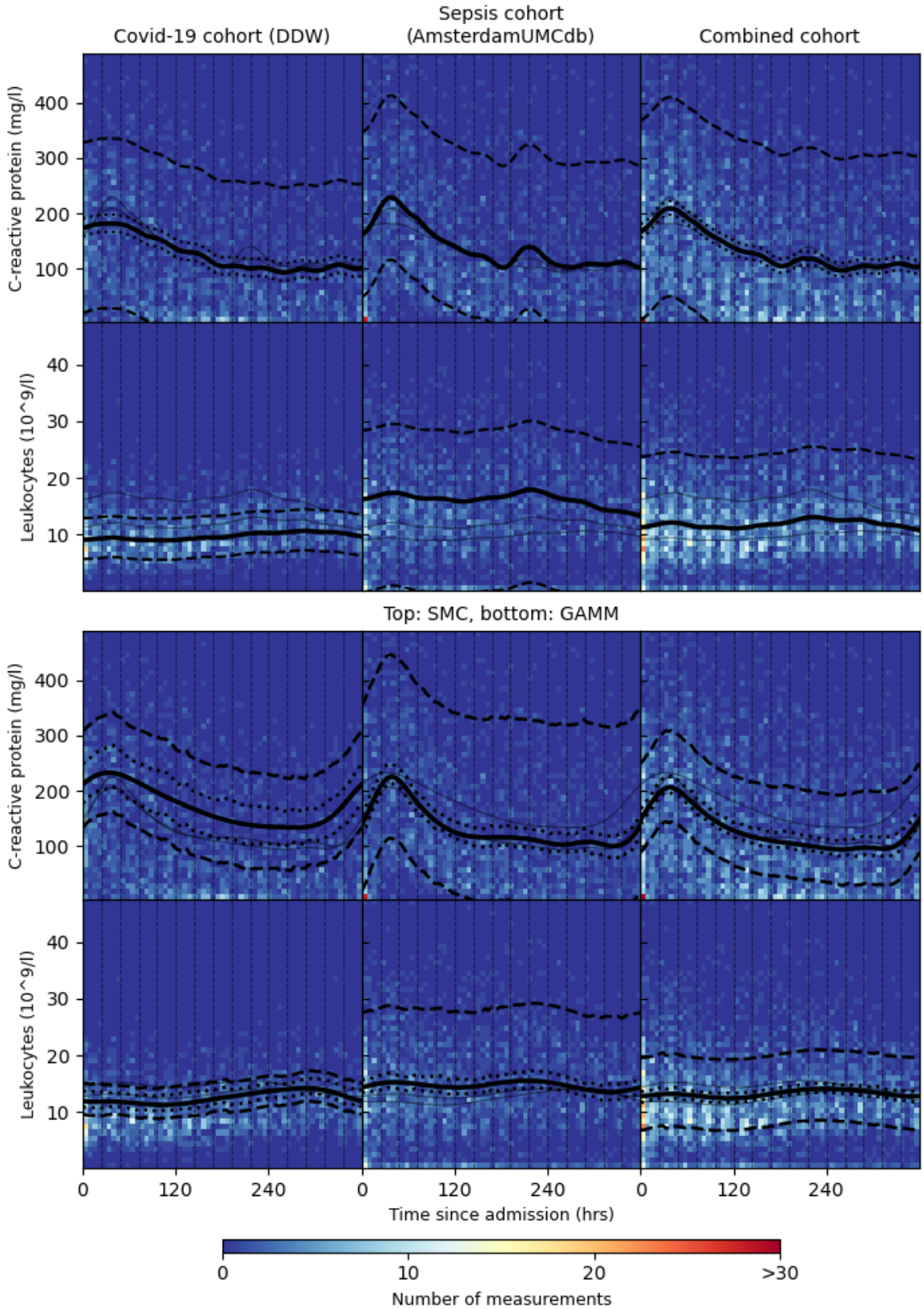

Figure 3.6: Both Bayesian and frequentist model fits for inflammatory markers, C-reactive protein and leukocytes. The former (top rows) are from SMC using integrated likelihoods and the latter (bottom rows) are from GAMM using cyclic splines. The heatmaps show counts of each variable across all patients, with 50 histogram bins. The (three-level) model fits (−) are the posterior mean or MLE of the function $f(t)$, adjusted by the sum of group-level deviations. The dotted lines (··) are the highest density interval (HDI) of $f(t)$ only, and the dashed lines (--) are the the HDI of $f(t)$ and group-level deviations combined.

or possibly due to the hospital groups 'explaining away' some of the unknown error-term within the two-level model, but it was not clear what this meant in practice.

3.3.3 ICU mortality and endpoint alignment

The rationale behind choosing a model with each patient following the same trajectory (up to individual intercept terms) came from anecdotal observations from intensive care. Clinical colleagues noted that patients admitted to ICU often had high levels of inflammation during the first 24 hours, which then decreased over time. This observation was reflected in the C-reactive protein measurements (Figure 3.6). As a result, we assumed that patients were admitted to intensive care at similar points in their disease progression. However, the previous analysis showed there was clearly significant variability within the population-wide trends.

One issue with this assumption is that there may be subgroups within the population who responded differently over time. These subgroups may have experienced very different disease progressions, which would reflected in the trajectories of information-theoretic measures. Patient outcomes were recorded in both databases, and it is reasonable to assume that there are differences in cohort-averaged trajectories between patients who died in ICU and patients who were discharged. In normal circumstances, patients need to have shown some clinical improvement, or at least stability, in order to be discharged from ICU (excluding those discharged for palliative care, which I accounted for in our definition of ICU death if they died within 24 hours of discharge). In this instance, the scarcity of ICU beds during the Covid-19 pandemic may have meant that patients were more readily discharged to another ward, but discharged patients would still have had better clinical prognosis than those who had died or were not yet discharged.

With these considerations in mind, I decided to stratify by ICU mortality, and align the data by the endpoint rather than by the time of admission. This latter change was important, as ICU length of stay was very different across the entire cohort. In any case, it is possible that the time of departure from ICU (discharge or death) was more meaningful than the time of admission to ICU, in terms of disease progression. As the original dataset included patients who were still in ICU after 15 days, I used a reduced dataset in order to align by ICU departure (Table 3.5), and excluded the patients with length of stay greater than 15 days. This analysis involved 9 cohorts. Four of these were stratified by ICU mortality and by disease (Covid-19 vs sepsis), and the remaining 5 were combined datasets of one or both stratified cohorts. As before, I used the integrated likelihood method with SMC to model the trajectories for each cohort. I also performed various model comparisons, comparing a baseline model, in which two cohorts shared model parameters, against an alternate model, in which they had separate model parameters.

Figures 3.7 and 3.8 show the Bayesian model fits for the stratified cohorts. For

	Cohorts								
	Covid-19			Sepsis			Covid-19 and sepsis		
	Death	Discharge	All	Death	Discharge	All	Death	Discharge	All
No. of patients, n	17	57	74	39	72	111	56	129	185
No. of hospitals, K	3	3	4	1	1	1	4	4	5
Number of measurements, m									
Entropy, T	412	1194	1606	601	1089	1750	1073	2283	3356
Entropy, HR	563	1705	2268	1078	1928	3006	1651	3633	5274
Entropy, ABP	563	1680	2243	1074	1871	2945	1637	3551	5188
MI, T and HR	412	1192	1604	661	1086	1747	1073	2278	3351
MI, T and ABP	412	1188	1600	659	1080	1739	1071	2268	3339
MI, HR and ABP	562	1680	2242	1071	1857	2928	1633	3537	5170

Table 3.5: Dataset sizes for information-theoretic measures, centred at endpoints. These datasets included only those patients with ICU length of stay less than 15 days. For each pair of physiological time-series, the number of transfer entropy measurements (in each direction) was the same as the number of mutual information measurements. For information-theoretic measures, there was a maximum of 57 measurements per patient.

patients who were discharged (Covid-19 and sepsis), the entropy values increased sharply prior to ICU departure, while mutual information and transfer entropy values decreased by varying amounts. The decreases in MI and TE were somewhat surprising, since it suggests that different physiological subsystems were interacting less strongly despite these patients presumably showing clinical improvement. It is possible that entropy within physiological signals separately is a more important indicator of health, in terms of the 'decomplexification' of allostasis, than their interaction. Another possible explanation was that the decreasing mutual information and transfer entropy between variables values arose as an artefact of increasing variability within each variable. However, there was only very weak negative correlation between the information-theoretic measures (Figure B.2). In any case, there were clear differences between ICU cohorts, when stratified by ICU mortality. At the population-level, patients who were discharged experienced more pronounced changes in all information-theoretic trajectories prior to ICU departure, than patients who died in ICU. From these figures, it appeared that differences between ICU death and discharge cohorts (for both Covid-19 and sepsis cohorts) were greater than differences between the disease cohorts (for both ICU death and discharge). This was supported by estimation of Bayes factors in Table 3.6. For example, the top left cell in this table shows the Bayes factor for a baseline model that treated all Covid-19 patients as one cohort against an alternate model that treated Covid-19 patients who died as a separate cohort from Covid-19 patients who were discharged. Similarly, the bottom right cell shows the Bayes factor for a baseline model that treated all patients as one cohort against an alternate model that treated Covid-19 patients as a separate cohort from sepsis patients

Variable	ICU death vs discharged			Covid-19 vs Sepsis		
	Covid-19	Sepsis	All	ICU death	Discharged	All
Entropy, T	29.80	10.31	31.07	22.15	18.96	32.07
Entropy, HR	20.95	41.39	58.18	18.67	12.66	27.18
Entropy, ABP	47.28	35.89	80.90	0.72	13.60	12.05
MI, HR and T	11.52	10.06	21.06	0.18	-1.59	-1.93
MI, ABP and T	11.44	21.59	33.41	-1.50	-1.23	-2.35
MI, ABP and HR	34.35	31.69	64.12	0.03	-1.06	-2.94
TE, HR to T	0.28	1.22	-0.13	-1.17	3.71	0.90
TE, T to HR	0.86	1.65	-1.24	4.52	-0.27	0.51
TE, ABP to T	-0.31	-0.05	-1.94	1.64	1.07	1.14
TE, T to ABP	-2.35	2.54	1.50	0.52	-1.83	0.01
TE, ABP to HR	6.16	-0.73	3.56	-0.26	2.14	0.01
TE, HR to ABP	7.64	-2.55	5.37	5.04	7.41	12.73

Table 3.6: Bayes factors for model comparisons of all information trajectories, centred at the endpoint. This involved three-level models (Figures 3.7 and 3.8), with second-level patient groups and third-level hospital groups. The Bayes factors were calculated using log model evidence estimates from sequential Monte Carlo (SMC) with integrated likelihoods (Equations 3.10, 3.11, 3.14, 3.16). These are colour-coded according to the interpretation table of Jeffrey [128], including decisive / (very) strong / substantial in favour of the shared model and substantial / (very) strong / decisive in favour of split model.

(regardless of ICU mortality). Notably, there was evidence to support the observation that there were differences in entropy and mutual information trajectories between patients who died in ICU and patients who were discharged. There was also evidence of differences in the entropy trajectories between Covid-19 and sepsis cohorts. Apart from this, the general picture was more mixed.

3.3.4 Misspecification and model checking

The non-linear regression models used in this chapter made strong assumptions about structure of the information trajectories. In particular, it was assumed that individual patients follow shared trajectories up to patient-level (and hospital-level) intercept terms. Additionally, errors were assumed to be normally-distributed. Without a priori knowledge, I chose weakly-informative priors for the model parameters, including for the variance terms associated with multilevel effects and errors. These priors were chosen to ensure that the prior predictive distributions covered the data fully. However, if the assumptions within the non-linear regression models were incorrect and the model was misspecified, this could have a significant impact on model comparison.

The model comparisons in this chapter were to determine whether two cohorts share the same model parameters (and therefore the same trajectory shape, multilevel effects and

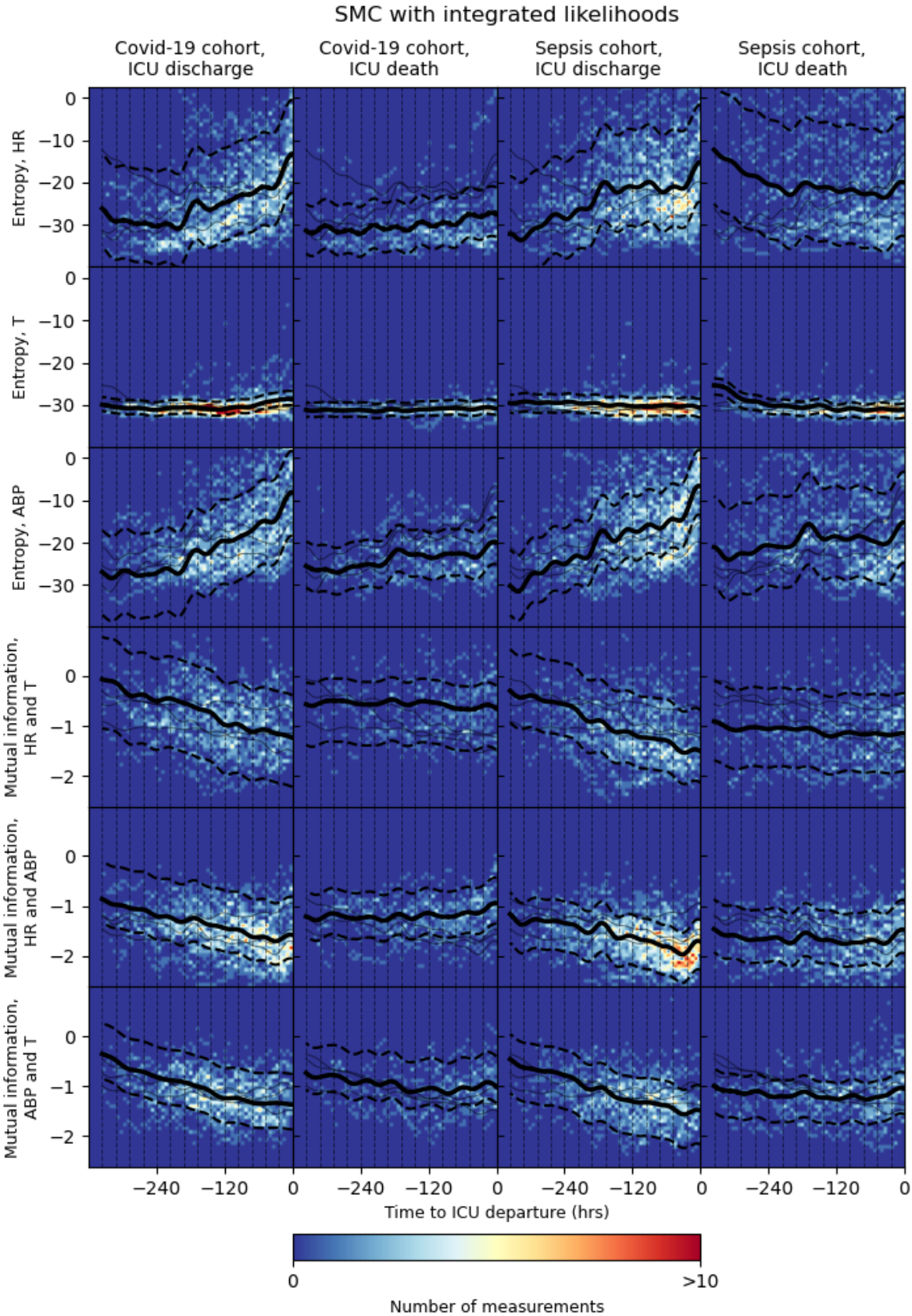

Figure 3.7: Bayesian model fits for information trajectories of entropy and mutual information estimates, using SMC with integrated likelihoods. The heatmaps show counts of each variable across all patients in the cohort, with 50 histogram bins. The (three-level) model fits (−) are the posterior mean of the function $f(t)$, adjusted by the sum of group-level deviations. The dotted lines (··) are the highest density interval (HDI) of $f(t)$ only, and dashed lines (--) are the HDI of $f(t)$ and group-level deviations combined.

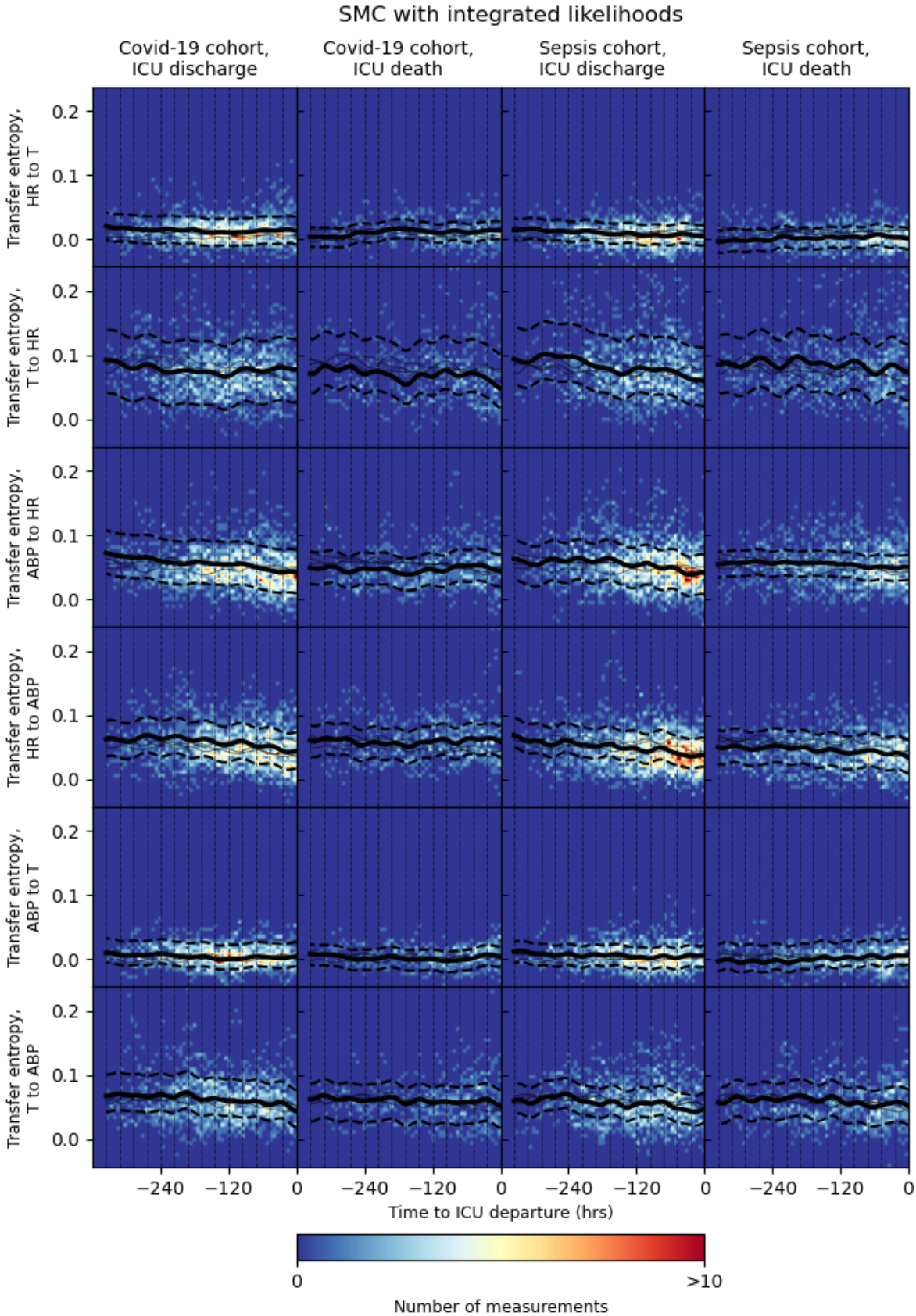

Figure 3.8: Bayesian model fits for information trajectories of transfer entropy estimates, using GAMMs with cyclic splines. The heatmaps show counts of each variable across all patients in the cohort, with 50 histogram bins. The (three-level) model fits (−) are the maximum likelihood estimate of the function $f(t)$. The dotted lines (··) are the highest density interval (HDI) of $f(t)$ only, and the dashed lines (--) are the HDI of $f(t)$ and group-level deviations combined.

error variance). The baseline model that both cohorts share parameters, $\mathcal{M}_a$, was a reduced version of the alternate model $\mathcal{M}_b$, in which both cohorts independently had different model parameters. In model comparison using Bayes factors, model misspecification can results in bias towards the more complex, flexible model (i.e. $\mathcal{M}_b$). This is true if the simpler model is misspecified or if both models are misspecified. This may have had some consequences on the results in Tables 3.3, 3.4 and 3.6. For example, in the two-level models without hospital-level effects, the alternate model was favoured in every instance. In the three-level models with hospital-level effects, the alternate model was generally favoured less strongly and the baseline model was favoured in one instance (transfer entropy from HR to T). In this particular case, the model evidence for the three-level models was much less than for the corresponding two-level models. One explanation for this is that the two-level models were misspecified, so, without hospital-level effects, the alternate model was incorrectly favoured. However, it is worth noting that, although the simpler three-level baseline model was favoured ahead of the three-level alternate model here, this does not necessarily mean that this baseline model was not also misspecified.

There are several approaches for checking how well a model is specified in Bayesian and frequentist settings. Goodness-of-fit tests can include test statistics that are shown to converge in distribution to a known distribution when the model is correct. The model fit is then evaluated by comparing the value of the test statistic against this known distribution. For regression models with multilevel effects, one example of this type of goodness-of-fit test is a χ^2 test [142]. Another option is to perform posterior predictive checks. These seek to quantify discrepancies between the data and the fitted model, and whether these discrepancies could have occurred randomly given the model assumptions [143]. Posterior predictive checks involve sampling multiple replicated data, denoted y^{rep}, from the posterior predictive distribution. These replicated data have the same covariate data and implicit multilevel structure as the observed data, but their response variables are computed directly from the model via the posteriors of the model parameters. This means that the replicated data is data that could have been observed, if the model was true. The posterior predictive distribution for the three-level model $\mathcal{M}_{31}$ is discussed in Appendix B.2.

The most common types of posterior predictive checks include visual checks of residuals and comparison of a test statistic $T(y)$ or test quantity $T(y, \theta)$, between the observed data and the replicated data. Assuming the model is true, generating multiple instances of replicated data allows an estimate of the distribution of a test statistic or test quantity. It is then possible to estimate the posterior predictive tail probabilities, i.e. $p(T(y^{\text{rep}}) \geq T(y)|y)$ or $p(T(y^{\text{rep}}, \theta) \geq T(y, \theta)|y)$. In order to check the models described in this chapter, I decided to perform posterior predictive checks using the within-patient residual sum of squares (RSS) and the between-patient RSS. For a three-level model, fitted with the

posterior means for β, η and ζ, these were defined as the following:

$$\text{RSS}_w = \sum_{ijk}(y_{ijk} - \hat{\beta}^T x_{ijk} - \hat{\eta}_{jk} - \hat{\zeta}_k)^2, \ \text{RSS}_b = \sum_{jk}(\sum_{i}(y_{ijk} - \hat{\beta}^T x_{ijk} - \hat{\eta}_{jk} - \hat{\zeta}_k))^2$$

Additionally, to test the model assumption that there were group-level intercept terms but not group-level gradient terms, I also decided to estimate the intercepts and slopes of each patient, using simple linear regression, with test statistics based on the spread of each variable. As the number of observations for each patient was quite small in some cases, I decided to use the interquartile range (IQR) for this purpose, rather than the standard deviation, because it is more robust. In theory, the intercept IQR should be accounted for within the model by the patient-level effects.

I evaluated these test statistics for 1000 instances of replicated data. Figures B.1 and B.3 in Appendix B.2 show some examples of the replicated data for HR entropy, and of the test statistics under the posterior predictive distribution. Figure 3.9 shows the proportion of times that the test statistics were lower for replicated data than for the observed data, across all 144 models (12 datasets and 12 variables). In almost every case, the within-patient RSS for the observed data were much higher than for the replicated data. Only for 16 out of the 144 models was this proportion less than 0.99. For the between-patient RSS, there was a marked difference between the sepsis cohorts and the Covid-19/combined cohorts. The between-patient RSS was generally much higher for the observed data than for the replicated data for the Covid-19 cohort, but the opposite was true for sepsis cohort. The sepsis cohort came from just one hospital, so was modelled without hospital-level effects. It appears that the inclusion of hospital-level effects caused the model to underestimate between-patient variability and the exclusion of hospital-level effects caused the model to overestimate between-patient variability. Similarly, the test statistic based on patient-specific gradients was greater for the observed data than for any replicated data, in almost every case. Averaged across all models, the mean proportion was 0.996. This suggested the observed data was implausible under the assumption that the group-level effect was intercept-only. Adding a patient-level gradient term to the model may reduce this discrepancy, but this was not tested further.

Overall, these results suggest that the models were indeed misspecified and a different model structure or assumptions may have been better suited for this analysis. Despite these model failings, it should be noted that misspecified model can still be useful for providing insight. Though the model clearly masks some patient-specific effects, part of the motivation for these models was to describe population-wide trends, which arguably these models have achieved. Additionally, while model misspecification can impact the model comparison using Bayes factors, many of the Bayes factors in this analysis had extremely large absolute values, and favoured the separate-cohort alternate model with

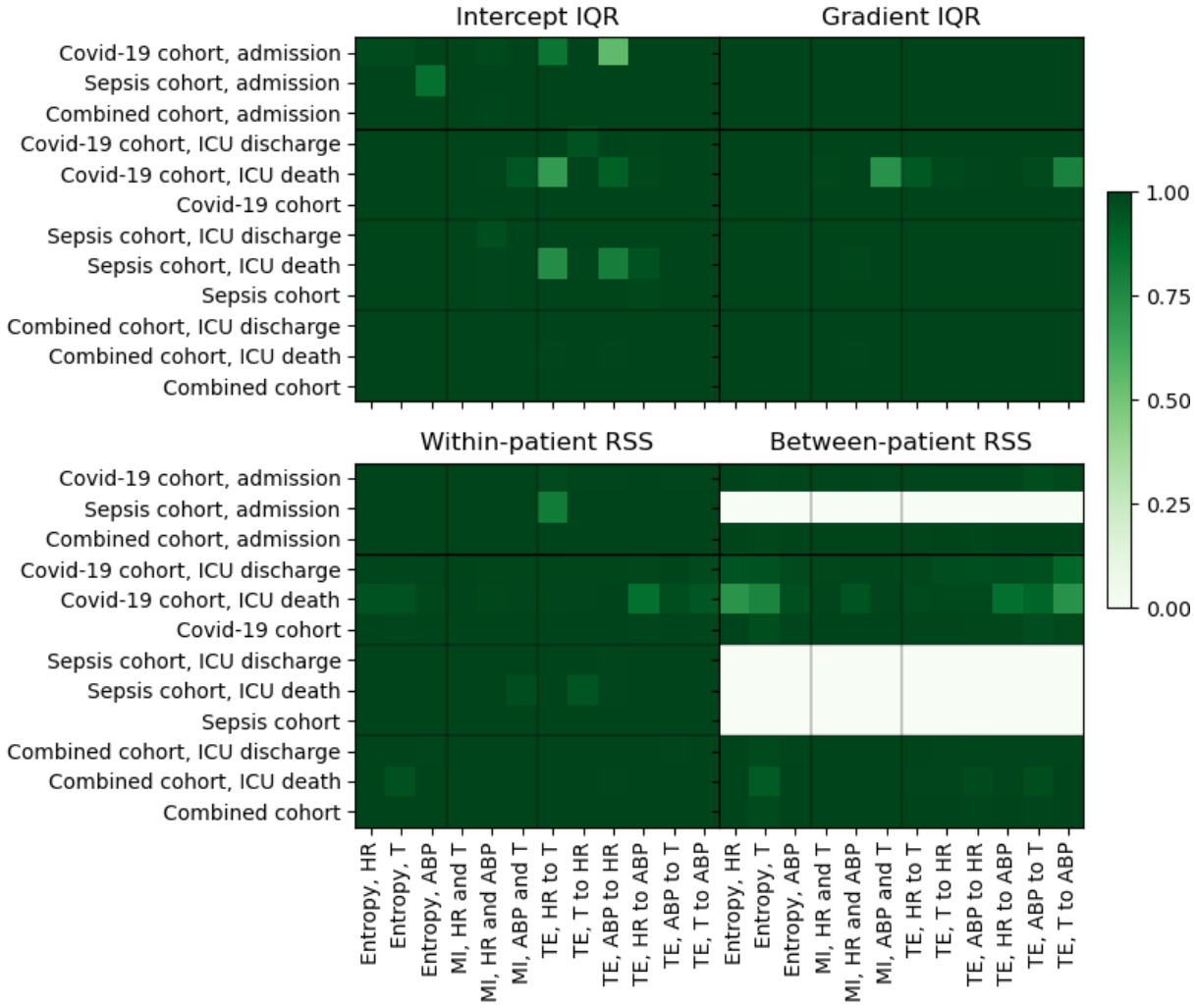

Figure 3.9: Posterior predictive checks for all models. Four test statistics were defined for model checking: IQRs of patient-specific intercepts and gradients, within-patient RSS and between-patient RSS. The test statistics were evaluated for the observed data and for $S = 1000$ instances of replicated data. For each test statistic and each model, the figure shows the value of $\frac{1}{S}\sum_s (T(y^{\text{rep},s}) < T(y))$. Values close to 0 or 1 indicate that the test statistic evaluated for observed data was consistently outside the posterior predictive distribution for this test statistic.

and without the addition of hospital-level effects. While these results should be interpreted with caution because of model misspecification, it is likely that there were in fact significant differences between the cohorts.

3.3.5 Revisiting the clinical hypothesis

As a brief reminder of the concepts involved: entropy is a measure of 'expected surprise' in variable X, mutual information is the reduction in the 'expected surprise' for variable X that is gained when the value of the variable Y is known (or vice versa), and transfer entropy is the average information that past values of variable X provide about the current value of variable Y, above and beyond the past values of variable X (transfer entropy is a necessary but not sufficient condition for a causal relationship, when separability is assumed). Using these information-theoretic concepts applied to three physiological time-series (all of which are regulated by the brainstem), I compared a cohort of patients with severe illness caused by respiratory virus (Covid-19 cohort) and a cohort of patients with any respiratory infection (sepsis cohort). Generally, there were significant differences between the cohorts for most of information-theoretic measures, which I had evaluated on individual time-series or pairs of physiological time-series. In particular, mutual information (and some transfer entropies) tended to decrease over time within both cohorts, while entropy values tended to increase. This suggested that there was some physiological dysregulation in both cohorts. For instance, an increase in entropy ('expected surprise') in HR and in ABP suggests that cardiovascular allostasis ('stability in variation' of cardiovascular function) was gradually lost during ICU stay. Decreases in mutual information suggested a reduced interaction between cardiovascular function and thermoregulation during ICU stay, which could be also be interpreted as a symptom of brainstem dysfunction.

Ultimately, these results are challenging to interpret objectively, for several reasons. Firstly, these relationships are clearly complicated, but the trends were broadly similar between sepsis and Covid-19 cohorts (though differences between the cohorts were generally statistically significant). Secondly, there is no objective, meaningful interpretation of the absolute value (or change in value) of the information-theoretic indices. This is mentioned further in Chapter 6. In the absence of this, any conclusions are somewhat speculative. However, there was some evidence that physiological relationships were affected in patients with severe respiratory infection and Covid-19, but more work is required in order to contextualise what this means and what the true underlying causes of this are. For example, there is currently no equivalent data for the information trajectories of physiological systems in healthy humans. A comparison between mild and severe Covid-19 may have been more illuminating, had this data been available, because this may have elucidated the severity of symptoms. Furthermore, a breakdown in the interaction between physiological systems may be related to brainstem dysfunction, but there are potential confounding factors that

deserve brief mention. For example, it may be the case that bed rest and limited patient movement contributed to decreased physiological interactions, more than the underlying physiology of Covid-19 of respiratory viral infection. However, accumulation of evidence from different sources, e.g. [24, 25], should help to consolidate the evidence and findings from each individual source. I believe this data-driven approach has the potential to add more weight to the Covid-19 brainstem dysfunction hypothesis, and to help further illuminate the mechanisms involved in, and the consequences of, brainstem dysfunction induced by respiratory viral infection.

CHAPTER 4

Artefact detection in time-series data using generative deep learning

This chapter presents work from the first year of my PhD, on artefact detection in physiological time-series. At this time, I was primarily interested in representation learning of ICU time-series, and whether novel clinical insights about the patient state could be gained by visualising temporal trajectories in the latent representation space from a generative deep learning model. This is a difficult task, because generative deep learning models are usually 'black box' functions and latent representations are not easily human-interpretable. However, a latent representation space is also useful beyond the representation learning itself. If the model learns to encode meaningful representations only for 'valid' data, then combining the generative ability of the model with careful preprocessing and post-processing can help to discriminate between 'valid' and 'invalid' segments of physiological time-series. Unless properly handled, artefacts in data can bias clinical summary variables, cause unnecessary 'false alarms' in automatic warning systems, and confound subsequent downstream research tasks. Using a variational autoencoder model, I built an artefact detection algorithm to identify, remove and impute time-series data that contained artefacts, without the model itself being shown any types of artefactual data. I presented this in an oral presentation at the International Symposium on Intracranial Pressure and Neuromonitoring (ICP2019) in Leuven, with an accompanying conference proceedings paper published in *Acta Neurochirurgica* [23]. I also presented this work as a poster and elevator pitch oral presentation at the CCIMI workshop session 'Geometric and Topological Approaches to Data Analysis', in a poster session at STEM for Britain 2021, and as an invited oral presentation in the ESICM Lives 2021 virtual conference.

4.1 Introduction

Continuous monitoring of physiological waveform (time-series) data is a crucial component in patient care for critically ill patients in ICU, in particular providing real-time alerting of rapid changes in the patient state and allowing estimation of a wider set of useful clinical parameters. Waveform recordings are susceptible to data artefacts, which must be removed before the data can be used for automatic alerting or repurposed for other clinical or research purposes. Waveform artefacts arise from a variety of internal and external sources, including sensor noise, patient movement and clinical interventions. Examples of the latter include arterial flushing [144], when the arterial line is transduced to re-establish a relative pressure baseline, and draining of intracranial fluid. The former is repeated at infrequent intervals either to decrease damping caused by blood clotting or upon moving the patient, while the latter alleviates pressure on the brain caused by swelling after traumatic brain injury. Waveform artefacts may reduce reliability in downstream estimation of derived parameters and can be a distraction in clinical analysis and decision making. Artefacts also contribute to a high false positive rate of automatic ICU alarms, and alarm fatigue can leave clinical staff with a perception that the alarm systems are generally unhelpful [145]. As a result, this creates a potential risk that a missed true positive alarm leads to a delay in the appropriate clinical intervention [146]. Accurate artefact identification and removal can therefore contribute to a safer ICU environment, and reduce both bias and uncertainty in clinical assessment.

An artefact identification method will mark time-series segments that contain an artefact as 'invalid'. In essence, the time-series segment is subsequently treated as missing data. However, missing data can have similar effects to artefacts in downstream tasks, particularly as missing data preprocessing is routinely handled in a simplistic manner that introduces bias or underestimates natural variability (e.g. linear interpolation or last measurement carried forward). Imputation methods that maintain some statistical properties or features of the data can mitigate these issues [147], and an ideal artefact detection algorithm should include scope for data imputation in segments where an artefact has been identified.

Artefact detection has traditionally been a difficult and costly task [148], requiring time-consuming manual annotation or (fragile) signal-specific automatic thresholding based on feature engineering [149]. Due to the complex structure of physiological waveform data, automated systems are still inferior to annotation by experienced clinicians, which remains the gold standard in spite of internal biases and issues with replicability. Many supervised deep learning methods require pre-annotated samples for model training, which can be costly to produce. Unsupervised approaches use generative deep learning to construct 'realistic' synthetic observations, and use spectral anomaly detection [150] to separate

artefacts from 'valid' data by their representations in a lower-dimensional latent space. This lower-dimensional 'information bottleneck' is intended to capture salient features of 'valid' data and disregard artefactual features, which then results in synthetic observations that do not contain any artefacts. By comparing the distribution of matching real-synthetic observation pairs, real observations containing artefacts can be discriminated from 'valid' observations or, equivalently, real artefactual observations will have high reconstruction error and low probability under the generative model distribution [151].

4.1.1 Key contributions

In this chapter, I focused on unsupervised artefact detection for time-series data. My contributions to this area of research were:

1. I presented an artefact detection algorithm called DeepClean, which used a variational autoencoder (VAE) with deep convolutional neural networks. As an unsupervised algorithm, this avoids costly and painstaking manual annotation of physiological waveforms, required only easily-obtained 'valid' time-series for model training. The generative deep learning model in this algorithm does not provide binary labels identifying artefacts itself. These come from a post-processing step, in which real-synthetic observation pairs are evaluated using a simple distance-based decision rule.
2. I demonstrated the performance of DeepClean using a test case of high-frequency invasive arterial blood pressure (ABP) measurements. I showed that the DeepClean framework was able to detect the presence of artefacts within 10s long observations, with sensitivity and specificity at about 90%, massively outperforming a baseline 'information bottleneck' approach that used principal component analysis (PCA). Additionally, DeepClean identified within-observation artefactual segments with high accuracy. As a generative deep learning model, DeepClean can produce synthetic observations (conditional on a real observation), which can be used for data imputation when an artefact is identified within a real observation.
3. While this work is a standalone chapter in my thesis, I wanted to highlight how the VAE framework fits in with other concepts I have previously discussed at length, so I reviewed the literature on VAEs to highlight links between this generative deep learning framework, information theory and importance sampling.

4.1.2 Mathematical definition and notation

I denote observations from a real dataset $\mathcal{D}_r$ as x_i^r for $i = 1, \ldots, n$, or as $X_r = (x_1^r, \ldots, x_n^r)$, where n is the number of observations within the dataset. In this chapter, these observations

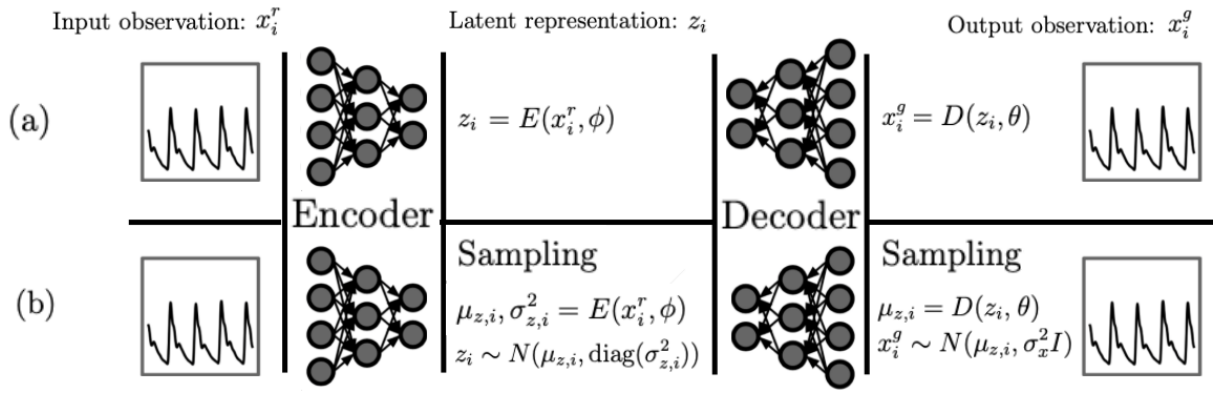

Figure 4.1: (a) An autoencoder and (b) a variational autoencoder. The autoencoder trains an encoder and decoder simultaneously to reconstruct input observations via sparse low-dimensional latent representations. The VAE includes additional sampling steps, which explicitly task the decoder to train a probabilistic generative model. The weights of the encoder neural network are ϕ and the weights of the decoder neural network are θ.

are time-series of length T with constant frequency (i.e. the difference between successive timestamps is fixed). The observation x_i^r can be written as $x_i^r = (x_{i1}^r, \ldots, x_{iT}^r)^T$, and is a realisation of a distribution $\mathbb{P}_x$ on a sample space $\mathcal{X} \subseteq \mathbb{R}^T$. As usual, I denote any distance metric by $d(\cdot, \cdot)$, where this is assumed to be Euclidean unless otherwise specified.

The main generative model in this chapter has an autoencoder-like structure. An autoencoder is made up of a pair of models, an encoder and a decoder, which are usually deep neural networks [152]. These are the functions $E(\cdot)$ and $D(\cdot)$ and are parameterised by (trainable) weights ϕ and θ respectively (Figure 4.1). For (fixed) parameters ϕ, the encoder maps an observation x_i^r either (i) to a 'latent representation' variable z_i in latent space $\mathcal{Z}$ or (ii) to parameters of a known 'variational' distribution $\mathbb{P}_{z,i}$ on latent space $\mathcal{Z}$. The latent space $\mathcal{Z}$ has dimension $d_z \ll T$, so forces an information compression of the input observation. The variational autoencoder (VAE) used in the DeepClean framework in this chapter does the latter mapping, from the input observation to a mean and variance of a multivariate Gaussian distribution $N(\mu_{z,i}, \mathrm{diag}(\sigma_{z,i}^2))$. In this case, the observation x_i^r is represented in the latent space by a latent representation variable z_i sampled from $\mathbb{P}_{z,i}$, e.g. $z_i \sim N(\mu_{z,i}, \mathrm{diag}(\sigma_{z,i}^2))$. In mathematical terms, the encoder variants are:

(i) $E: \ \mathcal{X} \times \Phi \to \mathcal{Z}, \ (x_i^r, \phi) \mapsto z_i = E(x_i^r, \phi)$

(ii) $E: \ \mathcal{X} \times \Phi \to \mathcal{Z} \times [0, \infty)^{d_z}, \ (x_i^r, \phi) \mapsto (\mu_{z,i}, \sigma_{z,i}^2) = (E_\mu(x_i^r, \phi), E_{\sigma^2}(x_i^r, \phi)) = E(x_i^r, \phi)$

For (fixed) parameters θ, the decoder maps a latent representation z_i either (iii) directly to a synthetic observation x_i^g or (iv) to parameters of a known distribution $\mathbb{P}_{x,i}$. For example, in the DeepClean VAE, the decoder $D(\cdot, \theta)$ maps z_i to a vector mean $\mu_{x,i}^g = D_\mu(z_i, \theta) = D(z_i, \theta)$, for a multivariate Gaussian with fixed variance σ_x^2. In this case, the synthetic observation x_i^g is then sampled from this distribution, e.g. $x_i^g \sim N(\mu_{x,i}^g, \sigma_x^2 I)$. These are

formulated as:

$$\text{(iii) } D: \ \mathcal{Z} \times \Theta \to \mathcal{X}, \ (z_i, \theta) \mapsto x_i^g = D(z_i, \theta)$$
$$\text{(iv) } D: \ \mathcal{Z} \times \Theta \to \mathcal{X}, \ (z_i, \theta) \mapsto \mu_{x,i}^g = D_\mu(z_i, \theta)$$

As shown in Figure 4.1, an autoencoder [152] has an encoder and decoder of the form (i) and (iii), defined as:

		(4.1)
Input observation:	x_i^r	
Encoder (deep neural network):	$z_i = E(x_i^r, \phi)$	
Latent representation:	z_i	
Decoder (deep neural network):	$x_i^g = D(z_i, \theta)$	
Output observation:	x_i^g	

The DeepClean VAE has an encoder and decoder of the form (ii) and (iv), defined as:

		(4.2)
Input observation:	x_i^r	
Encoder (deep neural network):	$\mu_{z,i}, \sigma_{z,i}^2 = E(x_i^r, \phi)$	
Sampling:	$z_i \sim N(\mu_{z,i}, \text{diag}(\sigma_{z,i}^2))$	
Latent representation:	z_i	
Decoder (deep neural network):	$\mu_{x,i}^g = D(z_i, \theta)$	
Sampling:	$x_i^g \sim N(\mu_{x,i}^g, \sigma_x^2 I)$	
Output observation:	x_i^g	

In a VAE, a synthetic output observation can be generated conditionally on a given real input observation, by passing the input observation through the full process (encoder, sampling, decoder, sampling). A key property of either model is that this synthetic observation should approximate the real observation x_i^r, forming a real-synthetic observation pair in which the distance $d(x_i^r, x_i^g)$ is small. As a result, in a VAE, the distribution of all paired synthetic observations (i.e. a mixture distribution of $\mathbb{P}_{x,i}$ for all $i = 1, \ldots, n$) is approximately equal to the 'true' distribution $\mathbb{P}_x$. The role of weights ϕ and θ is to optimise this condition during model training (as one part of the loss function). Repeatedly passing a given real observation through the full process (encoder, sampling, decoder, sampling) will result in multiple similar but not identical synthetic observations, since the process involves random sampling at two points. Synthetic output observations can also be generated independently, by 'uncoupling' the encoder and decoder networks and ignoring the former to sample $z_j \sim \mathbb{P}_z$ and $x_j^g \sim N(D(z_j, \theta), \sigma_x^2 I)$, independently of any

real input observation.

4.1.3 Overview and related work

Generative deep learning. Generative modelling describes a class of models in which we want to learn a generative distribution $\mathbb{P}_g$ to approximate the 'true' distribution $\mathbb{P}_x$ of data $\mathcal{D}_r$. This approximation can help us to understand the generative process behind the data, and in turn to include useful abstractions within downstream modelling tasks. Some causal relationships may also be captured implicitly during model training and consequently become embedded within the generative model [153]. New synthetic observations sampled directly from the distribution $\mathbb{P}_g$ should match some statistical properties and features of the real data $\mathcal{D}_r$. Directly learning the generative distribution $\mathbb{P}_g$ is not straightforward. A simpler task is to use a class of models called latent variable models (LVMs), in which synthetic observations $x_j^g \in \mathcal{X}$ are generated conditional upon unobserved latent variables $z \in \mathcal{Z}$. This is how the decoder $D(\cdot, \theta)$ introduced above functions. However, in order for this to work properly, the latent space $\mathcal{Z}$ must be structured in a way that encourages latent variables $z \in \mathcal{Z}$ to meaningfully encode information about the real data. This is achieved by training a second model, e.g. encoder $E(\cdot, \phi)$, alongside the generative decoder model. Latent variables z_i from this encoder model then form representations of the real observations $x_i^r \in \mathcal{D}_r$. This results in a secondary dual task in the generative deep learning, which is representation learning of the real observations. Several prominent approaches use this template, including VAEs and generative adversial networks (GANs). I return to synthetic data generation (and GANs) in Chapter 5.

Autoencoders and variational autoencoders. Autoencoders seek to embed data in an 'information bottleneck' latent space, which is a lossy transformation that must prioritise the most important features of the data. An autoencoder, as described in Equation 4.1 and Figure 4.1(a), maps real observations directly to representations in a sparse latent space, but this often results in latent representations that are neither meaningful nor well-structured. As a result, autoencoders often encounter issues with fragility and overconfident predictions [154]. Extensions have been proposed to try to alleviate some of these problems [155–158], but autoencoders have largely fallen out of favour, with VAEs and GANs garnering much wider popularity.

For a generative latent variable model, defined by function $D(\cdot, \theta)$, one obvious approach for training the weights θ is for the model objective function to maximise the 'model evidence' over θ. Strictly speaking, this is not a model evidence if θ are considered variables of the generative model, but rather a partially-integrated model likelihood. Semantics aside, this is defined as the probability of observing the entire data under the generative model for fixed θ, i.e. $p(X_r|\theta) = \int_{\mathcal{Z}} (\prod_i p(x_i^r|z, \theta)) p(z) dz$, where $p(z)$ is a prior distribution

function for z. However, maximum likelihood of $p(X_r|\theta)$ (with respect to θ) is intractable. Variational autoencoders (VAEs) [159] resolve this using a technique called variational Bayes, which approximates an intractable distribution or integral using known tractable distributions (or mixture distributions). This has clear parallels to both relative entropy (Section 2.1.3, Equation 2.5) and sequential Monte Carlo (SMC) (Section 3.1.3). The variational Bayes approximation provides a lower bound for the (fixed θ) model evidence. Combining variational Bayes with the autoencoder framework requires the encoder model to learn a 'variational' distribution function $q(z|x_i^r, \phi)$ that approximates the intractable 'true' posterior function of the decoder model, defined as $p(z|x_i^r, \theta) \propto p(x_i^r|z, \theta)p(z)$. In this case, the full posterior, when evaluated over all real observations, is an 'aggregate posterior', i.e. $\prod_i p(z|x_i^r, \theta)$ [160]. For example, in Equation 4.2 the tractable variational distribution is a Gaussian distribution. As shorthand, I denote the variational distribution as $q_\phi(z|x_i^r) = q(z|x_i^r, \phi)$ and I denote all distributions conditional on θ (i.e. generative distribution, true posterior) as $p_\theta(\cdot) = p(\cdot|\theta)$.

During model training, the variational Bayes approach simultaneously learns to encode meaningful information about x_i^r in the latent representation z_i using variational distribution $q_\phi(z|x_i^r)$, and to generate synthetic observations x_i^g from the LVM generative distribution $p_\theta(x|z_i)$. There are dual competing goals to the model training, which are to minimise the divergence between $q_\phi(z|x_i^r)$ and the unknown 'true' posterior $p_\theta(z|x_i^r)$ and to reconstruct real observations with minimal error via the generative distribution $p_\theta(x|z_i)$. The lower bound of variational Bayes can be used as the model objective for these dual goals, in which case it is called the evidence lower bound objective (Equation 4.3).

Evidence lower bound objective. Under the (decoder) generative model, The joint distribution of observation x_i^r and any $z \in \mathcal{Z}$ is $p_\theta(x_i^r, z)$. Bayes' theorem gives $p_\theta(x_i^r) = p_\theta(x_i^r, z)/p_\theta(z|x_i^r) = p_\theta(x_i^r|z)p(z)/p_\theta(z|x_i^r)$. Then:

$$
\begin{aligned}
\log p_\theta(X_r) &= \sum_i \log p_\theta(x_i^r) = \sum_i \log p_\theta(x_i^r) \int_{\mathcal{Z}} q_\phi(z|x_i^r) dz \\
&= \sum_i \int_{\mathcal{Z}} q_\phi(z|x_i^r) \log p_\theta(x_i^r) dz = \sum_i \int_{\mathcal{Z}} q_\phi(z|x_i^r) \log \left(\frac{p_\theta(x_i^r|z)p(z)}{p_\theta(z|x_i^r)} \right) dz \\
&= \sum_i \int_{\mathcal{Z}} q_\phi(z|x_i^r) \log \left(p_\theta(x_i^r|z) \frac{p(z)}{q_\phi(z|x_i^r)} \frac{q_\phi(z|x_i^r)}{p_\theta(z|x_i^r)} \right) dz \\
&= \sum_i \int_{\mathcal{Z}} q_\phi(z|x_i^r) \left(\log p_\theta(x_i^r|z) - \log \left(\frac{q_\phi(z|x_i^r)}{p(z)} \right) + \log \left(\frac{q_\phi(z|x_i^r)}{p_\theta(z|x_i^r)} \right) \right) dz
\end{aligned}
$$

$$\log p_\theta(X_r) = \sum_i \Bigg(\int_{\mathcal{Z}} q_\phi(z|x_i^r) \log p_\theta(x_i^r|z) dz - D_{KL}(q_\phi(z|x_i^r) \,\|\, p(z)) \\ + D_{KL}(q_\phi(z|x_i^r) \,\|\, p_\theta(z|x_i^r)) \Bigg)$$

The evidence lower bound objective (ELBO) is the analytically tractable function $L_{\theta,\phi}(X_r)$:

$$L_{\theta,\phi}(X_r) = \sum_i \left(\int_{\mathcal{Z}} q_\phi(z|x_i^r) \log p_\theta(x_i^r|z) dz - D_{KL}(q_\phi(z|x_i^r) \,\|\, p(z)) \right) \tag{4.3}$$

This satisfies the inequality $\log p(X_r|\theta) \geq L_{\theta,\phi}(X_r)$, since $D_{KL}(q_\phi(z|x_i^r) \,\|\, p_\theta(z|x_i^r))$ is always non-negative. In an 'ideal encoder' setting, the model has arbitrarily high encoding capacity and the ELBO efficiently minimises the divergence between variational and true posterior, while jointly maximising the model evidence, which makes the lower bound tight. However, the KL-divergence $D_{KL}(q_\phi(z|x_i^r) \,\|\, p_\theta(z|x_i^r))$ is intractable, so the tightness of the bound cannot be reliably estimated. The ELBO splits into two conflicting terms, $L_{\theta,\phi}(X_r) = L^{(1)}_{\theta,\phi}(X_r) - L^{(2)}_{\phi}(X_r)$, with:

$$L^{(1)}_{\theta,\phi}(X_r) = \sum_i \int_{\mathcal{Z}} q_\phi(z|x_i^r) \log p_\theta(x_i^r|z) dz, \quad L^{(2)}_{\phi}(X_r) = \sum_i D_{KL}(q_\phi(z|x_i^r) \,\|\, p(z))$$

In the special case in which these distributions are multivariate Gaussians (as in Equation 4.2), these loss terms are estimated as the following, where $(\mu_{z,i}, \sigma^2_{z,i}) = E(x_i^r, \phi)$ and $z_i^{(k)}$ are n_z samples from the latent distribution space:

$$L^{(1)}_{\theta,\phi}(X_r) = \sum_{i=1}^{n} -\frac{1}{n_z \sigma_x^2} \sum_{k=1}^{n_z} d(x_i^r, D(z_i^{(k)}, \theta)), \quad z_i^{(k)} \sim N(\mu_{z,i}, \mathrm{diag}(\sigma^2_{z,i})), \quad k = 1, \ldots, n_z$$

$$L^{(2)}_{\phi}(X_r) = \sum_{i=1}^{n} \frac{1}{2} (1_{d_z}^T (\mu_{z,i} + \sigma^2_{z,i} - \log(\sigma^2_{z,i})) - d_z), \quad 1_{d_z} = (1, \ldots, 1)^T \in \mathbb{R}^{d_z}$$

The first term $L^{(1)}_{\theta,\phi}(\mathcal{D}_r)$ minimises the reconstruction error of synthetic output observations compared to real input observations, and is therefore associated with the performance of the decoder. The KL-divergence term $L^{(2)}_{\phi}(X_r)$ regularises the latent space by enforcing similarity between $q_\phi(z|r_i^r)$ and a known prior $p(z)$. These two terms create competition between the quality of the latent representation learning and the generation of high fidelity synthetic observations with low reconstruction error. The ELBO itself cannot distinguish between models that use the latent variable to learn meaningful representations (e.g. an autoencoder) and non-autoencoder models that map the observations X_r to itself via a high-dimensional latent space $\mathcal{Z}$ that does not learn a lower-dimensional representation [161]. However, the competition between the two terms can be managed by appropriate choices of network architectures and distributions (e.g. prior and variational), or balanced

by tuning a global hyperparameter β, in a β-ELBO objective function [162]:

$$L_{\theta,\phi}(X_r) = L^{(1)}_{\theta,\phi}(X_r) - \beta L^{(2)}_{\phi}(X_r) \tag{4.4}$$

In general, the use of variational Bayes helps to encode more robust and meaningful latent representations, since it encourages the variational distribution to place probability mass on a range of latent values that could feasibly generate a synthetic observation similar to the real input observation, rather than placing all of the mass on a single point in $\mathcal{Z}$. Regularisation ensures this range of latent values are close to each other in the latent representation space, which means that similar input observations have similar latent representations and vice versa.

Naïve Monte Carlo integration estimates (Equation 3.8) of the ELBO gradient are impractical due to high variance [159], so model training uses stochastic gradient descent. Stochasticity during model training helps to further improve regularisation of the latent representation space, resulting in more robust latent representations. However, stochastic variables are not automatically differentiable, so a 'reparameterisation trick' is needed to allow backpropagation [159]. Optimising the ELBO can also be improved by annealing the KL-divergence term early in model training [163, 164].

Information theory, ELBO and representation learning. Though there is tension between the quality of latent representations and of reconstructed synthetic observations, the former is important for artefact detection because latent representations must learn salient features of 'valid' time-series data in order for post-processing discrimination of observations containing artefacts. The trade-off between the reconstruction error and KL-divergence terms can be described using phase diagrams in a rate-distortion plane. The rate R_ϕ is defined as the excess number of 'bits' or 'nats' needed to encode real observations in the latent space using an ideal coding channel, and the distortion $D_{\theta,\phi}$ is a measure of the amount of lossy compression. For a VAE-type model, these are defined as [165]:

$$\begin{aligned} D_{\theta,\phi} &= -\int_{\mathcal{X}\times\mathcal{Z}} p(x) q_\phi(z|x) \log p_\theta(x|z) dx dz \\ R_\phi &= \int_{\mathcal{X}\times\mathcal{Z}} p(x) q_\phi(z|x) \left(\log \frac{q_\phi(z|x)}{p(z)} \right) dx dz \end{aligned} \tag{4.5}$$

Though the true data distribution $p(x)$ is intractable, for the purposes of visualising models in the rate-distortion plane, this can be estimated using an empirical approximation $\hat{p}(x)$ (e.g. Monte Carlo integration, Equation 3.8). Rate and distortion are concepts from information theory. Together, they satisfy the following inequality involving entropy $H(\cdot)$ (Equation 2.4) and mutual information $I(\cdot\,,\cdot)$ (Equation 2.6), where $E(X_r)$ is the entire

encoded data $E(X_r, \theta) = (E_\mu(x_1^r, \theta), \ldots, E_\mu(x_n^r, \theta))$:

$$0 \leq H(X_r) - D_{\theta,\phi} \leq I(X_r, E(X_r, \theta)) \leq R_\phi$$

In rate-distortion theory, the optimal rate is considered as a function of distortion in the rate-distortion plane and the goal is to maximise the rate without exceeding some predefined expected distortion, i.e. $\inf_{\theta,\phi} R_\phi$ subject to $D_{\theta,\phi} \leq D^*$. A Legendre transformation of this optimisation problem with $\beta^{-1} = \partial R_\phi / \partial D_{\theta,\phi}$ converts this optimisation problem to the β-ELBO objective (Equation 4.4):

$$\inf_{\theta,\phi} R_\phi + \frac{1}{\beta} D_{\theta,\phi} = \inf_{\theta,\phi} \frac{1}{\beta} \int_{\mathcal{X}} \hat{p}(x) \left(- \int_{\mathcal{Z}} q_\phi(z|x) \log p_\theta(x|z) dz + \beta D_{KL}(q_\phi(z|x) \,\|\, p(z)) \right) dx$$

This reframes the tension between latent representation and reconstruction error within the model objective in terms of information theory, since making either of the inequalities $I(X_r, E(X_r, \theta)) \leq R_\phi$ and $H(X_r) - I(X_r, E(X_r, \theta)) \leq D_{\theta,\phi}$ tighter will generally result in the other becoming looser.

Additionally, this highlights the flaw in using generative deep learning algorithms for representation learning, which is by definition ill-posed [161, 166], since learning 'good' latent representations (i.e. that encode meaningful information about the input observations) is largely detached from the model objective during training, while many possible generative models can achieve an identical model evidence through different variational distributions [161]. Furthermore, a sufficiently powerful decoder may entirely neglect the latent representation to store an encoding within the decoder weights θ instead, which prevents generalisation to other unseen datasets. Under the information-theoretic principle of 'minimum descriptive length', which motivated VAE predecessors such as the Helmholtz machine [167], a 'bits back' argument provides interpretation about the 'extra information' gained about x_i^r, when $z \in \mathcal{Z}$ is sampled from $q_\phi(z|x_i^r)$ rather than from the prior $p(z)$ [168]. This can be leveraged alongside inverse autoregressive flow (IAF) transformations to create a VAE-type generative model with a decoder capable of explicitly modelling known structural elements of the data that are viewed as unnecessary to its latent representation [153, 169], but with a lossy latent representation space that captures other unspecified features [166]. This information preference allows an element of control over the representation learning process.

Importance weighting and ELBO. Instead of using more flexible and expressive variational distributions (e.g. IAF transformations) in order to minimise the KL-divergence between the variational and posterior distributions, an alternative is to explicitly tighten the evidence lower bound using importance weighting [170], which is similar to Equation 3.9 in Section 3.1.3. Denoting unnormalised importance weights for the joint distribution

$p_\theta(x_i^r, z)$ as $w_k = p_\theta(x_i^r, z^{(k)})/q_\phi(z^{(k)}|x_i^r)$ and expectation with respect to $q_\phi(z|x_i^r)$ as $\mathbb{E}_{q_\phi}$, the importance weighted estimate $L_{\theta,\phi,k}(X_r)$ achieves a tighter bound than the ELBO:

$$L_{\theta,\phi}(X_r) = \mathbb{E}_{q_\phi}\big[\log w_1\big] = L_{\theta,\phi,1}(X_r) \leq L_{\theta,\phi,k}(X_r) = \mathbb{E}_{q_\phi}\left[\log \frac{1}{n_k}\sum_{k=1}^{n_k} w_k\right] \leq \log p_\theta(X_r)$$

Principal component analysis. A simple autoencoder alternative that does not involve deep learning is to use principal component analysis (PCA), with the PCA representation as an encoder and PCA reconstruction as a decoder. PCA is a statistical technique for dimension reduction of high-dimensional data, via a coordinate system that describes the maximal amount of data variation. For the (mean-centred) $X_r \in \mathbb{R}^{n\times T}$ (an $n \times T$ matrix), the PCA decomposition is defined as $Z = X_r W$, where the columns of $W \in \mathbb{R}^{T\times d}$ are the d (orthogonal) eigenvectors of $X_r X_r^T$ that correspond to the largest ordered eigenvalues. The subsequent PCA reconstruction is $X_g = ZW^T = X_r W W^T$. The PCA latent representation z_i (also referred to as the principal component scores) and reconstruction of a (mean-centred) real observation $x_i^r = (x_{i1}^r, \ldots, x_{iT}^r)^T$ are:

$$z_i = (z_{i1}, \ldots, z_{id})^T = (\sum_{j=1}^{T} x_{ij}^r W_{j1}, \ldots, \sum_{j=1}^{T} x_{ij}^r W_{jd})^T$$
$$x_i^g = (x_{i1}^g, \ldots, x_{iT}^g)^T = (\sum_{j=1}^{d} z_{ij} W_{1j}, \ldots, \sum_{j=1}^{d} z_{ij} W_{Tj})^T$$

This maximises the amount of variance preserved while minimising the total distance between X_r and their reconstructions, i.e. $\sum_i d(x_i^r, x_i^g)$. Under some conditions (which include independent Gaussian noise), PCA also minimises information loss from X_r to its latent representation. Additionally, PCA is equivalent to a one-layer autoencoder with linear activations, in which the encoder weights span the same subspace as the principal components [171].

In practice, it is almost always more efficient to perform PCA by computing the singular value decomposition (SVD) of matrix X_r, rather than via the eigendecomposition of the (unscaled) covariance $X_r X_r^T$. The truncated SVD of a matrix X_r is $\tilde{X}_r = U\Sigma V^T$, where $U \in \mathbb{R}^{n\times d}$ and $V \in \mathbb{R}^{T\times d}$ are semi-orthogonal, and $d < \text{rank}(X_r)$. The singular values on the diagonal of the diagonal matrix $\Sigma \in \mathbb{R}^{d\times d}$ correspond to the PCA eigenvalues. $\tilde{X}_r$ provides a low-rank approximation to X_r and is equal to X_g above, while $Z = U\Sigma$ and $W = V$.

However, PCA is not particularly robust to outliers. Outliers, e.g. samples containing artefacts, may have high leverage when viewing PCA as a linear regression [172]. By definition, the distance $d(x_i^r, x_i^g)$ will be relatively small for this set of samples. This would mean that the reconstruction error is unlikely to be a good classifier for identifying

artefacts. There are several alternatives to this, including robust PCA methods that similarly seek a low-rank approximation of noisy or corrupted data [173]. I have briefly explored this below with $L1$-PCA.

Another alternative for classifying outliers is to use a statistic based on the PCA scores. The PCA reconstruction is a projection onto a lower-dimensional hyperplane and the reconstruction error is the orthogonal distance to this hyperplane. PCA eigenvectors describe orthogonal vectors of maximal variance, which may be influenced by the presence of artefacts. As a result, observations containing artefacts may be very dissimilar from valid observations and yet exist close enough to the linear hyperplane that they have small reconstruction error. These artefacts may be found instead by a distance metric applied to the projections on the hyperplane, e.g. their ℓ_∞-norm (Chebyshev distance from the origin). However, principal components describe a decreasing proportion of explained variance and the first PCA score will have larger variance, so may dominate this distance. Rescaling the PCA scores by their singular values will mitigate against this, as the SVD matrix $U = Z\Sigma^{-1}$ is orthogonal. This is very similar to the Mahalabonis distance from the origin in the lower-dimensional 'latent' space, but with the Chebyshev distance instead of the Euclidean distance. Therefore, a potential test statistic for observation x_i^r is:

$$t_i = \|u_i\|_\infty = \max_j |U_{ij}| = \max_j |Z_{ij}/s_j|, \;\; s_j = \Sigma_{jj} \tag{4.6}$$

I used PCA as a baseline model for comparison with my VAE-based artefact detection framework. The number of PCA principal components d is somewhat comparable to the latent dimension d_z of the VAE, in the sense that each sample can be encoded as d_z components. As such, I referred to both as the latent dimension in Section 4.2. However, it is worth noting that the VAE is a much more flexible model, so it is expected that the VAE is more efficient at encoding and decoding than PCA, and direct comparison of the two methods should be interpreted accordingly.

$L1$ principal component analysis. Under the ℓ_2-norm, the following low-rank optimisation problems are equivalent for fixed $d < \text{rank}(X_r)$ [174]. Their solution is given by PCA, with $Z = X_r V$ in problems P_2 and P_3.

$$\begin{aligned}
P_1 &: Z, V = \arg\min_{R,S} \|X_r - SR^T\|, \;\; R \in \mathbb{R}^{T\times d}, \;\; S \in \mathbb{R}^{n\times d} \\
P_2 &: V = \arg\min_{R} \|X_r - X_r R R^T\|, \;\; R \in \mathbb{R}^{T\times d}, \;\; R^T R = I \\
P_3 &: V = \arg\max_{R} \|X_r R\|, \;\; R \in \mathbb{R}^{T\times d}, \;\; R^T R = I
\end{aligned} \tag{4.7}$$

The ℓ_1-norm (Manhattan distance) is more robust and less sensitive to outliers. However, the three optimisation problems are much more difficult to solve when the ℓ_1-norm is

used instead of the ℓ_2-norm, and they are also no longer equivalent. Previous research has focused most on P_3 for the ℓ_1-norm [175]. This can be translated into more tractable optimisation problems involving the matrix nuclear norm, via a bit-flipping algorithm. As problems P_2 and P_3 do not necessarily admit similar solutions, the solution to P_3 does not necessarily perform well as an autoencoder. However, it should provide a more robust test statistic for classifying artefacts based on the PCA scores.

4.2 DeepClean

Using this generative deep learning methodology, I constructed an artefact detection framework for small segments of physiological waveform data, which I named DeepClean. This involved a preprocessing step to create a 'mostly clean' training dataset, and to identify and label a suitably chosen test set. After VAE training, I generated synthetic observations corresponding to each real observation, using both encoder and decoder networks. The idea behind the artefact detection is that 'valid' waveform data comes from a 'true' underlying distribution $\mathbb{P}_x$ and artefacts in the waveform come from a 'true artefact' distribution $\mathbb{P}_a$. A single real observation comes from a mixture of these distributions, depending on the amount of 'valid' and artefactual waveform it contains. By training a VAE to generate observations from a distribution that approximates $\mathbb{P}_x$ (with minimal exposure to $\mathbb{P}_a$), any subsequent test observations that are mostly from $\mathbb{P}_a$ will have sub-optimal synthetic reconstructions under the VAE. I used automatic post-processing to identify artefacts, based on the reconstruction error between real-synthetic observation pairs (in practice, the synthetic observation is replaced in this step by the decoder mean $\mu^g_{x,i}$). I evaluated the DeepClean framework on 10s long observations from a dataset containing high-frequency arterial blood pressure (ABP) waveform data, obtained as part of routine ICU clinical care. I repeated the process for a number of different latent dimensions.

4.2.1 Methods and artefact detection

Preprocessing. The DeepClean VAE model must be trained on 'clean' preprocessed waveform data in order to learn to encode latent representations of 'valid' real observations without also learning features of artefactual data. This preprocessing involved basic thresholding heuristics that removed large, grossly abnormal segments of the waveform (Figure 4.2). Waveform segments were marked as 'abnormal' if any of the following occurred:

(i) values exceeded global, signal-specific thresholds (for the ABP waveform, $c_{\min} = -5$mmHg and $c_{\max} = 240$mmHg),

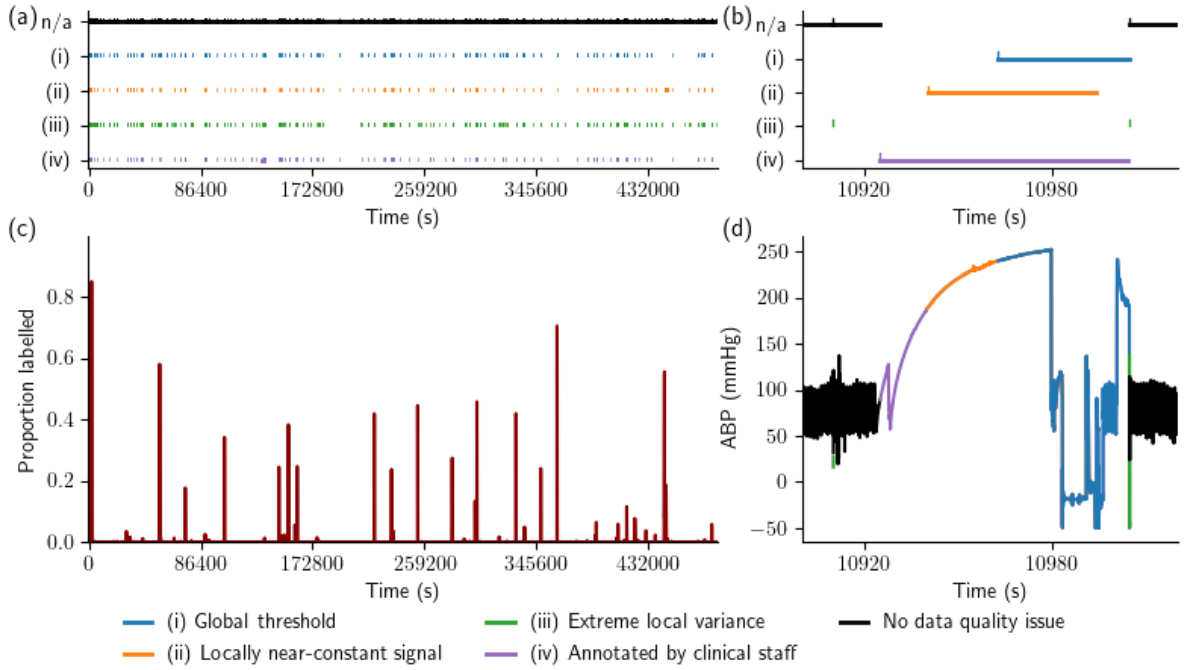

Figure 4.2: The heuristic preprocessing step for DeepClean. (a) and (b) show the location of marked 'abnormal' sections across the whole dataset and for a short example segment. The widths of these segments were often short, so additional vertical lines indicate the start of a marked section. (c) shows the proportion of data within each 100s window that was marked as 'abnormal' in preprocessing and (d) shows marked data for the example segment.

(ii) the range within a sub-pulse window stayed within an extreme small threshold, i.e. the signal was essentially static (for the ABP waveform, the minimum local change was 0.5mmHg in a window of length 0.25s),

(iii) an extreme local change in the signal within the same sub-pulse window exceeded a much larger threshold (for the ABP waveform, the maximum local change was 80mmHg in a window of 0.25s),

(iv) real-time signal quality annotations by clinical staff were included within the dataset.

In addition, successive marked segments were merged if the proportion of marked 'abnormal' waveform was greater than 0.7 within any given time interval.

These preprocessing heuristics were straightforward to implement and did not require detailed manual observation-level annotation. Importantly, they did not require particularly high sensitivity or specificity in removing artefacts, but simply needed to reduce the proportion of abnormal data across the whole dataset in order to create a predominantly 'clean' dataset for VAE training. In practice, the proportion of artefacts within a physiological waveform is likely to be reasonably small, so I was able to apply conservative thresholds for these heuristics, to allow only most artefacts to be removed without unwanted removal of valid data with unusual morphology. The VAE can be trained on the raw waveform data without this preprocessing step. However, excluding the preprocessing will make model training more difficult, since abnormal waveform segments

have larger reconstruction errors and so contribute a disproportionate weighting in the model objective.

Though the DeepClean framework is unsupervised, a 'ground-truth' test set manual annotations was required to evaluate model performance against this 'gold standard' annotation. I wanted a closely balanced test set with a similar proportion of 'valid' and artefactual observations, but the raw waveform was unbalanced in favour of normal data rather than abnormal. To avoid potential selection bias by mutually choosing test set observations, I sampled these randomly from the unprocessed data in a two-step process. First, I sampled a 100s window from the entire waveform, with bias towards 100s windows that contained higher proportions of labelled 'abnormal' data, then I uniformly sampled a 10s observation from within this 100s window. The aim was that this two-step process would result in test observations having an approximately similar chance of being artefact-free or of containing an artefact. The choice of the larger 100s window length in this process was an educated guess to enable this. I repeated the selection process with replacement until I had a test set containing n^* observations x_j^* (where $n^* = 200$). I then annotated each 10s test observation on an observation-wide basis as either containing an artefact or not, with annotations independently verified by Ari Ercole. I also annotated segments of artefact within each test observation to allow assessment of within-observation artefact detection.

After performing the preprocessing step, and removing both test observations and waveform segments marked as 'abnormal', I split the remaining waveform was into 10s training and validation observations, forming a dataset $\mathcal{D}_r$ with observations X_r. The test set observations were denoted $X^* = (x_1^*, \ldots, x_{n^*}^*)$.

Model specification. In the DeepClean framework, I used a VAE with deep convolutional neural networks (CNNs) for both encoder and decoder models. Recurrent neural networks (RNNs) may perform better for general time-series, but high-frequency ABP waveform data is quasi-periodic and highly-structured. This motivated the choice of CNNs, since these allow the model to learn translation-invariant local patterns in a spatial hierarchy of increasing scale. The encoder architecture included three small convolutional layers, alternated with pooling layers to increase the receptive field and decrease tensor granularity, ending with two dense layers that split into mean and variance parameters for the variational distribution (Table C.1). The decoder architecture was a mirror image of this in reverse, with pooling replaced by up-sampling. Both encoder and decoder contained approximately 20,000 trainable parameters, with the exact number depending on the latent space dimension.

I used the β-ELBO (Equation 4.4) as the training objective function, and used standard distributions for both variational and generative distributions (Equation 4.2). I used a

d_z-dimensional multivariate Gaussian $N(0, I)$ as the prior distribution for the latent representations. In theory, diagonal covariance matrices for the variational distribution encourage independent feature learning and therefore an efficient encoding in the latent representation space, though this can lead to latent dimension 'pruning' [163]. I fixed the variance term σ_x^2, but this satisfies $\beta^{-1} = \sigma_x^2$ under the distribution assumptions that I made, so it was essentially the main tunable hyperparameter.

Post-processing for artefact detection. The DeepClean framework identifies artefacts within waveform segments by comparing real observations to their synthetic reconstruction means from the VAE model. An artefact is therefore identified in observation x_i^r if the mean squared error (MSE) distance satisfies $d(x_i^r, \mu_{x,i}^g) > \gamma$, for a suitable threshold γ. In order for DeepClean to be fully unsupervised (and completely blind to test data), the threshold must be automatically defined as a function of the training data, i.e. $\gamma = \gamma(X_r)$, instead of being specified post-hoc by the user. I set this automatic threshold to be the 90^{th} percentile of the sum of MSE values between training and validation real-synthetic observation pairs. If the threshold γ is decreased, a larger number of observations are classified as artefacts, regardless whether this classification is correct. Therefore, this increases the specificity while decreasing the sensitivity. I assessed the DeepClean classification using both of these binary performance metrics. I also evaluated performance using the area under the curve of the receiver operator characteristic (ROC AUC), when $\gamma(X_r)$ was allowed to vary across all training and validation set MSE percentiles. Figure 4.4 highlighted that the distribution of MSE values was dependent on at least one of the model hyperparameters (the latent dimension), validating this automatic thresholding approach.

In addition to using the reconstruction MSE, I also classified observations by their latent representations z_i, which is similar to the rescaled PCA score vector (Equation 4.6). As before, this test statistic used the Chebyshev ℓ_∞-norm: $t_i = \max_j |Z_{ij}/s_j|$. The singular values s_j here were calculated using SVD on the latent representation matrix $Z \in \mathbb{R}^{T \times d_z}$. The same process was used to classify artefacts based on this statistic, with the 90^{th} percentile of values of the statistic from the training set observations used as a threshold.

For each test observation that was identified as containing an artefact (by manual annotation, by DeepClean or by PCA), I calculated within-observation MSE values on a 1s sliding-window (approximately one or two typical heartbeat periods) and applied the same automatic thresholding method, in order to identify within-observation artefacts. If the MSE for a given 1s window exceeded the automatic threshold, then the entire 1s window was identified as artefact. I used a stricter threshold in this instance, which was the training and validation set MSE 99^{th} percentile. There were several reasons behind this decision. Firstly, I argued that it is more useful to restrict the within-observation false positive rate than the between-observation false positive rate, since former will result in a much larger

overall number of false artefact segments than the latter. Consequently, this could place a higher burden on clinical staff to verify the more granular artefact segments. Secondly, for an observation that contains both an artefact segment and a non-artefact segment, the whole-observation MSE is an average of MSE values over both segments separately, even though the artefact segment often resulted in a distorted synthetic reconstruction. The MSE for the artefact segment will almost certainly be much higher, but averaging over both segments down-weights this. This means that MSE values on more granular sliding-windows should have an empirical distribution with much wider tails. The idea is that these wider distribution tails should allow a stricter threshold without compromising on the true positive rate.

4.2.2 Results and discussion

Data. Fully anonymous ABP waveform data was obtained as part of routine ICU clinical care from a single adult patient monitored almost continuously throughout an ICU stay of several days (486,984s). This ABP waveform was obtained from a standard indwelling arterial line connected to a pressure transducer and was recorded at frequency of 125Hz. Under UK regulations, ethical approval was not required for reuse of anonymous data that was obtained as part of routine clinical practice, for clinical research.

First, I performed the preprocessing step on this ABP waveform, marking 11,082s (2.28%) as 'abnormal'. I then formed a (reasonably balanced) test set of size 200 and annotated each 10s observation in this set, marking 130 test observations as containing an artefact. Having removed test set observations and all segments marked as 'abnormal', I then split the remaining waveform into 10s observations. I shuffled and split these into training and validation sets, containing 37,821 and 4,728 observations respectively (90%-10% split).

Model training. For each of 8 latent dimensions ($d_z = 2, 3, 4, 5, 10, 20, 50, 100$), I trained the DeepClean VAE model five times, and selected the model with the smallest validation loss (which was not directly related to artefact detection performance). Training the VAE required under 10 minutes of computation time on average, although this increased with latent dimension. I also calculated PCA representations and reconstructions, for the same latent dimensions (i.e. the number of PCA components).

Between-observation performance. The DeepClean VAE model was able to accurately reconstruct real observations, with much higher fidelity than the PCA reconstruction (Figure 4.3). In particular, it was able to encode sub-pulse components, e.g. the dicrotic notch and diastolic peak, even when the latent dimension was small. In comparison, the

PCA reconstructions were especially poor for small latent dimensions, and overfit to the data when the latent dimension was large.

DeepClean substantially outperformed the PCA baseline in between-observation artefact detection, using the reconstruction error $d(x_i^r, x_i^g)$ with the automatic 90^{th} percentile threshold (Table 4.1). This was consistent across all latent dimensions. In particular, DeepClean had similar specificity to PCA but significantly higher sensitivity (Figure 4.4). It is worth noting that the accuracy of artefact detection did not depend monotonically on the quality of the reconstruction, which can be seen in Figure 4.4. Varying the automatic threshold via MSE percentiles, DeepClean also had much higher AUC in all cases (Figure C.1 shows all ROC curves). DeepClean tended to perform best when the latent dimension was relatively small, and achieved highs of 0.994 in AUC and 0.945 in accuracy when the latent dimension was 5. However, the reverse occurred when using the rescaled latent representation scores $\|u_i\|_\infty = \max_j |Z_{ij}/s_j|$ for classification. In this case, DeepClean had poor sensitivity and PCA achieved much better results. $L1$-PCA performed well as an encoder-like model, but the reverse transformation poorly as a decoder-like model, which meant that the reconstructions were of limited utility. The reason for this is likely that the optimisation problems P_2 and P_3 in Equation 4.7 do not admit similar solutions in this case. However, the ℓ_1-norm is more robust to outliers, and $L1$-PCA did perform slightly better than vanilla PCA. Overall, DeepClean outperformed $L1$-PCA, when the best performing classifier was used for each model.

In general, there was a clear distinction between MSE values of artefacts and of 'valid' data using DeepClean reconstructions (Figure 4.4, with log MSE for easier visualisation), because DeepClean was able to distinguish between observations from 'true' waveform distribution $\mathbb{P}_x$ and observations not from this distribution (i.e. those mostly from the 'true' artefact distribution $\mathbb{P}_a$). In contrast, the PCA reconstructions did not appear to make any such distinction between observations, so the distribution of (log) MSE values using PCA reconstructions were similar for all observations, regardless of any training/test or artefact/non-artefact split. The regularisation term in the VAE model objective that penalises new observations that are unlike any it has previously seen during training (i.e. artefacts), resulting in latent representations with low probability mass under the 'aggregate variational' distribution, i.e. the mixture $\prod_i q_\phi(z|x_i^r)$). However, PCA does not appear to penalise unseen observations as strongly. In both cases, non-artefact test observations followed a similar distribution of (log) MSE values to the training observations, which meant that the specificity was generally about 0.9 (matching the 90% percentile threshold). As the thresholds were dependent on the training set reconstructions, these were not aligned for PCA and for DeepClean, which meant that a given synthetic observation may be classified differently depending on whether it was a reconstruction from DeepClean or from PCA.

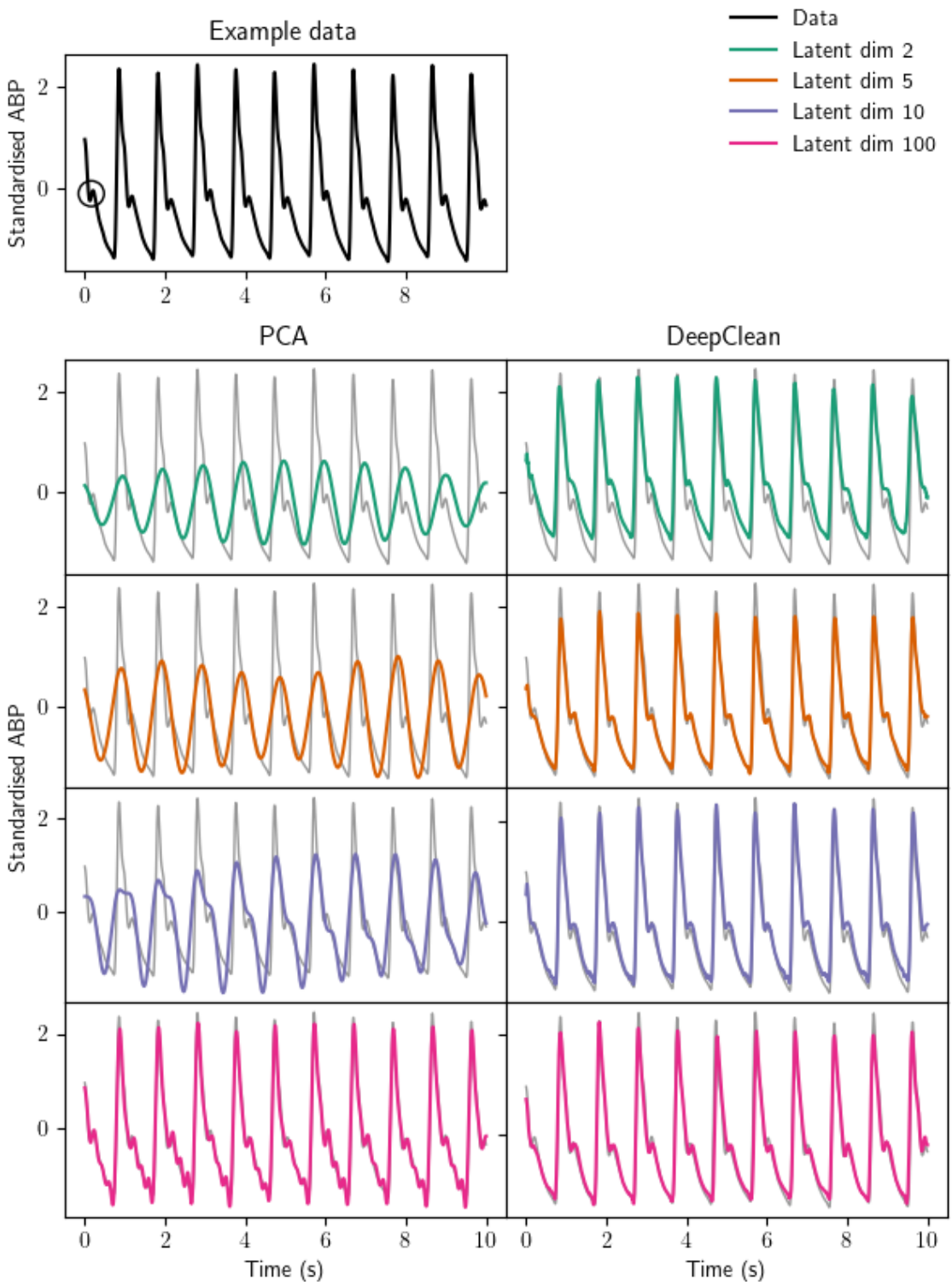

Figure 4.3: Example 10s observations with their PCA and DeepClean VAE synthetic reconstructions. PCA performed poorly without a large number of principal components, whereas the DeepClean VAE managed to encode sub-pulse components, including the dicrotic notch and diastolic peak (circled in the example data).

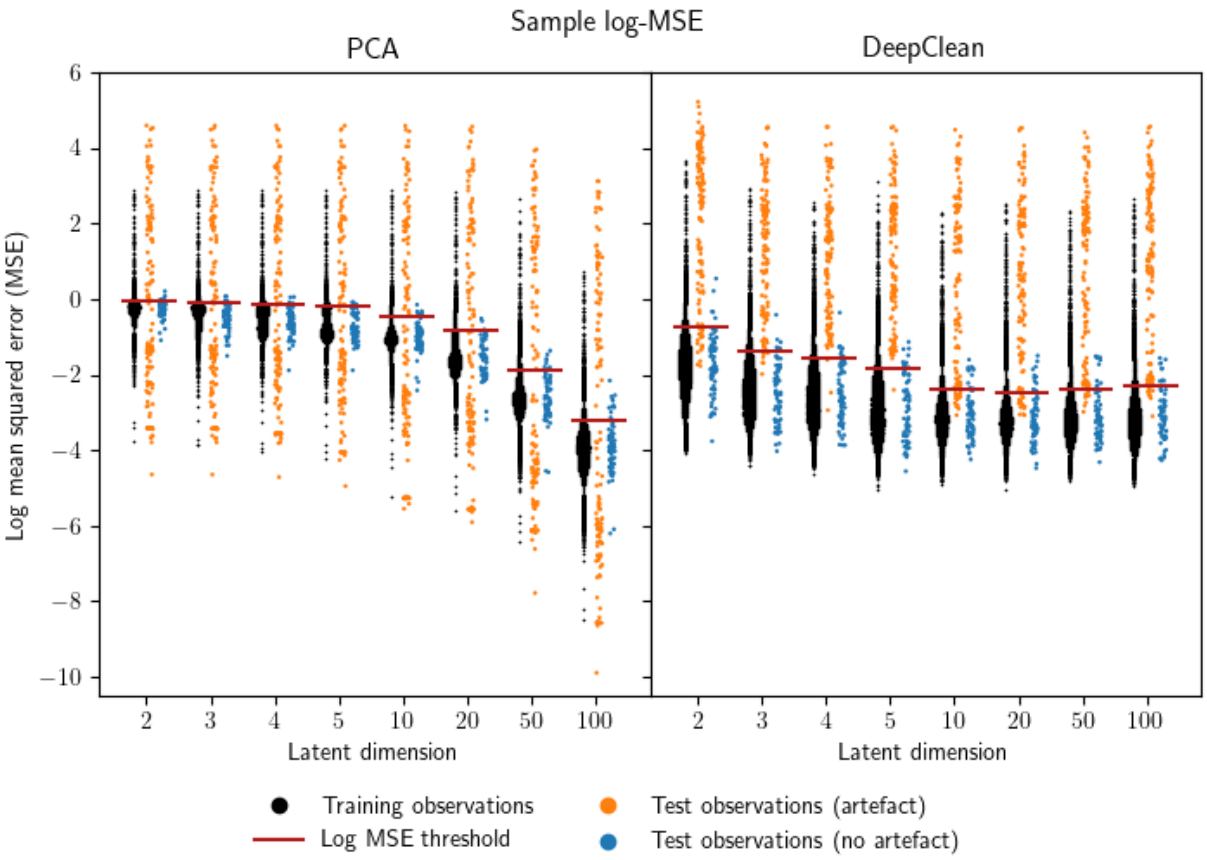

Figure 4.4: Log mean squared error (MSE) values for both PCA and DeepClean reconstructions, with increasing latent dimension. The distribution of training set log MSE values is shown alongside the automatic threshold (i.e. training and validation set 90^{th} percentile). Test observations were categorised according to whether or not they contained an artefact in 'ground-truth' manual annotations. 'Ground-truth' artefacts in test set observations below the threshold were incorrectly categorised as false negatives by the artefact detection algorithm, and 'ground-truth' non-artefacts above the threshold were incorrectly categorised as false positives. Therefore, the proportion of 'ground-truth' artefacts above the threshold is the sensitivity and the proportion of 'ground-truth' non-artefacts below the threshold is the specificity.

	Sensitivity					Specificity				
	$d(x_i^r, x_i^g)$		$\|u_i\|_\infty$			$d(x_i^r, x_i^g)$		$\|u_i\|_\infty$		
d_z	PCA	VAE	PCA	$L1$-PCA	VAE	PCA	VAE	PCA	$L1$-PCA	VAE
2	0.454	0.854	0.839	0.857	0.031	0.900	0.929	0.932	0.966	0.929
3	0.462	0.977	0.848	0.866	0.492	0.900	0.886	0.920	0.932	0.957
4	0.462	0.977	0.848	0.643	0.761	0.914	0.886	0.909	0.932	0.943
5	0.462	0.992	0.866	0.821	0.431	0.914	0.857	0.909	0.932	0.929
10	0.470	0.869	0.857	0.920	0.508	0.900	0.843	0.909	0.932	0.843
20	0.470	0.877	0.795	0.830	0.408	0.929	0.814	0.909	0.943	0.900
50	0.454	0.931	0.830	0.839	0.592	0.886	0.857	0.943	0.943	0.886
100	0.446	0.908	0.839	0.857	0.469	0.828	0.871	0.932	0.909	0.900

	Accuracy					ROC AUC				
	$d(x_i^r, x_i^g)$		$\|u_i\|_\infty$			$d(x_i^r, x_i^g)$		$\|u_i\|_\infty$		
d_z	PCA	VAE	PCA	$L1$-PCA	VAE	PCA	VAE	PCA	$L1$-PCA	VAE
2	0.610	0.880	0.885	0.905	0.345	0.471	0.967	0.930	0.939	0.551
3	0.615	0.925	0.895	0.895	0.655	0.478	0.984	0.923	0.941	0.679
4	0.620	0.945	0.900	0.770	0.825	0.482	0.987	0.920	0.918	0.879
5	0.620	0.945	0.900	0.870	0.605	0.487	0.994	0.917	0.923	0.655
10	0.620	0.860	0.900	0.930	0.625	0.496	0.953	0.923	0.954	0.688
20	0.630	0.855	0.900	0.880	0.580	0.528	0.960	0.907	0.955	0.652
50	0.605	0.905	0.900	0.885	0.695	0.474	0.976	0.936	0.931	0.667
100	0.580	0.895	0.920	0.880	0.620	0.478	0.969	0.935	0.933	0.639

Table 4.1: Artefact classification performance for PCA, $L1$-PCA and DeepClean (VAE). Two different statistics were used for classification, the reconstruction error $d(x_i^r, x_i^g)$ and the statistic $\|u_i\|_\infty$ based on the rescaled score vector (Equation 4.6). ROC AUC is the area under the receiver operating characteristic curve (Figure C.1). $L1$-PCA performed extremely poorly as an autoencoder model for all latent dimensions, but provided robust 'encodings'. As a result, only $\|u_i\|_\infty$ is provided for $L1$-PCA. All classifiers had comparable specificity, i.e. the proportion of correctly identified non-artefact observations. However, the sensitivity of each model was very dependent on the test statistic. DeepClean had the highest ROC AUC values, and was the only method that had high sensitivity using the reconstruction error $d(x_i^r, x_i^g)$. For each performance metric, the best performing model was highlighted (PCA or VAE, across all latent dimensions).

	Mean proportion correctly identified of the:							
	Entire sample		Artefact within sample		Non-artefact within sample		Proportion 100% correct	
d_z	PCA	VAE	PCA	VAE	PCA	VAE	PCA	VAE
2	0.507	0.818	0.779	0.896	0.950	0.943	0.235	0.595
3	0.500	0.818	0.778	0.889	0.945	0.951	0.220	0.555
4	0.504	0.796	0.779	0.893	0.956	0.949	0.235	0.490
5	0.505	0.820	0.776	0.889	0.944	0.949	0.260	0.560
10	0.530	0.752	0.796	0.857	0.946	0.940	0.315	0.510
20	0.551	0.794	0.807	0.853	0.946	0.965	0.350	0.510
50	0.545	0.777	0.772	0.853	0.949	0.955	0.340	0.570
100	0.556	0.776	0.926	0.755	0.982	0.965	0.375	0.540

Table 4.2: Assessment of within-observation artefact detection, for both PCA and DeepClean (VAE). For each observation, I calculated the proportion that was correctly identified with respect to the 'ground-truth' manual annotation, and report the average over all observations. In addition, I calculated the proportion of within-observation artefact segments that were correctly identified as artefactual, and similarly for within-observation non-artefacts. Finally, I calculated the proportion of observations in which the model artefact detection was 100% correct. For each performance metric, the best performing model was highlighted (PCA or VAE, across all latent dimensions).

Within-observation performance. The second task of the granular within-observation artefact detection was a more difficult problem. This was only possible using reconstruction error between a real observation and its synthetic counterpart. Both PCA and DeepClean performed less well in this (Table 4.2). For each observation, I calculated the proportion within the observation that was correctly categorised with respect to the 'ground-truth' manual annotation. In addition, for each observation that contained at least one 'ground-truth' artefact segment, I calculated the proportion of 'ground-truth' artefact segments within the observation that were correctly identified as artefact by the models. Similarly, for observations containing at least one 'ground-truth' non-artefact segment, I calculated the proportion of 'ground-truth' non-artefact segments within the observation that were correctly identified. These three scores were roughly analogous to the binary performance metrics (accuracy, sensitivity and specificity respectively). For approximately half of the test observations, there was 100% agreement in the within-observation artefact labels between DeepClean and the 'ground-truth' annotation. These mostly corresponded to cases in which the observation was either entirely artefact or contained zero artefact segments, according to the 'ground-truth' manual annotation. PCA was also able to classify some of these observations 100% correctly, but it also classified others 100% incorrectly. I included some examples of the within-observation artefact detection (both 'ground-truth' and DeepClean) in Figure 4.5, which is discussed in more detail in the next few paragraphs.

Imputation. Imputation methods should be considered a key element of artefact detection algorithms, since removal of artefacts creates missing data that may also bias downstream analysis. One major advantage of using a generative learning model within the artefact detection framework is that this model can generate new realistic synthetic observations post-training, by sampling directly from the latent representation space. For real observations that contain both artefact segments and non-artefact segments, one solution is to replace the entire observation (or only the artefact segment) with its synthetic reconstruction (for example, left column reconstructions in Figure 4.5(a)). Similarly, if an observation is partially missing, the missing segment can be set to a fixed value and viewed as artefactual, to be replaced by its synthetic reconstruction in the same manner. When an observation contains fully missing data or is fully artefactual, identifying a suitable latent representation (and similarly a viable synthetic reconstruction) is much less straightforward, since the observation contains no information about what the counterfactual, i.e. what waveform would have been had it been recorded as a 'valid' waveform in the absence of any artefact (e.g. Figure 4.5(a), bottom right reconstruction).

In practice, the task of identifying when to use a synthetic reconstruction is aided by the latent representations of artefactual observations. An observation that contained an artefact but had high probability mass within the 'aggregate variational' distribution tended to have a more realistic synthetic reconstruction, since its latent representation learnt 'valid' features of the observation. Conversely, some observations containing artefacts had a latent representation in regions of $\mathcal{Z}$ with very low probability mass, because these were unlike anything the generative model encountered during model training (Figure 4.5(b)). As the generative model did not spend any time in these regions of $\mathcal{Z}$ during training, the corresponding synthetic reconstructions were of poor fidelity (e.g. Figure 4.5(a), right column and second row). This subset of poor fidelity artefactual observations can be identified and excluded from imputation, using density-based anomaly detection methods in the latent representation space, but they do not have any obvious alternate synthetic reconstructions. One potential avenue for imputation with the excluded synthetic observations (when the latent representation is unavailable or of poor quality) is to reorder all observations in chronological order by their metadata timestamps, and to track the latent representation trajectory of successive observations (around the excluded observation) within $\mathcal{Z}$. If $\mathcal{Z}$ is well-structured, this could allow the missing latent representation to be approximated using interpolation. However, this approach has not been verified.

Justification and summary. I used an ABP waveform as a proof-of-concept test case for several reasons. Firstly, ABP is particularly artefact-prone, due to the effects of arterial flushing and patient movement. Secondly, it has universal physiological importance, especially in the extremes (hypotension and hypertension), which tend to have higher rates

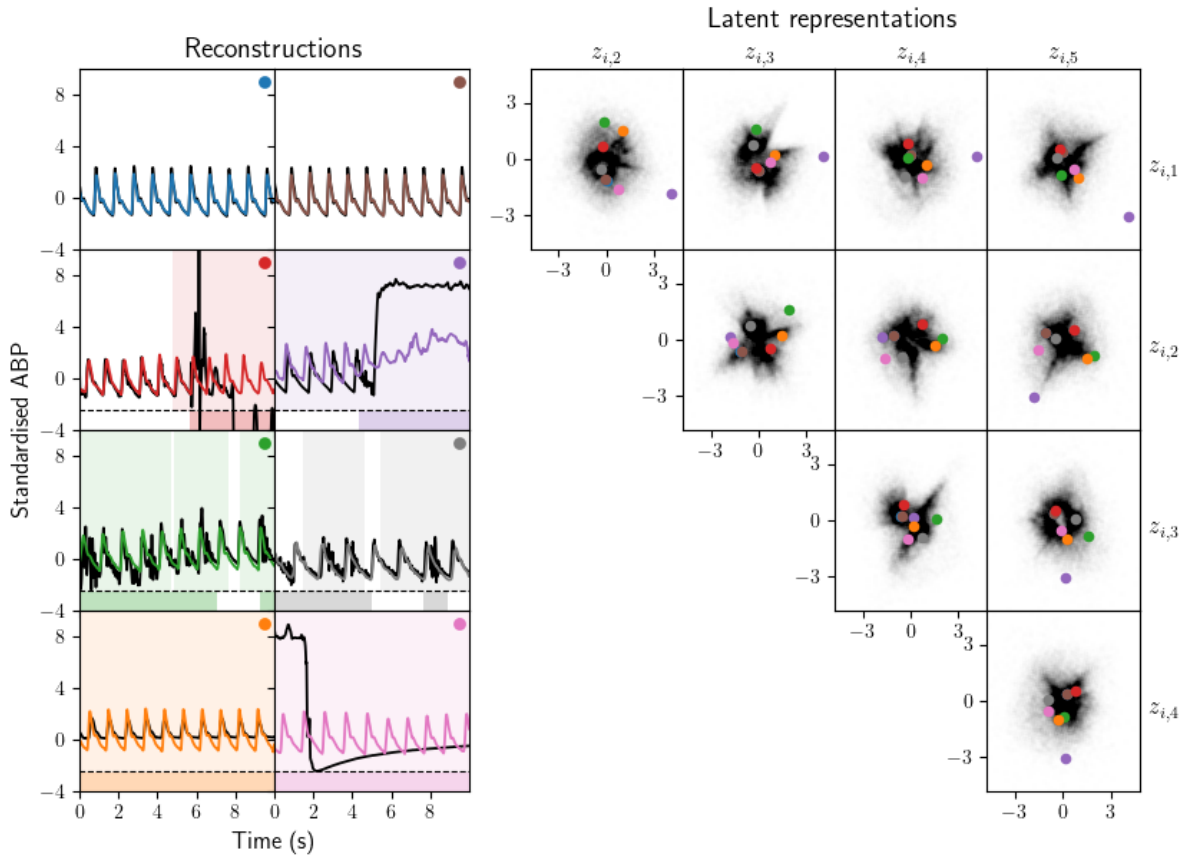

Figure 4.5: Example observations and their latent representations and synthetic reconstructions. All are from DeepClean with latent dimension 5. (a, left) The real observations (black) and their reconstructions (coloured). These example observations include noisy data (third row), attenuation (left column, fourth row) and flushing (right column, second and fourth row). Shaded regions show the DeepClean within-sample artefact detection (above the dotted line) and the 'ground-truth' annotation (below the dotted line). (b, right) Latent representations in 5-dimensional latent space $\mathcal{Z}$. Each subplot shows the distribution of a pair of latent coordinate variables, after marginalising over all others. The 'aggregate variational' distribution of the training observations is shown in black, and the latent representations of the example observations in (matching) colour.

of artefacts. Finally, ABP morphology generally has good signal-to-noise properties and consistent periodicity, particularly for patients in sinus rhythm, which made it an easier waveform to work with in proof-of-concept. The idea is that an autoencoder-like generative model can learn the structure of 'clean' waveforms, and can subsequently classify artefacts based on differences between unseen inputs and their corresponding outputs. However, this ABP waveform is perhaps too simplistic for this approach to be generalised to other noisier or less predictable waveforms, without more careful consideration about properties of the waveform and about the architectures of the deep learning networks. More flexible architectures, such as transformer networks, may be better suited to other physiological signals.

It was surprising that PCA performed poorly in reconstructing observations with high fidelity. The waveform was well-structured and periodic, and should be described well by a relatively small number of Fourier components [176]. It is possible that an optimal number of PCA components was not considered. The reason that PCA performs poorly using a classifier based on the reconstruction error is likely because the hyperplane that PCA fits to the data is linear and is sensitive to any outliers remaining in the training set. However, it did perform better when using a classifier based on the rescaled PCA scores. This is not that surprising, as the former is a distance perpendicular to the hyperplane and the latter is a distance on the hyperplane. It seems reasonable that only one of these classification statistics performed well for a linear transformation of the data. In contrast, DeepClean should find a non-linear lower-dimensional manifold to describe the data, and indeed a classifier based on the distance to this manifold performed better for DeepClean.

I divided the ABP waveform into 10s samples for training and evaluation, since observations of this length typically contain a small number of beats, and clinical experience suggested that ABP physiology should not vary grossly over this time period. This was really a middle-ground choice. It had to be a long enough time period for a generative model to learn meaningful latent representations, and a 10s observation should contain sufficient beats for this. However, the generative model should not be allowed to learn features on longer timescales, i.e. timescales relating to infrequent clinical events that describe changes in the patient state, because this may result in artefact detection framework incorrectly identifying such clinical events as artefacts. At worst, this scenario could result in delayed alerting and clinical intervention.

I considered expert manual mark-up as a 'ground-truth' artefact identification. Some artefacts were clearly identifiable because their waveform profile was so extreme, e.g. blood sampling and arterial line flushing, both of which may be variable in profile but will clearly contain un-physiological waveform excursions. However, other waveforms were not so clear-cut. Patient movement may introduce vibration, which renders the waveform unusable in the extreme, but usually only decreases the signal-to-noise ratio. Changes in

the resonant properties of the ABP transduction system, due to blood clots or bubbles, represents another difficulty. This may occur because of over-damping (mean pressure preserved) or attenuation (mean pressure not preserved) [177]. In either case, the pulse amplitude is reduced and high-frequency features are lost. Since the presence of high frequency features varies with cardiac output, it is impossible to absolutely identify these segments as valid or otherwise.

I focused little on hyperparameter optimisation and architecture choices, consistently using CNNs with only a small number of layers. This was deliberate, since I expected any gains from increasing the network size or architecture complexity would be small for this type of waveform data. Additionally, increasing the learning capacity of the decoder may mean that information is encoded within the decoder weights rather than within the latent representation. I briefly investigated relaxing distributional assumptions. I had fixed the generative distribution variance term σ_x^2 (and, in this case, $\sigma_x^2 = \beta^{-1}$), but this can instead be made into another output of the decoder (alongside the synthetic observation mean $\mu_{x,i}^g$). When the generative distribution is a multivariate Gaussian with diagonal covariance, artefacts can be identified using confidence regions, e.g. Mahalabonis distance [178], instead of MSE. However, this may hinder artefact detection, because the generative model would assign higher variance to artefact segments than to non-artefact segments, which means that any confidence region for an artefact segment would likely be much wider, at the same α-level. Introducing the generative distribution variance as a function of the decoder also introduces more decoder weights θ, which increases the flexibility of the model, but may result in overfitting, vanishing gradients during training and increased training costs.

Analysis of physiological waveform data is a key component in the treatment of critically ill patients in ICU. These waveforms are susceptible to artefacts, which must be removed before the data can be reused for clinical alerting or for clinical research purposes, including derivation of important secondary clinical parameters. Accurate artefact identification and removal often reduces bias and uncertainty in clinical assessment, while lowering the false positive rate of ICU alarms. I presented an unsupervised artefact detection framework in this chapter, which used a VAE to generate synthetic reconstructions of the real observations. This avoids costly manual annotation and offers a promising alternative to artefact detection in physiological waveform data. The DeepClean framework does not require much data preprocessing, unlike similar methods e.g. the pulse pre-segmentation in [179], and only needs basic heuristic rules to create mostly 'clean' training dataset. Including a generative model within the artefact detection framework has additional benefits, in that it can also function as an imputation method, by replacing artefact segments with synthetic observations.

CHAPTER 5

Challenges in generating synthetic medical time-series data

This chapter presents the most recent work from my PhD. In it, I examined three main pillars (privacy, fidelity and utility) of assessing state-of-the-art generative deep learning models, from the perspective of ICU time-series data. Towards the end of my PhD, I had discussed with Ari Ercole the technical challenges involved in releasing open-access large-scale anonymised ICU datasets. As well as providing new insights into physiological trajectories, synthetic data from generative modelling should (in theory) allow an alternative ethical and legal route to releasing open-access ICU data resources. To explore issues around synthetic time-series, I used a state-of-the-art generative model [180] and generalised a concept of identifiability [181], both of which were introduced by the van der Schaar group in Cambridge, but I have not collaborated with this group on the content of this chapter. In Section 5.4, I used the analysis from a recent project I was involved in, which aimed to describe sepsis epidemiology in AmsterdamUMCdb. This project was a collaboration with Chris Williams, a junior doctor at Addenbrooke's, who provided clinical expertise and wrote a forthcoming manuscript (currently under review). I developed the code, figures and tables for this project. In this chapter, I extended this analysis to synthetic data generated using AmsterdamUMCdb. Section 5.4 includes some parts of mine and Chris's manuscript, which I have rewritten in my own words. The content of this chapter is as yet unpublished.

5.1 Introduction

The primary goals of synthetic data generation are that:

- the synthetic data are similar enough to the real data that it can be used as a substitute for the real data when the latter cannot be readily shared, and

- synthetic observations are dissimilar enough from any single real observation, such that it is not possible to make meaningful inference about any individual whose data are included in the real dataset (or at least no more probable than doing so given all other observations from the real dataset, with this individual removed).

This is clearly a difficult task, as these dual goals are competing, though the former (fidelity and utility) is a global statement of all observations and the latter (privacy) is a local statement of individual observations.

I illustrate these abstract goals with a hypothetical example from medical imaging. Patients who have had a stroke will require neuroimaging to help clinicians understand the extent of the brain injury. Suppose there are research questions around possible risk factors associated with the severity of hemorrhagic and/or ischaemic strokes. The dataset could include magnetic resonance imaging (MRI) scans, and demographic and comorbidity risk factors, some of which may be obviously identifiable. For this reason, the real dataset cannot be published as it is, without some de-identification process. The data is high-dimensional, as each observation is a vector containing pixel values of the MRI, concatenated with demographic and comorbidity data. There are clearly complex interdependencies across both neighbouring (i.e. pixels next to each) and far away elements (i.e. shape of the skull) of the pixel vector. A good generative model needs to recognise and reproduce global and local features, in order to achieve sufficient fidelity i.e. it needs to produce synthetic MRIs that look 'realistic' to a clinician and it needs to capture well-established relationships within demographic and comorbidity data. Some features in an MRI may seem intuitive to humans but will not be easy to define explicitly in statistical terms. A good synthetic dataset will contain a similar number of observations as the real dataset, but without allowing re-identification of any individuals who contributed to the real data. Under sufficient privacy guarantees, e.g. use of a differentially private algorithm, it should not be possible to identify a synthetic observation as 'belonging' to any real individual, with a high confidence. The synthetic dataset should be constructed in a way that prevents external parties from, for instance, inferring the extent of a real patient's brain injury by matching any information they (the external party) already have about the patient's demographic data to synthetic observations, and consequently identifying an synthetic MRI that is almost identical to the patient's real MRI. This would be classed as information leakage or an attribute inference attack. The utility of this synthetic data could come from its capability for new knowledge generation. This cannot be directly evaluated before synthetic dataset publication, but some utility can still be checked by verifying established results, i.e. whether known risk factors are still risk factors within the synthetic dataset. Finally, one mechanism that allows development of downstream models is called TSTR (train on synthetic, test on real) [182]. In this framework, external researchers are provided access to the synthetic dataset and can develop models on the

synthetic data, before handing their final downstream model to a 'trusted data guardian' (i.e. hospital) for testing on the original real dataset, in order to validate the research findings.

5.1.1 Key contributions

This chapter is split into three, each exploring an aspect of synthetic data from the perspective of data from intensive care. The key questions and my contributions were as follows:

1. In Section 5.2, I sought to generalise identifiability, an observation-level measure of privacy, extending this from a property of a synthetic dataset to a property of the underlying generative model. In particular, I considered the following question: what is the probability under a generative 'approximate' distribution of generating a synthetic observation that 'identifies' any given real observation? I then explored how identifiability related to real and synthetic dataset sizes. I visualised the generalised identifiability with a geometric example, then evaluated this both simple (resampling with Gaussian noise) and state-of-the-art (a generative deep learning model called TimeGAN) approaches. Finally, I summarised this measure in the context of common privacy attacks.
2. I previously showed in Chapter 3 how information-theoretic measures of causal influence in bivariate physiological time-series contains latent signal about a patient's physiology. To test the fidelity of synthetic datasets from both generative models (resampling with Gaussian noise and TimeGAN), in Section 5.3 I wanted to investigate whether the mutual information and transfer entropy were preserved in each synthetic dataset. However, I faced issues relating to the observation length, which impacted both the performance of synthetic data generation and of entropy-like causal influence estimation, in contrasting directions (i.e. longer time-series meant more reliable estimation of information-theoretic measures, but also meant that there were fewer and more complex observations and so poorer generative models). I highlighted that there is still work to be done to ensure these information-theoretic measures are maintained by generative models.
3. Finally, in Section 5.4, I investigated whether a state-of-the-art generative model for time-series data (TimeGAN) could reproduce the incidence and epidemiology of sepsis within a real ICU dataset (AmsterdamUMCdb). I used the TimeGAN model to generate synthetic datasets of physiological variables involved in identifying sepsis incidence, at various levels of granularity. I then built on analysis from a collaborative research project that I am involved in, which aimed to provide a descriptive analysis of the Sepsis-3 criteria in AmsterdamUMCdb. I used the

Sepsis-3 clinical research criteria to identify cases of septic shock and sepsis without shock, for both real and synthetic datasets. Grouping patients by their sepsis status at admission, I investigated whether synthetic datasets would preserve a 'sepsis trajectory' in ICU and would reproduce summary characteristics of demographic information, physiological variables, admission categories and outcomes. In particular, I wanted to determine whether the link between sepsis (including septic shock) and ICU mortality was present in synthetic datasets. In this analysis, I highlighted glaring discrepancies between the synthetic datasets and the real dataset, as well as huge variances across repeated initialisations when training otherwise identical generative models, showing that these synthetic datasets would be of limited utility in downstream tasks relating to sepsis.

5.1.2 Mathematical definition and notation

I denote a real dataset as $\mathcal{D}_r$, which contains observations x_i^r for $i = 1, \ldots, m$, where m is the number of observations within the dataset. In the most general case, an observation x_i^r may contain both time-independent and time-varying variables, as well as timestamps at which time-varying components were recorded. For simplicity, I assume that each observation in the dataset contains the exact same set of features at each timestamp and that the timestamps t_τ (for $\tau = 1, \ldots, T$) are regularly-spaced with a fixed period between successive timestamps. The observation x_i^r can then be written in vector format as $x_i^r = (u_i, v_{i1}, \ldots, v_{iT})^T$, where the time-independent component $u_i \in \mathbb{R}^U$ and time-varying component $v_{i\tau} \in \mathbb{R}^V$ may be vector-valued themselves. The observation x_i^r has length $D = U + VT$ and is a realisation of a 'true' distribution $\mathbb{P}_r$ on sample space $\mathcal{X} \subseteq \mathbb{R}^D$. A subset $\mathcal{D}_{r,-i}$ contains the same set of observations as $\mathcal{D}_r$ but without x_i^r.

A synthetic dataset $\mathcal{D}_g$ contains synthetic observations x_j^g for $j = 1, \ldots, n$, where the size of the dataset n does not need to match that of the real dataset. The synthetic observations x_j^g contain the same variables as the real observation, and exist in the same sample space $\mathcal{X} \subseteq \mathbb{R}^D$. The goal is to learn a generative model $G(\cdot)$ that returns synthetic data observations x_j^g from an generative 'approximate' distribution $\mathbb{P}_g$, where $\mathbb{P}_g \approx \mathbb{P}_r$. As was the case for the VAE in the previous chapter, the generative model is usually a deep neural network, parameterised by weights θ. In most cases, the generative model is a latent variable model, and maps latent inputs $z_j \in \mathcal{Z}$ from a known distribution $\mathbb{P}_z$ to synthetic observations $x_j^g \in \mathcal{X}$, i.e. $G : \mathcal{Z} \times \Theta \to \mathcal{X}$, $(z_j, \theta) \mapsto x_j^g = G(z_j, \theta)$. The latent space $\mathcal{Z}$ is usually structured differently to the sample space $\mathcal{X}$, with lower dimension.

As in all previous chapters, I denote any distance metric and norm by $d(\cdot, \cdot)$ and $\|\cdot\|$ respectively, which are assumed to be the Euclidean distance and norm unless otherwise specified. I denote the indicator function for some event or condition A as $\mathbb{1}\{A\}$. Finally, I denote resampling of real observations from $\mathcal{D}_r$ using an index notation $i[j]$, i.e. $x_{i[j]}^r$ is

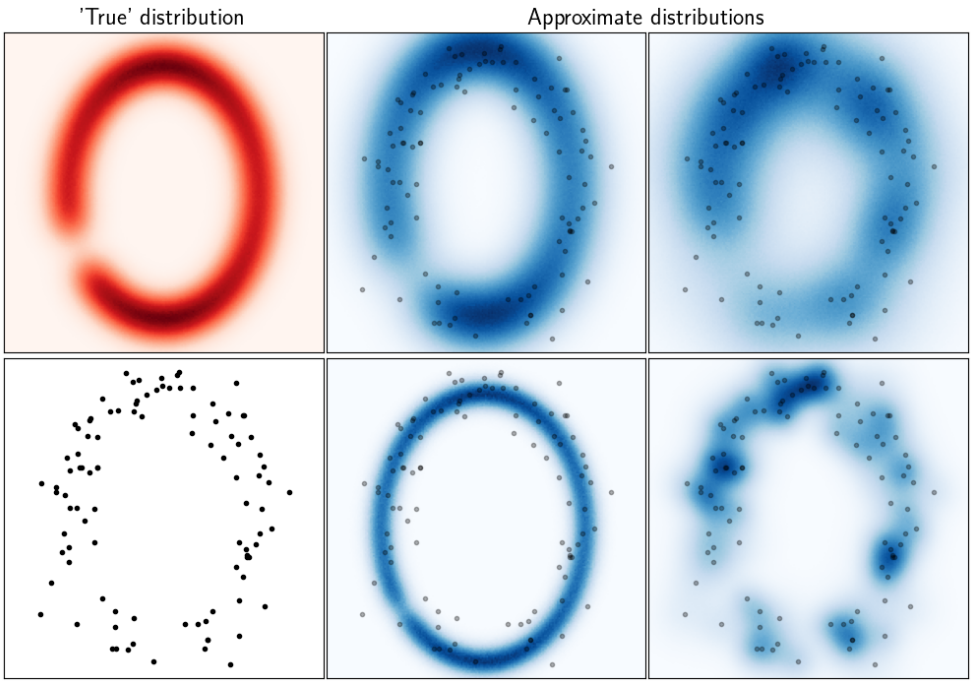

Figure 5.1: Example 2-dimensional 'true' and approximate distributions. The left column shows the underlying (unseen) 'true' distribution (top) and the real dataset (bottom). The remaining columns show several approximate distributions, alongside the real dataset. The distributions in the top row seem better than those in the bottom row, which either assign very low probability to most real observations (bottom centre) or overfit to the real observations (bottom right).

the j^{th} draw from $\mathcal{D}_r$. Sampling with replacement, the set of unique resampled indices after n draws is denoted as $I[n] = \{i[1], \ldots, i[n]\} \subseteq \{1, \ldots, m\}$.

5.1.3 Overview and related work

Approximating the true distributions. The two goals of synthetic data in Section 5.1 can be reconciled by viewing the real dataset as a probabilistic sample from an underlying 'true' distribution $\mathbb{P}_r$. The true distribution has continuous density and non-zero probability mass around any and all 'real' observations that could be observed if the data generation process was continuously repeated (i.e. for an infinite number of patients admitted to ICU). However, there are also hypothetical observations that are extremely unlikely to be observed in practice, and the true distribution $\mathbb{P}_r$ will therefore have almost-zero probability mass near these. By definition, $\mathbb{P}_r$ must have non-zero probability mass around the observations in $\mathcal{D}_r$ itself, since these observations were observed. If the generative model can similarly assign probability mass under the approximate distribution $\mathbb{P}_g$, without placing too much probability mass around the observations in $\mathcal{D}_r$, then it will be able to produce 'realistic' synthetic observations that are not 'almost-copies' of the real observations. Consequently, the balancing act for an ideal generative model involves learning a close approximation

to the true distribution without overfitting to the training data. In this case, a synthetic dataset $\mathcal{D}_g$ sampled from the approximate distribution $\mathbb{P}_g$ shares statistical properties with $\mathcal{D}_r$, without information leakage of any individual who contributed to the real dataset.

This is not an entirely straightforward task in practice, and there are several challenges to learning this approximate distribution. Firstly, the true distribution is almost always intractable. In many situations, the sample space is high-dimensional and the volume of real data is finite, which almost certainly means that real observations are sparse within the region of sample space that has non-zero probability (the support of $\mathbb{P}_r$). We assume that $\mathbb{P}_r$ has continuous density but, without infinite access to real observations, we cannot say for certain how the probability mass is distributed in regions away from the real observations, or even how concentrated the probability mass is around any particular real observation. We also assume $\mathcal{D}_r$ is an unbiased sample from the true distribution, in the sense that repeating data collection will result in another real dataset that is quantitatively similar, but in practice the data collection process is usually a noisy and biased process. Secondly, there are a number of established failure modes for generative models [183], which include overfitting or memorisation of real observations. One of the most common failures is mode collapse, where $\mathbb{P}_r$ is multimodal but the generative model only learns to generate observations from a small number of its modes (e.g. the dominant clusters in the real dataset). The ability of the model to generalise (i.e. to unseen regions of sample space that have non-zero probability under $\mathbb{P}_r$) is a desirable property, but this becomes undesirable if too much probability mass is assigned to these regions or if probability mass is erroneously assigned outside the support of $\mathbb{P}_r$ (mode invention).

Fidelity. If we managed to define a generative distribution $\mathbb{P}_g$ that successfully approximates $\mathbb{P}_r$, then sampling a synthetic dataset from $\mathbb{P}_g$ will be almost equivalent to constructing a new real dataset (i.e. taking observations from a new set of patients). In this ideal case, the true distribution and the approximate distribution are closely aligned, and so share statistical properties. Sample statistics of the real and synthetic datasets are inherited from these distributions and will therefore be similar, with the generative model achieving almost perfect fidelity. With perfect fidelity, any downstream task required from the real dataset can be performed interchangeably on the synthetic dataset, producing the same results, and insights gained from research on the synthetic data will translate to the real data (i.e. perfect fidelity implies perfect utility).

However, in practice there will be a non-zero divergence between the real and approximate distribution. An initial fidelity test usually involves visual inspection of synthetic observations by human experts, particularly in domains where humans can easily and rapidly identify whether an observation is 'realistic' or not (e.g. most image data) [184, 185]. The greater the divergence between the true and approximate distributions, the fewer

statistical properties are shared by the real and synthetic datasets. As a result, a developer has to make choices, on some level, that prioritise certain properties of the real dataset that they wish to reproduce, which is typically enforced by the training objective of the generative model. Another approach to this is conditional deep learning, e.g. [182, 186], which forces the generative model to remember labels or properties associated with each real observation, learn the relationship between property and observation, and subsequently generate synthetic observations with some desired value of that property.

Statistical properties to evaluate the fidelity include global divergence in distribution (e.g. Jensen-Shannon divergence, Wasserstein distance [187, 188]), differences in the distribution supports (precision and recall [189], density and coverage [190]), moment matching (maximum mean discrepancy [182, 191]) and domain-specific distance measures (Fréchet Inception distance for images [192]). Likelihood-based assessment of generative models may be misleading in high-dimensional settings, in which case 'realistic' observations are not sufficient or necessary for high model likelihood [193]. It is worth emphasising that there is no universal domain-agnostic notion of imperfect fidelity, and different domains will inevitably assign different relative importance to a given set of properties.

Utility. Utility measures the similarity in the image of real and synthetic datasets under some function or transformation, i.e. the similarity in distribution of outputs of that function rather than the similarity between $\mathbb{P}_g$ and $\mathbb{P}_r$ directly. In other words, utility relates to the overarching motivation or purpose behind generating the synthetic dataset. Two common reasons for generating synthetic datasets are (i) to allow external researchers to develop downstream statistical or machine learning models on realistic data without compromising on real dataset confidentiality, and (ii) to augment a biased real dataset by boosting minority phenotypes in order to improve data fairness [194–196]. Of these, the latter is often explicitly stated as the goal of the synthetic data (and so can sometimes influence the generative model), and the former is usually implicit.

The basic fundamental test of utility is whether the synthetic data can answer downstream questions, regardless of whether these answers can be considered correct, e.g. it must contain all the requisite variables at a sufficient granularity to provide an answer. Beyond this, there is no well-defined standard for evaluating the utility of synthetic data. One approach is to verify that the performance on known downstream tasks accurately reflects established results [197, 198]. This is difficult to define, because it may be unclear what downstream tasks will be required of the synthetic data in the future. The utility of the synthetic dataset for any particular downstream task can only be verified after evaluating this task on the real dataset, at which point the synthetic data appears obsolete. If the approximate distribution $\mathbb{P}_g$ does not fully capture all statistical properties and between-variable relationships of $\mathbb{P}_r$, then it follows that there are potential downstream

tasks for which the synthetic dataset will not produce answers that are faithful with respect to real dataset. Performance on downstream tasks should certainly be evaluated to some degree before the synthetic dataset is shared, but to what extent this should happen is something that does not currently have an answer. In most literature that introduces new generative modelling approaches or architectures, the utility of the new model is assessed by comparing the performance of one or two machine learning tasks, such as next-step prediction or classification [180, 182, 199]. Tasks such as this are important in many domains but are not universally applicable or useful, and I would argue instead that synthetic data utility is best evaluated on well-defined domain-specific baseline tasks that have a known ground-truth result [200, 201].

Privacy for real and synthetic data. There are several strategies to safeguarding confidential identifiers and building privacy-preserving datasets (real or synthetic). Before real medical data is made publicly available, it must first undergo preprocessing steps to preserve the anonymity of individuals within the dataset, as the raw data will almost certainly contain sensitive personal information. The standard for de-identification includes removing direct identifiers (names, social security numbers) and discretising quasi-identifiers (age, admission date, postcodes) to a level that meets k-anonymity and l-diversity requirements [202, 203]. In a dataset that satisfies both k-anonymity and l-diversity, each individual shares quasi-identifiers with at least $k-1$ other individuals and, for any sensitive attribute in the dataset, each group of individuals that share quasi-identifiers contains at least l distinct individuals for every attribute value. If the de-identification process is too strict, with too many variables removed or discretised, then the research value of the dataset may be severely limited [204]. The Amsterdam UMC intensive care database [15] uses both k-anonymity and l-diversity alongside a risk-based assessment of the likelihood of re-identification and an end-user license agreement, in order to comply with the Health Insurance Portability and Accountability Act (HIPAA) in the US and General Data Protection Regulation (GDPR) in Europe.

The taxonomy of privacy attacks against synthetic data includes membership inference attack [205, 206] and attribute inference. In a membership inference attack, the attacker is able to discriminate between observations used to train the generative model and those unseen by the model, typically as a result of model overfitting. In attribute inference, partial information from an alternate data source can help to re-identify an individual using similar synthetic observations, which can then reveal sensitive attributes associated with the individual, e.g. [204, 207]. Given suitable statistical privacy guarantees for individuals whose data is used to train the generative model, it is possible to apply a less stringent post-hoc de-identification on a synthetic dataset than on the real dataset, without compromising the confidentiality of individuals in the real dataset. As de-identification

is a lossy transformation of the data, a less stringent de-identification process does not impose as much restriction on the utility of the synthetic dataset, compared to a fully de-anonymised real dataset.

Differential privacy, identifiability and memorisation. In the literature, the dominant approach for establishing statistical privacy guarantees in dataset sharing is differential privacy [208, 209]. This states that for every pair of datasets that differ only by one observation (e.g. $\mathcal{D}_r$ and $\mathcal{D}_{r,-i}$), the probability of generating any set of synthetic observations (S) using a function $G(\cdot)$ of the first dataset ($G(\mathcal{D}_r)$) is at most a small multiplicative factor different from the probability of generating the same set S using the same function $G(\cdot)$ of the second dataset ($G(\mathcal{D}_{r,-i})$):

$$p(G(\mathcal{D}_r) \in S) \leq e^{\epsilon} p(G(\mathcal{D}_{r,-i}) \in S),\ \forall \mathcal{D}_{r,-i},\ \forall S \subseteq \mathcal{X}$$

In other words, the data for any single individual can be removed or perturbed without significantly changing the likelihood of generating any synthetic observations under the generative model, which means that their data alone will not contribute to any subsequent use of the synthetic dataset. For synthetic data generation, differential privacy is usually enforced during model training by gradient clipping and noisy gradients [210]. This only gives theoretical guarantees that are generally impractical to verify. However, the bigger issue with differentially-private algorithms is that they often return poor quality synthetic data [181]. As such, differential privacy is useful when low fidelity synthetic data is required, but currently it is generally too strong a condition for generation of high fidelity synthetic data.

Another approach to ensuring privacy is to perform preprocessing and post-processing steps to minimise disclosure of information from individuals within the training data. For example, sample-level evaluation of individual synthetic and real observations can directly assess the extent to which there has been overfitting or memorisation of the real training dataset. However, there is not yet a consistent, well-defined notion of privacy across multiple synthetic datasets derived from the same training data. I summarise two ideas in the following paragraphs, and expand on this further in Section 5.2. It is likely that differential privacy is a stronger privacy condition, at the cost of fidelity, but this is difficult to establish conclusively.

For any real dataset $\mathcal{D}_r$ and synthetic dataset $\mathcal{D}_g$, a real observation x_i^r is identifiable if there is at least one synthetic observation in $\mathcal{D}_g$ that is not sufficiently 'different enough' to x_i^r, where 'different enough' is defined in terms of the nearest-neighbouring real observation [181]. Identifiability is defined as a function of a real observation, given both datasets. This is estimated by comparing the nearest-neighbour distances between real and synthetic

observations for some suitable metric (usually Euclidean). This is formalised as:

$$f_I(x_i^r|\mathcal{D}_r, \mathcal{D}_g) = \mathbb{1}\{\min_j d(x_j^g, x_i^r) \leq r_i = \min_{k \neq i} d(x_k^r, x_i^r)\} \tag{5.1}$$

Averaging across all real observations gives a measure of the identifiability of the entire real dataset in the context of the synthetic dataset, with $\mathcal{D}_r$ classed as ϵ-identifiable from $\mathcal{D}_g$ if:

$$I(\mathcal{D}_r, \mathcal{D}_g) = \mathbb{1}\left\{\epsilon > \frac{1}{m}\sum_{i=1}^{m} f_I(x_i^r|\mathcal{D}_r, \mathcal{D}_g)\right\} \tag{5.2}$$

This definition of identifiability was proposed alongside a GAN model called ADS-GAN [181]. In this model, the authors incorporated an additional loss alongside the standard GAN model objective, to explicitly penalise identifiability. As it is computationally infeasible to calculate all possible pairwise distances between real and synthetic observations during model training, each synthetic observation was conditioned upon a real observation x_i^r in the ADS-GAN and the identifiability loss maximised the distance between this pair, without weighting by the radius r_i.

Using an alternative approach based on the posterior probability of real observations under a VAE generative model, the authors in [211] defined memorisation as "an increased probability of generating a sample that closely resembles the training data in regions of the input space where the algorithm has not seen sufficient observations to enable generalisation". They distinguished this from overfitting, which is more generally understood as the synthetic dataset containing 'almost-copies' of real observations. Evaluated over K repeated initialisations for otherwise identical models G_k, the memorisation score [211] is:

$$f_M(x_i^r) = \log \frac{1}{K}\sum_{k=1}^{K} p(x_i^r|\mathcal{D}_r, G_k) - \log \frac{1}{K}\sum_{k=1}^{K} p(x_i^r|\mathcal{D}_{r,-i}, G_k)$$

This leave-one-out memorisation score is a property of the generative model itself rather than any single synthetic dataset, but the posterior probability is difficult to compute unless an explicit density model is used. As such, this is not universally applicable and was only explored in [211] in the context of VAEs, using the evidence lower-bound approximation (Equation 4.3).

Post-hoc auditing. Synthetic datasets can be audited in order to improve sample-level fidelity and generalisation [183], independently of the underlying generative model. I argue that post-hoc auditing of synthetic observations acts similarly to acceptance-rejection algorithms used in Monte Carlo sampling (I am not aware of any previous suggestion of a link between these ideas). With the approximate distribution $\mathbb{P}_g$ viewed as a Monte

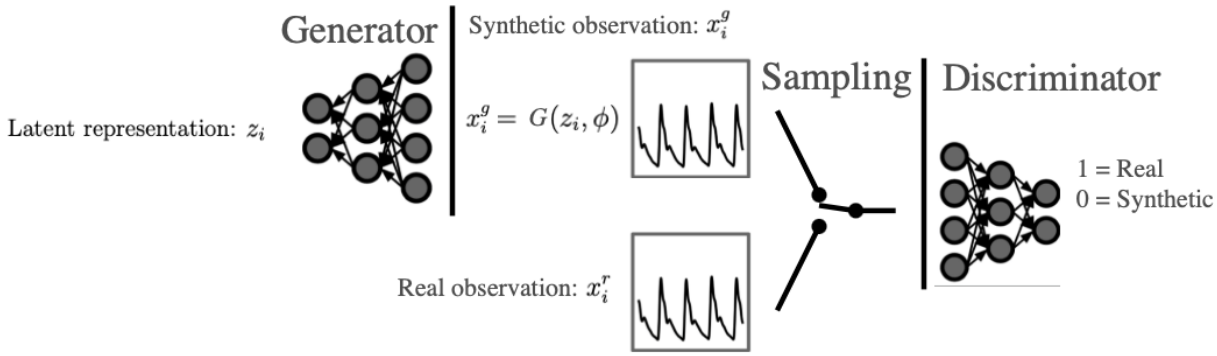

Figure 5.2: Generative adversarial network architecture. The basic GAN has two networks, a generator and a discriminator. The generator seeks to produce synthetic observations that are indistinguishable from real observations, and the discriminator is given either a synthetic or a real observation and must try to distinguish whether it is synthetic or real.

Carlo proposal distribution, observations can be repeatedly drawn from $\mathbb{P}_g$ and rejected if they fail to meet some pre-defined fidelity or privacy criteria, implicitly defining a new target distribution that has stricter fidelity or privacy guarantees. Similarly, the real training dataset can be audited after initial generative model training, in order to remove observations that are deemed to have high risk of information leakage, though this may require the model to be iteratively retrained at high computational cost.

Models: resampling with noise. One of the simplest strategies for generating new synthetic datasets is to resample observations from the real dataset and add random noise from some known distribution. This method performed well against generative deep learning models, with respect to fidelity metrics [183]. However, it can be argued that this process is not truly 'generative', i.e. it is incapable of producing novel 'realistic' observations. Resampling techniques such as the bootstrap [212] are easy to implement and can preserve the statistical properties of the data or correct for imbalanced datasets. Provided the noise has small enough amplitude that it does not destroy within-observation or between-observation dependencies, the synthetic dataset will have excellent fidelity compared to the true distribution. However, unless the noise is weighted appropriately, this approach will result in extreme overfitting to real observations. This is all true of noisy resampling from any known distribution. In the following sections, I used resampling with Gaussian noise (RwGN) to produce synthetic observations x_j^g, where $z_j \sim N(0, I)$ and σ^2 was fixed:

$$x_j^g = x_{i[j]}^r + \sigma^2 z_j \tag{5.3}$$

Models: TimeGAN. Many generative models for time-series and tabular data, e.g. [181, 182, 209, 210, 213], are based on a type of deep learning architecture called gener-

ative adversarial networks (GANs) [185, 214]. In a GAN, a generator model is trained concurrently with a second neural network that discriminates between real and synthetic observations (Figure 5.2). This discriminator model acts in direct competition to the generator during training, forcing the latter to create increasingly 'realistic' synthetic observations. One state-of-the-art model from recent years that was designed to generate synthetic time-series is the TimeGAN model [180]. This augments the generator-discriminator structure of a vanilla GAN with additional embedding and recovery neural networks that aid the model in learning local autoregressive dynamics alongside the global distribution $\mathbb{P}_r$ (see Appendix D for more details about the architecture). TimeGAN was shown to demonstrate strong performance on various benchmark tasks compared to similar methods [180]. Most TimeGAN components are recurrent neural networks (RNNs), which maintain an internal 'memory' of previous states allowing information to propagate from past states to future states. While TimeGAN was designed to have state-of-the-art performance in terms of fidelity and utility, it contains no mechanisms explicitly aimed at improving privacy-preservation, unlike other GAN models e.g. [181, 210, 215].

Data. As in previous chapters, I used data from AmsterdamUMCdb in this chapter. In Sections 5.2 and 5.3, I used (standardised) bivariate time-series, consisting of temperature and heart rate measurements recorded every minute. In Section 5.4, I used a much wider set of demographic, physiological and outcome variables, at two levels of temporal granularity. This is described in more detail within each section.

5.2 Privacy: extending identifiability

The first area of synthetic data that I investigated in the context of medical time-series was observation-level privacy. In particular, I wanted to extend identifiability (Equation 5.1) from a property of a single synthetic dataset to a property of the underlying generative model. Closely related to this, I sought to establish the maximum synthetic dataset size that could be published without exceeding a suitable pre-defined identifiability level. This is an important concern when the data owner wishes to release multiple synthetic datasets based on the same real dataset, e.g. providing separate synthetic datasets to different external research groups. Additionally, considering identifiability as a property of the underlying generative model is generally more useful during model development, when changes to the model architecture, model objective or real dataset can be made in order to satisfy privacy guarantees. Finally, in the previous definition of identifiability (Equation 5.1), there was no distinction between a real observation that is identified by just one synthetic observation and a real observation that is identified by many synthetic observations. It is reasonable to assume that a privacy breach is more likely in the latter

case, because the real observation is similar to a larger number of synthetic observations.

I aimed to evaluate the probability of generating a synthetic observation under $\mathbb{P}_g$, such that it 'identifies' a given real observation. To do this, I first considered identifiability as a function of both synthetic and real observations:

$$f_I(x_i^r, x_j^g | \mathcal{D}_r, \mathcal{D}_g) = \mathbb{1}\{d(x_j^g, x_i^r) \leq r_i = \min_{k \neq i} d(x_k^r, x_i^r)\}$$

In the previous definition, x_i^r was *identifiable* if this condition was satisfied for any x_j^g, where x_i^r is *identified* by x_j^g and x_j^g *identifies* x_i^r. In addition to this, I also considered x_j^g to be *identifying* if $f_I(x_i^r, x_j^g) = 1$ for any x_i^r. Both the set of real observations that are identifiable and the set of synthetic observations that are identifying can be made clearer by viewing this problem in terms of the geometry of the sample space.

5.2.1 Geometric interpretation

A real observation x_i^r is identifiable if there is any synthetic observation closer to it than the nearest-neighbouring real observation is. This condition is equivalent to the occurrence of any synthetic observation within a ball with centre x_i^r and radius $r_i = \min_{k \neq i} d(x_i^r, x_k^r)$. Using the Euclidean distance, this ball is a hypersphere. Denoting the set of points contained within (and on the surface of) this ball as $\mathcal{B}_i$, the condition that x_i^r is identifiable is equivalent to the following definition, where $\cup$ denotes set union:

$$f_I(x_i^r | \mathcal{D}_r, \mathcal{D}_g) = \mathbb{1}\{\exists\, x_j^g : x_j^g \in \mathcal{B}_i\} = \cup_j \mathbb{1}\{x_j^g \in \mathcal{B}_i\}$$

Contrasting this, a synthetic observation x_j^g identifies at least one real observation if it belongs to the union of every ball $\mathcal{B}_i$, i.e.

$$f_I(x_j^g | \mathcal{D}_r, \mathcal{D}_g) = \cup_i \mathbb{1}\{x_j^g \in \mathcal{B}_i\} = \mathbb{1}\{x_j^g \in \cup_i \mathcal{B}_i\} = \mathbb{1}\{\exists\, x_i^r : d(x_j^g, x_i^r) \leq \min_{k \neq i} d(x_k^r, x_i^r)\}$$

I then argued that the identifiability should be clearly associated with the size of both real and synthetic datasets (statements which seem reasonably intuitive but I have not sought to prove or disprove). Firstly, if the number of real observations increases, then the size of the balls $\mathcal{B}_i$ should decrease on average and so each individual real observation should become less identifiable. Secondly, for a fixed real dataset, the more synthetic observations are generated, the more likely it is that at least one synthetic observation identifies any given real observation. In this case, each ball $\mathcal{B}_i$ is fixed, since the real dataset is fixed. Provided the generative distribution $\mathbb{P}_g$ assigns non-zero probability mass over every $\mathcal{B}_i$, then the likelihood of observing at least one synthetic observation lies within $\mathcal{B}_i$ increases, as more and more synthetic observations are drawn from $\mathbb{P}_g$. This suggests there is a limiting synthetic dataset size before every real observation is identifiable in expectation.

I illustrated these ideas with a toy example in Figure 5.3. The real observations in this example were two-dimensional points, with (coloured) balls $\mathcal{B}_i$ around each real observation x_i^r. In this figure, the top row contains an example dataset of 5 observations and the bottom row contains another example dataset of 100 observations. When the number of real observations was increased, the regions $\mathcal{B}_i$ and their union decreased in area. Suppose a naïve generative model samples uniformly from the grey shaded parallelogram, defined as a parallelogram of minimum area that covers at least 80% of the real observations. This has sides parallel to the y-axis and the simple linear regression line of best fit. This is a contrived and inadequate model choice but, in using a uniform distribution, it allowed a direct link between the area of intersecting shapes and the probability of generating identifying synthetic observations. In the middle column, one real observation was singled out. Any synthetic observation that exists in the intersection of the parallelogram and the ball would identify this specific real observation. The probability p that a synthetic observation randomly sampled from $\mathbb{P}_g$ identifies this real observation is equal the area of this intersection as a fraction of the area of the parallelogram. For a synthetic dataset of size n, the probability that at least one synthetic observation identifies this real observation is equal to $1 - (1 - p)^n$, which tends to 1 as the synthetic dataset increases in size. This is true for any real observation in which the accompanying ball has at least some area overlap with the parallelogram. When the number of real observations was increased (bottom row), the probability that a single synthetic observation identified this a specific real observation, or was identifying for any real observation, decreased.

By design, there are some real observations in this example that will never be identified, since there is no intersection between the corresponding ball and the shaded parallelogram. In practice, it is generally unlikely that the distribution support of $\mathbb{P}_g$ and $\mathbb{P}_r$ differ by much, which means there is unlikely to be exactly zero probability of a given real observation being identifiable for infinite synthetic observations. However, it is also not unrealistic for regions of extremely low probability mass to be significantly different (i.e. the probability of some region S under $\mathbb{P}_g$ and under $\mathbb{P}_r$ may be many orders of magnitude apart), in which case the probability that a specific real observation is identifiable could be infinitesimally small. This is the case under some common failure modes for generative models, such as mode collapse.

5.2.2 p-Identifiability

Following this, I defined a generalisation of identifiability, as the probability of identifying a real observation under the generative model:

Definition 5.1. The **p-identifiability** of a real observation x_i^r is the probability, under the approximate distribution $\mathbb{P}_g$ of a generative model $G(\cdot)$, of generating a synthetic

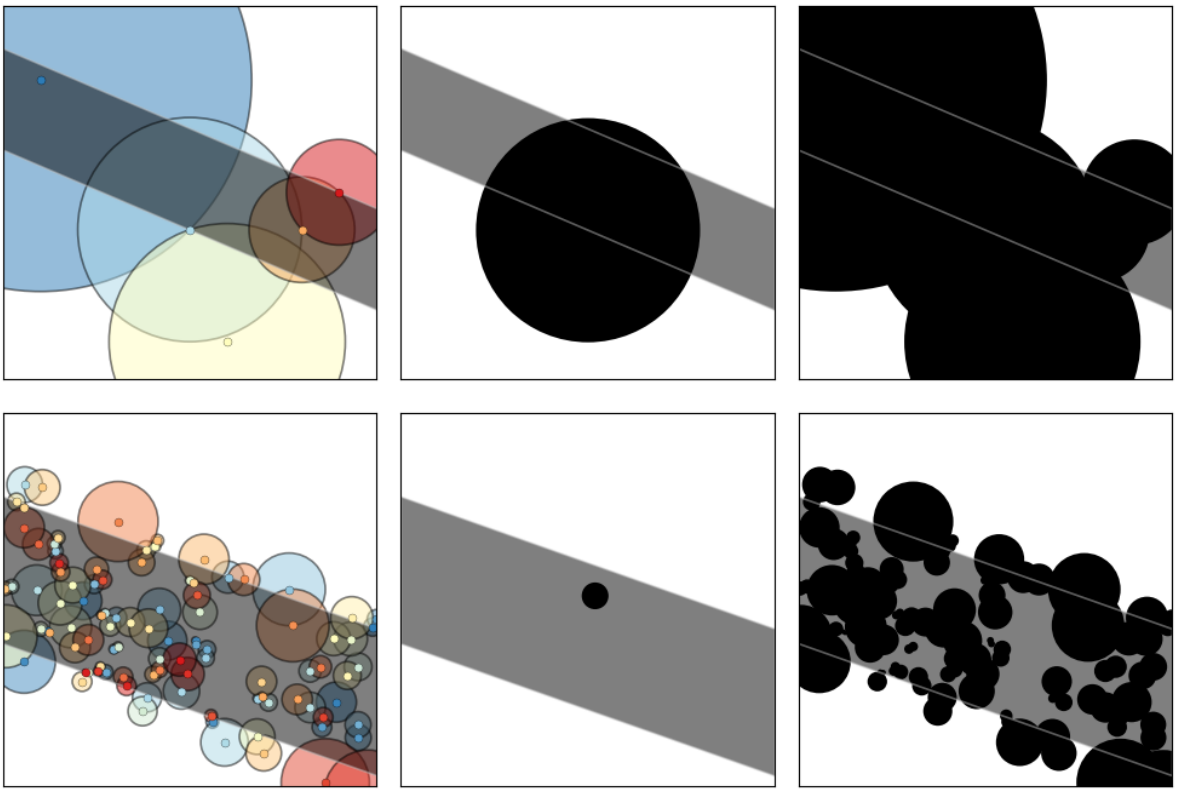

Figure 5.3: Identifiability as the number of real observations increases, with two-dimensional toy data. Each real observation has a ball around it, indicating the region in which it is identifiable by synthetic observations. The top row shows a real dataset with 5 observations and the bottom row shows a real dataset with 100 observations. In each case, I defined a simple generative model, which samples uniformly from the grey shaded parallelogram. The intersection of this parallelogram and any given ball indicates a region in which a synthetic observation is possible and identifying. The larger the intersection between parallelogram and ball, the greater the identifiability of the corresponding real observation (middle column). The probability that no real observations are identified by a given synthetic observation increases when the non-intersecting grey region is smaller (right column).

observation x_j^g that is closer to x_i^r than the distance between x_i^r and its nearest neighbour in the real dataset $\mathcal{D}_r$:

$$\begin{aligned} f_p(x_i^r|\mathcal{D}_r, G(\cdot)) &= p\big(x_j^g \sim \mathbb{P}_g \mid d(x_j^g, x_i^r) \leq \min_{k \neq i} d(x_k^r, x_i^r)\big) \\ &= p(x_j^g \sim \mathbb{P}_g,\ x_j^g \in \mathcal{B}_i) = p_i \end{aligned} \tag{5.4}$$

The most straightforward approach to estimating p-identifiability is using an empirical Monte Carlo integration (Equation 3.8), i.e. as the expectation $\mathbb{E}_g$ with respect to $\mathbb{P}_g$. For $n \to \infty$ (or at least, for $n \gg m$):

$$f_p(x_i^r|\mathcal{D}_r, G(\cdot)) = \mathbb{E}_g[\mathbb{1}\{x \in \mathcal{B}_i\}] = \frac{1}{n}\sum_{j=1}^{n} \mathbb{1}\{x_j^g \in \mathcal{B}_i\} \tag{5.5}$$

The process of generating new synthetic observations involves sampling random inputs z_j from a known latent distribution P_z, where x_j^g is a deterministic function of z_j, i.e. $x_j^g = G(z_j)$. This means that p-identifiability can be equivalently defined in terms of the distribution P_z:

$$f_p(x_i^r|\mathcal{D}_r, G(\cdot)) = \mathbb{P}_z(z_j \sim \mathbb{P}_z \mid (d(G(z_j), x_i^r) \leq \min_{k \neq i} d(x_k^r, x_i^r))) = p(z_j \sim \mathbb{P}_z, G(z_j) \in \mathcal{B}_i)$$

Instead of estimating p-identifiability empirically, an alternative approach involves optimisation over the random input $z_j \in \mathcal{Z}$. With the generative model now fixed, this minimises the distance between the corresponding generated output and a given real observation:

$$z_i^* = \underset{z_j \in \mathcal{Z}}{\operatorname{argmin}}\, d(G(z_j), x_i^r),\ \ \epsilon_i = d(G(z_i^*), x_i^r)$$

Optimisation algorithms are typically terminated upon reaching some threshold. Setting individual thresholds equal to the radii r_i means that the corresponding synthetic observation $G(z_i^*)$ lies on or just inside the surface of the ball $\mathcal{B}_i$. For a VAE, this z_i^* should be close to the latent representation (the output of the encoder). However, latent optimisation depends on the form of $\mathcal{Z}$ and $\mathbb{P}_z$. For some choices of generative model, it may be straightforward to calculate, but unfortunately it was not useful for the TimeGAN model, whose latent space is a high-dimensional joint uniform distribution.

The p-identifiability scores of different real observations are not necessarily independent, particularly if they are adjacent. If there is significant overlap between two or more identifying balls, then a single synthetic observation could identify multiple real observations simultaneously. This means that, using the p-identifiability, it is difficult to define the maximal size of synthetic dataset before the entire real dataset has ϵ-identifiability equal to 1 in expectation (i.e. the analogue of Equation 5.2). However, for any given real

observation x_i^r, the expected number of synthetic observations that are drawn before x_i^r is identified follows a geometric distribution with parameter $p = p_i$, which has mean $(1 - p_i)/p_i$.

5.2.3 Results and discussion

I evaluated the p-identifiability (Equation 5.4) for a bivariate time-series real dataset containing temperature and heart rate measurements, with corresponding synthetic datasets from TimeGAN and from resampling with Gaussian noise (RwGN). In each case, I estimated p-identifiability empirically using Equation 5.5 with $n = 100m$ synthetic observations, where m was the size of the real dataset. I implemented TimeGAN using the default model architecture (3 layers with gated recurrent units and hidden dimension of 24 for each component network) and default hyperparameters (including training batch size of 128), which had 48099 trainable parameters [180].

Data. Temperature and heart rate are routinely and constantly recorded in ICUs. I constructed a dataset containing minute-by-minute recordings of both variables from the first 24hr of ICU admission. 23080 of the 23106 patients in the database had at least some temperature or heart rate measurements during this period, but not all had recordings at the required frequency. For each patient, I calculated the difference in successive timestamps for temperature and heart rate separately, and then selected only patients where at least 10% of the timestamp differences were equal to or less than one minute, leaving an initial subset of 2011 patients. I split the time-series for each patient into hourly segments so that each patient contributed up to 24 observations. For both variables separately, if there was more than one measurement recorded within the same minute, I took the median value of these. Temperature was measured to 1 decimal place and heart rate to the nearest integer. I did not want this measurement precision to unduly influence any results or findings, so I added uniform random noise between $-x$ and $+x$, where x was 0.05 for temperature and 0.5 for heart rate. I defined an invalid measurement as below 33 or above 42 for temperature, and below 50 or above 150 for heart rate. Any observation that had invalid or missing values were discarded, leaving 25216 observations from 1751 patients (up to 24 measurements per patient). I labelled this real dataset $\mathcal{D}_0$.

I defined training and test datasets by splitting the real dataset according to the patient pseudo-identifier, with an 80-20% split. Since the number of observations per patient was not consistent, the proportion of observations in each set did not exactly match the proportion of patients. The training dataset, labelled $\mathcal{D}_1$, contained 19996 observations from 1400 patients. To investigate the effect of real dataset size, I created two further training datasets, $\mathcal{D}_2$ and $\mathcal{D}_3$, nested within $\mathcal{D}_1$, i.e. $\mathcal{D}_3 \subset \mathcal{D}_2 \subset \mathcal{D}_1 \subset \mathcal{D}_0$. $\mathcal{D}_2$ contained 10% of patients included in $\mathcal{D}_1$, with 2017 observations from 140 patients, and $\mathcal{D}_3$ contained

10% of patients included in $\mathcal{D}_2$, with 192 observations from 14 patients. Finally, I min-max scaled both temperature and heart rate separately, within each real dataset.

Resampling with Gaussian noise. The single free parameter in this model (Equation 5.3) was the variance σ^2, which controls both the fidelity and privacy of the synthetic dataset. If σ^2 is too small, then the distance between the synthetic observation x_j^g and the real observation $x_{i[j]}^r$ will be close to 0, and every real observation indexed by the set $I[n]$ will be identifiable, or equivalently:

$$\forall i[j] \in I[n], \;\; \min_l d(x_{i[j]}^r, x_l^g) \leq d(x_{i[j]}^r, x_j^g) < r_i = \min_k d(x_{i[j]}^r, x_k^r)$$

Conversely, if σ^2 is too large, the Gaussian term will dominate, synthetic observations will be of poor quality and the synthetic dataset will have poor fidelity. I decided to set the value of σ^2 such that the median distance between x_j^g and $x_{i[j]}^r$ was approximately equal to the median radius $r_i = \min_{k \neq i} d(x_i^r, x_k^r)$ of the full real dataset $\mathcal{D}_0$. Using the Euclidean distance, $d(x_{i[j]}^r, x_j^g)/\sigma^2 = ||z_j||$ follows a chi distribution, with D degrees of freedom (where $z_j \in \mathbb{R}^D$). The median value of the chi distribution, $m(||z_j||)$, is approximately equal to $\sqrt{D - 2/3}$. With $D = 120$ and the median radius $m(r_i) = 0.202$, I set the value $\sigma^2 = 0.02$. In practice, this meant that the two medians weren't completely equal, as $m(\sigma^2||z_j||) = 0.218$ and only 43% of the radii were greater than this, instead of 50%. Moments of chi distribution give the following:

$$\begin{aligned}
\mathbb{E}[d(x_{i[j]}^r, x_j^g)] = \mu &= \frac{\sigma^2\sqrt{2}\;\Gamma(\frac{T+1}{2})}{\Gamma(\frac{T}{2})} \approx \sigma^2\sqrt{T - 1/2} \\
\text{var}(d(x_{i[j]}^r, x_j^g)) &= T\sigma^4 - \mu^2 \approx 0.5\sigma^4
\end{aligned}$$

The distribution of $d(x_{i[j]}^r, x_j^g)$ was therefore narrow and almost symmetric, with 95% probability mass between 0.192 and 0.247 (inverse CDF values at $p = 0.025$ and $p = 0.975$ respectively). However, the support of the chi distribution is $(0, \infty)$, which means that there was non-zero probability of the event $\mathbb{1}\{r_i < \sigma^2||z_j||\}$ for every $r_i > 0$ (though this probability may be infinitesimally small for $0 < r_i \ll 0.192$).

Next, sampling with replacement (independently and with equal probability of selection) was performed on the indices denoting real observations. In a synthetic dataset of size n, the probability that any given index $i \in \{1, \ldots, m\}$ was sampled at least once is:

$$p(i \in I[n]) = 1 - (1 - 1/m)^n$$

For a single synthetic dataset equal in size to the real dataset, $p(i \in I[n]) \approx 1 - 1/e \approx 0.632$. In the context of p-identifiability, $n \to \infty$ with m fixed, so $p(i \in I[n]) \to 1$. In practice, I

estimated p-identifiability with $n = 100m$, in which case $p(i \in I[n]) \approx 1 - 1/e^{100} \approx 1$.

Putting both parts of this together (resampling and Gaussian noise) gives some theoretical insight into the p-identifiability for this choice of synthetic dataset. Firstly, r_i are a function of the real dataset and should become smaller as the size of the real dataset increases (as observed in Figure 5.3). I used $\mathcal{D}_0$ to define a consistent, fixed value of σ^2, but each of the training datasets ($\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_3$) were smaller in size than this one. This meant that the proportion satisfying $\mathbb{1}\{r_i > 0.218\}$ increased as the size of the training dataset decreased. Under this model, there were two possible reasons for a real observation x_i^r to be identified by some synthetic observation x_j^g, (i) this real observation was selected in the resampling ($i = i[j]$) and the corresponding Gaussian noise was less than the nearest neighbour distance ($r_i < \sigma^2 ||z_j||$) or (ii) another, sufficiently close, real observation (such as its nearest neighbour within the real dataset) was selected in resampling and by chance the addition of Gaussian noise brought the synthetic observation within the ball $\mathcal{B}_i$. For this real observation x_i^r, the p-identifiability is greater than or equal to the former, which equals the probability $1/m$ of selecting the index $i = i[j]$ on the j^{th} draw multiplied by the scaled chi distribution CDF evaluated at r_i. This lower bound can be observed in the top right panel of Figure 5.4, where almost all real observations with small values of r_i had essentially zero p-identifiability and the most real observations with large values of r_i had p-identifiability approximately equal to $1/m$.

p-identifiability and nearest neighbour distance. The distribution of p-identifiability scores differs greatly between TimeGAN and RwGN for the real dataset $\mathcal{D}_1$, which was the largest dataset both models were trained on (Figure 5.4). For TimeGAN, most real observations had essentially zero p-identifiability, regardless of their radius r_i, and remained unidentified by any synthetic observations within a synthetic dataset of size $100m$ (top left). Because they are so distinct, real outlier observations (real observations that are far apart from their nearest neighbour) were typically more vulnerable to re-identification in RwGN. This was not the case for TimeGAN, where many outliers were not identified by any synthetic observations, which suggests that the approximate distribution $\mathbb{P}_g$ for TimeGAN assigned little or no probability mass near these outlier observations. However, the distribution of p_i (i.e. y-values of the top row of Figure 5.4) was heavily skewed for TimeGAN with a much longer tail compared to RwGN, which had a bimodal empirical distribution (top right). As a result, the mean p-identifiability was over 5 times greater for TimeGAN than for RwGN (Table 5.1). In the second and third rows of Figure 5.4, I discretised the p-identifiability values on a logarithmic scale to highlight differences in p-identifiability between both models, for increasing radius r_i. As expected, the p-identifiability p_i was clearly related to r_i for RwGN, with p_i essentially 0 for small r_i and a steep rise in p_i as r_i increased above the median $m(r_i)$. This was in contrast to

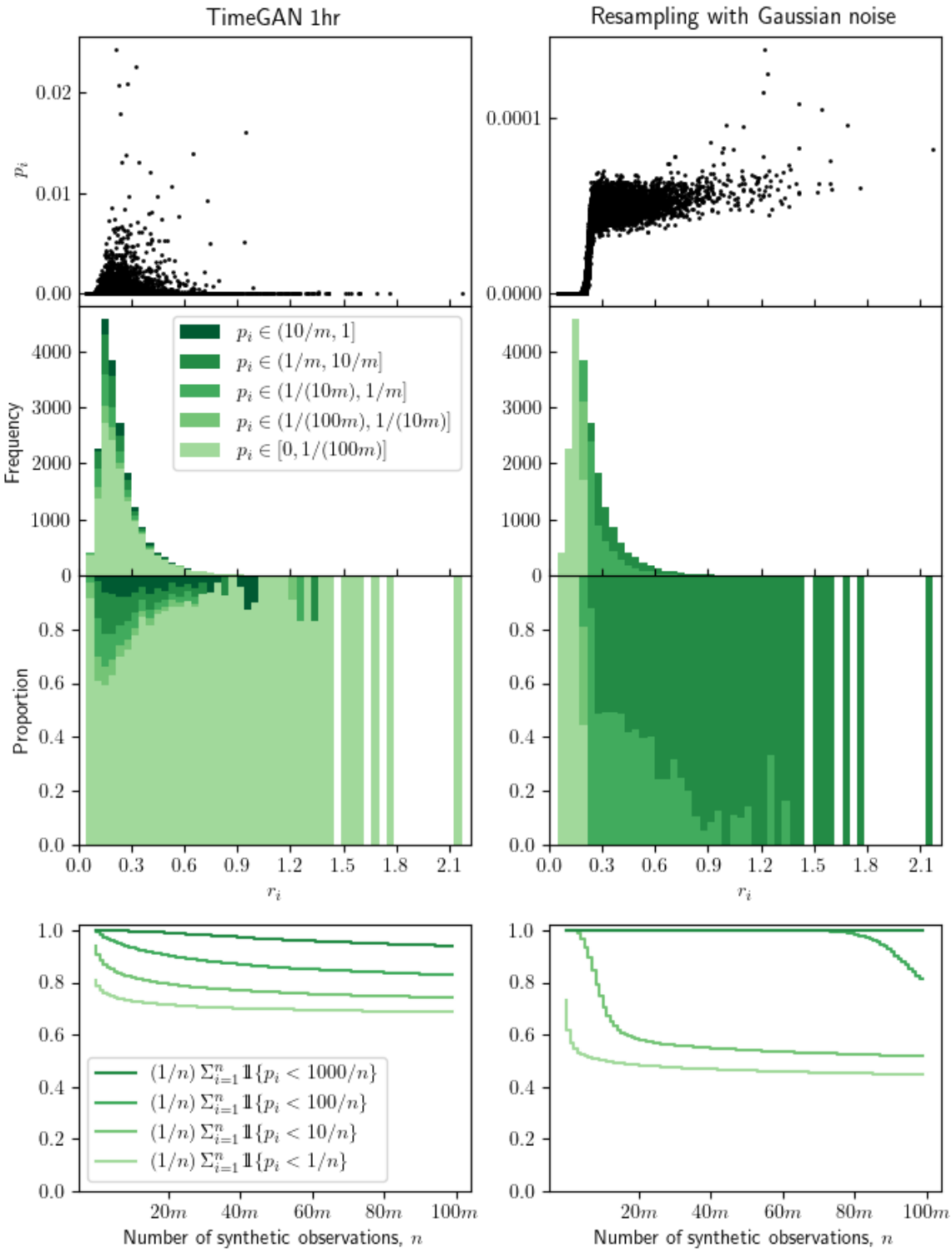

Figure 5.4: The relationship between the p-identifiability scores and the nearest neighbour radius. This used the temperature and heart rate time-series real dataset $\mathcal{D}_1$, and both TimeGAN model and resampling with Gaussian noise (RwGN). The p-identifiability was estimated empirically using $100m$ synthetic observations, where $m = 19996$ was the size of $\mathcal{D}_1$. For TimeGAN, most real observations had $p_i < 1/(100m)$. As well as scatter plots (top row), the p-identifiability scores were split into five categories linked to the number of times each real observation was identified. The second row shows stacked histograms against the radius and the third row shows the proportion of each of these categories within every stacked histogram bar. The final row shows the proportion of p-identifiability scores below certain thresholds (which match the previous categories when $n = 100m$), as the amount of synthetic data increased.

TimeGAN, where the histogram bin of r_i that had the highest proportion of identifiable real observations was for values of r_i less than the median $m(r_i)$ (third row, left).

Real and synthetic dataset size. In the final row of Figure 5.4, I calculated the proportion of real observations that were identified by synthetic observations fewer than x times (for $x \in \{1, 10, 100, 1000\}$), as the size of the synthetic dataset was increased from $n = m$ to $100m$ (where $m = 19996$ was the size of $\mathcal{D}_1$). These were decreasing functions of the synthetic dataset size n, which offered empirical evidence for my observations in Section 5.2.1. Assuming the support of the generative distribution $\mathbb{P}_g$ is contained within the support of $\mathbb{P}_r$ in every case, then each curve will eventually decrease to 0.

Conversely, a generative model trained on an insufficient quantity of real observations m is generally more prone to overfitting, so I also decreased the size of the real dataset to observe the relationship between m and p_i (Table 5.1). Two things occurred here, as expected: the mean p-identifiability values increased and the proportion of unidentified real observations (within a synthetic dataset of size $100m$) decreased. This suggested that increasing the number of real training observations does help to reduce their identifiability. In particular, almost all of the real observations for the smallest real dataset $\mathcal{D}_3$ were identified by at least one synthetic observation in RwGN.

I also estimated the maximum synthetic dataset size that could be generated and published, before an ϵ-identifiability threshold of 0.1 was reached. To do this, I set a counter on the proportion of identified real observations, as an increasing number of synthetic observations were generated. When this ϵ threshold was reached, I recorded the number of synthetic observations that had been generated, then reset and restarted the counter. Under this ϵ-identifiability condition, no generative model was capable of releasing a synthetic dataset that was at least as large as the corresponding real dataset (Table 5.1). The state-of-the-art TimeGAN model performed particularly poorly here, which may seem surprising given that a high proportion of real observations remained unidentified. However, it met the ϵ threshold in fewer synthetic observations because the mean p_i value was still much higher for TimeGAN than for RwGN, and the empirical distribution tail was much longer.

Comparing training data and unseen data. High p-identifiability for a real observation x_i^r suggests that the generative model has memorised the observation and that synthetic observations similar to this have a higher probability under $\mathbb{P}_g$ than they would if this real observation was not included in model training. Furthermore, a real observation with high p-identifiability may be vulnerable to attribute disclosure, as correspondingly similar synthetic observations could be used to infer other attributes of the individual (which is unlikely to be a pressing concern for this particular time-series data). However,

Model	Dataset	Processing	$\frac{1}{n}\sum_i p_i$	$\frac{1}{n}\sum_i \mathbb{1}\{p_i < 1/n\}$	$n_{0.1}$
TimeGAN	$\mathcal{D}_1$	None	1.23e-4 (5.5e-8)	0.688 (0.001)	880.3 (275.9)
TimeGAN	$\mathcal{D}_2$	None	7.26e-4 (1.2e-6)	0.628 (0.002)	150.6 (82.0)
TimeGAN	$\mathcal{D}_3$	None	8.42e-3 (3.4e-5)	0.255 (0.008)	18.4 (11.5)
RwGN	$\mathcal{D}_1$	None	2.10e-5 (1.1e-8)	0.447 (0.001)	3859.6 (1389.0)
RwGN	$\mathcal{D}_2$	None	4.21e-5 (5.4e-7)	0.272 (0.005)	297.7 (113.6)
RwGN	$\mathcal{D}_3$	None	4.56e-5 (1.8e-7)	0.001 (0.001)	21.6 (7.1)
TimeGAN	$\mathcal{D}_1$	Rounding	4.30e-6 (1.0e-8)	0.948 (0.000)	5897.5 (4208.7)
RwGN	$\mathcal{D}_1$	Rounding	2.10e-5 (1.8e-7)	0.459 (0.013)	3713.5 (1625.0)

Table 5.1: Summary of p-identifiability for both generative model types and varying training set sizes. This was calculated for both generative model types, TimeGAN model and resampling with Gaussian noise (RwGN). Where 'rounding' was applied, both real and synthetic datasets were rounded to match the same precision level as the original unprocessed time-series. The table shows the mean p-identifiability, the proportion of observations that remained unidentified within $100m$ synthetic observations, and minimum number of synthetic observations $n_{0.1}$ that could be generated before a proportion >10% of the real observations were identified. $\mathcal{D}_1$ had 19996 real observations, $\mathcal{D}_2$ had 2017 and $\mathcal{D}_3$ had 192. Each column contains the mean and standard deviation of empirical estimates. For the first two columns, this was computed using a jackknife procedure. The latter was estimated by iterating through the entire synthetic dataset, counting the number of synthetic observations before this threshold was exceeded.

a high p-identifiability score does not necessarily indicate that the real observation was used to train the generative model.

In a membership inference attack, the attacker's goal is to determine whether each real observation was used during generative model training or not. The p-identifiability score is only relevant to this if there is a significant difference in the empirical distribution of p_i values between real observations in the training dataset and real observations in an unseen dataset. To evaluate whether this was the case, I estimated p-identifiability scores with respect to the full dataset $\mathcal{D}_0$, but where the generative model was trained only a subset of the data (one of $\mathcal{D}_1$, $\mathcal{D}_2$ or $\mathcal{D}_3$). As the radius r_i depends on the set of real observations (in this case $\mathcal{D}_0$), these p_i values were now slightly different from before. In Table 5.2, I performed a non-parametric approximate permutation test, using the two-sample Kolmogorov-Smirnov (KS) statistic, to evaluate the statistical significance in the difference between empirical p-identifiability distributions of the training dataset and the unseen observations. Although there are known critical values for the KS statistic, the empirical distributions were heavily skewed, so it was unclear whether these critical values would be appropriate. Instead, I repeatedly pooled and permuted the p-identifiability scores between training and unseen groups before re-calculating the KS statistic, in an approximate permutation test. The p-value reported in Table 5.2 was the fraction of permutations for which the permuted-group KS statistic was higher than the original KS

Model	Training dataset	Processing	KS statistic (p-value)	ROC AUC
TimeGAN	$\mathcal{D}_1$	None	0.0316 (<0.001)	0.484
TimeGAN	$\mathcal{D}_2$	None	0.0356 (<0.001)	0.517
TimeGAN	$\mathcal{D}_3$	None	0.5634 (<0.001)	0.787
RwGN	$\mathcal{D}_1$	None	0.4728 (<0.001)	0.752
RwGN	$\mathcal{D}_2$	None	0.5603 (<0.001)	0.783
RwGN	$\mathcal{D}_3$	None	0.9421 (<0.001)	0.971
TimeGAN	$\mathcal{D}_1$	Rounding	0.0050 (0.195)	0.502
RwGN	$\mathcal{D}_1$	Rounding	0.0088 (<0.001)	0.504

Table 5.2: Summary of differences in empirical p-identifiability distributions between training and unseen datasets. This was calculated for both generative model types, TimeGAN model and resampling with Gaussian noise (RwGN), and for varying training dataset size. Where 'rounding' was applied, both real and synthetic datasets were rounded to match the same precision level as the original unprocessed time-series. $\mathcal{D}_1$ had 19996 real observations, $\mathcal{D}_2$ had 2017 and $\mathcal{D}_3$ had 192. The table shows the Kolmogorov-Smirnov statistic with p-values (computed using an approximate permutation test), and the AUC ROC for a membership inference attack that used p-identifiability scores to distinguish between training and unseen observations.

statistic, from 10000 permutations. This was significant at $\alpha = 0.01$ in every scenario except one, which suggests the empirical distributions were indeed different, but it does not necessarily provide evidence that an attacker could distinguish between seen and unseen observations. Therefore, I also calculated the area under curve of the receiver-operating characteristic (ROC AUC), for a binary classification using p_i (i.e. for every $q \in [0, 1]$, any real observation with $p_i > q$ was classified as a training observation and any real observation with $p_i < q$ as an unseen observation). Apart from $\mathcal{D}_3$, where there was clearly insufficient training data, the ROC AUC was about 0.5 for TimeGAN models, which showed that membership inference in this case was essentially random. For RwGN, the AUC was very high, suggesting that RwGN overfits the training data, despite having relatively low mean p-identifiability.

Measurement precision and distance metric. One final consideration was that the original unprocessed time-series (temperature and heart rate) were actually recorded at a certain measurement precision. In the preceding paragraphs, I had added uniform noise prior to generative model training (where rounding to same measurement precision restored the original data). Generating synthetic observations at the same precision level as the original data impacted both computations and results. Using $\mathcal{D}_1$ againm, I generated synthetic observations with TimeGAN and with RwGN, having first reverted back to the original precision of the data. Synthetic observations were discretised in post-processing to the same precision levels. As the real observations now had discretised values, it was possible that some nearest neighbours were no longer unique (i.e. if there

were multiple nearest neighbours on the surface of the ball $\mathcal{B}_i$). I did not make the inequality in Equation 5.4 strict, so any synthetic observation on the surface of the ball was deemed identifying. For TimeGAN, discretising the synthetic observations reduced the p-identifiability, allowed larger synthetic datasets at the ϵ-identifiability threshold, and reduced the difference between training and unseen empirical p-identifiability distributions. For RwGN, there was little change in the p-identifiability scores, but the difference in empirical distribution between training and unseen observations was much reduced.

5.3 Fidelity: time-series length and information

In Section 5.1.3, I described how fidelity is the difference between the intractable true distribution $\mathbb{P}_r$ and the approximate distribution $\mathbb{P}_g$, but that there is no universal notion of how best to define or minimise this as a statistical divergence. Multiple elements from both model setup and model training can influence the similarity between the real and synthetic datasets, both implicitly and explicitly. Given that complete alignment between the true distribution $\mathbb{P}_r$ and the approximate distribution $\mathbb{P}_g$ is essentially impossible (unless the true distribution is unrealistically simple), these will almost certainly differ on some statistical properties. If the generative model is trained on a large number of real observations, then it should be able to learn many properties of the true distribution without any specific intervention by the modeller, but this is not guaranteed. I showed in Chapters 2 and 3 that that causal influence between the bivariate time-series can be described by information-theoretic measures, and that this can provide evidence of causal relationships in the underlying physiological systems. It is highly likely that this extends to relationships between other physiological systems (e.g. cardiovascular, immune, endocrine, respiratory), so there are clear reasons why synthetic physiological waveform data should preserve these information-theoretic temporal relationships.

In this section, I investigated whether this was the case for synthetic (temperature and heart rate) time-series data, again using the TimeGAN model and for real observations with added Gaussian noise (without resampling). After generating the synthetic datasets, I estimated the mutual information (Equation 2.6) and transfer entropy (Equation 2.7) for the two variables, and compared their empirical distributions between real and synthetic datasets. I decided not to adapt the TimeGAN model to this goal, preferring instead to see if it would learn information-theoretical relationships without external modification. As before, I calculated the mutual information and transfer entropy using the Kraskov-Stögbauer-Grassberger algorithm (Equation 2.8) [69].

5.3.1 Time-series length

The first problem I faced was related to the length of each observation T. The datasets described in Section 5.2.3 contained up to 24hrs of data for each patient, split into hour-long observations of length $T = 60$. However, I was unsure whether these observations would be of sufficient length to allow a reliable estimation of entropy-like measures, using any of the algorithms described in Section 2.1.3 (e.g. Equations 2.8 and 2.10). In Section 2.2, I investigated the performance of causal influence indices for bivariate time-series of length $T \geq 1000$. I briefly returned to the Ulam lattice (UL) experiments from Section 2.2 (Equation 2.16) with simulated data of length $T = 100$, and found that the transfer entropy estimates were unreliable, biased and had high variance (as did almost all other causal influence indices). This meant that it was necessary to use a dataset that contained longer observations here, which reduced the number of observations available for generative model training.

Data. I followed the same process as Section 5.2.3 to create a dataset containing temperature and heart rate time-series data, but I split the time-series for each patient into 6hr segments. This was a compromise between the number of observations and the length of observations. Only 1498 patients (of the previous 1751) had at least one continuous 6hr period of data collection, leaving a dataset containing 3000 observations from 1498 patients (with dimension $D = 2 \times 360$). As this was a proof-of-concept exercise, I was not looking to optimise model hyperparameters or to test on unseen data, so I trained on the whole dataset without a validation or test set, in order to maximise the amount of training observations available to the generative models.

Visualisation. Visualisation of model outputs is not always the most reliable model assessment, but it is useful as a sanity check for identifying when the generative model has poor performance or displays obvious signs of failure modes. When trained on the 6hr time-series data, synthetic observations from the TimeGAN model had clearly unrealistic behaviour, with large periodic spiking in both variables. Examples are shown in Figure 5.5 (centre left column). In this figure, I took the nearest-neighbouring synthetic observation to the example real observations (far left column), to form matching real-synthetic observation pairs. If these synthetic observations were observed in a real setting, they would almost certainly be labelled as artefactual. As the TimeGAN model failed the first fidelity hurdle, I felt that this synthetic dataset was unsuitable for providing comparisons between the empirical distributions of mutual information and transfer entropy values. As I did not intend to modify the TimeGAN model in order to improve fidelity, exploring why the model learnt this behaviour was not relevant to the analysis here (even though it would be of interest). TimeGAN had nearly 50000 trainable parameters and sampled latent random

inputs of length $D = 720$, so I believe that there was either insufficient training data or that the model architecture is simply not well-suited to long time-series.

Instead, I decided to return to the dataset and already-trained TimeGAN model from from Section 5.2, which contained 1hr observations. Synthetic observations from this model had appeared more realistic. I built a synthetic dataset containing 6hr observations as follows: I split each 6hr real observation back into 1hr sections, found the nearest-neighbouring 1hr synthetic observation and then concatenated both back up to 6hr observations. Each of the six 1hr segment within the 6hr synthetic observations should approximately match the corresponding 1hr segment of the 6hr real observation, and so this 6hr synthetic observation should appear more 'realistic' to a human eye. I made no attempts to post-process the observations, e.g. to align the endpoints of successive 1hr observations. The effect of this was occasional steep jumps at each 1hr mark, but overall these synthetic observations appeared more similar to the real observations than the full 6hr synthetic observations. This can be observed in the second example observation in Figure 5.5 (middle row, centre left column).

Gaussian noise. Finally, I also created synthetic observations by adding Gaussian noise to the real observations (Equation 5.3). As I was comparing 'neighbouring' synthetic observations from TimeGAN to the real observations, I did not use resampling here. I used the same variance noise as in Section 5.2 (i.e. $\sigma^2 = 0.02$).

5.3.2 Comparing empirical distributions of information-theoretic measures

Figure 5.6 show histograms of the mutual information and transfer entropy (both directions) between temperature and heart rate for the real observations, the concatenated 1hr TimeGAN synthetic observations and the observations with added Gaussian noise. As these information-theoretic measures are generally not independent of each other, I also showed the joint distribution of each pair of measures in Figure 5.7 using heatmaps to visualise 2d histograms. The latter shows that there is moderate positive correlation between mutual information and both transfer entropies but little to no correlation between the two transfer entropies. I returned to this point later, in Figure 5.8.

Somewhat surprisingly, synthetic observations from TimeGAN had higher values than the real data, for all information-theoretic measures. This suggests a stronger causal influence between the two variables, when the opposite (i.e. that the generative model fails to fully capture the complexities of their association) is perhaps more intuitive. There are two artefacts of the synthetic observations that may offer partial explanation. Firstly, some synthetic observations still appear to have an invented periodic spiking in both variables (for example, the centre left, bottom row observation in Figure 5.5). Joint periodicity can

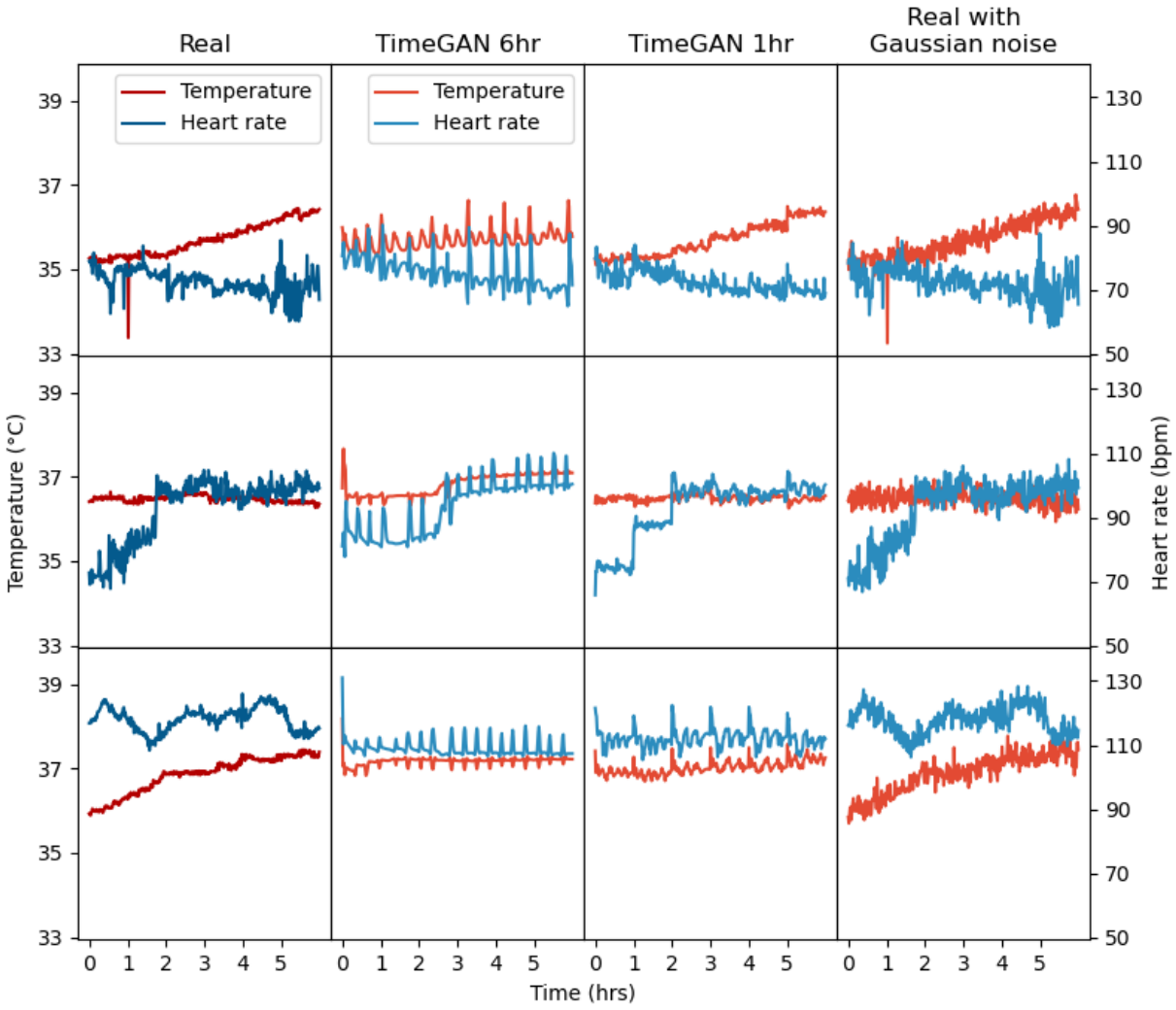

Figure 5.5: Example real and synthetic observations of temperature and heart rate time-series. Three real observations (far left column) were randomly selected from the training dataset and the corresponding synthetic observations were their nearest neighbours within each synthetic dataset. The TimeGAN model struggled to reproduce realistic looking observations when trained on the the full 6hr time-series observations (centre left column), with unrealistic periodic spiking. The centre right column instead shows synthetic observations from a TimeGAN model trained on 1hr segments, with the nearest neighbours (of the corresponding real 1hr observations) concatenated together to form a 6hr observation. The far right column shows the real observations with added Gaussian noise.

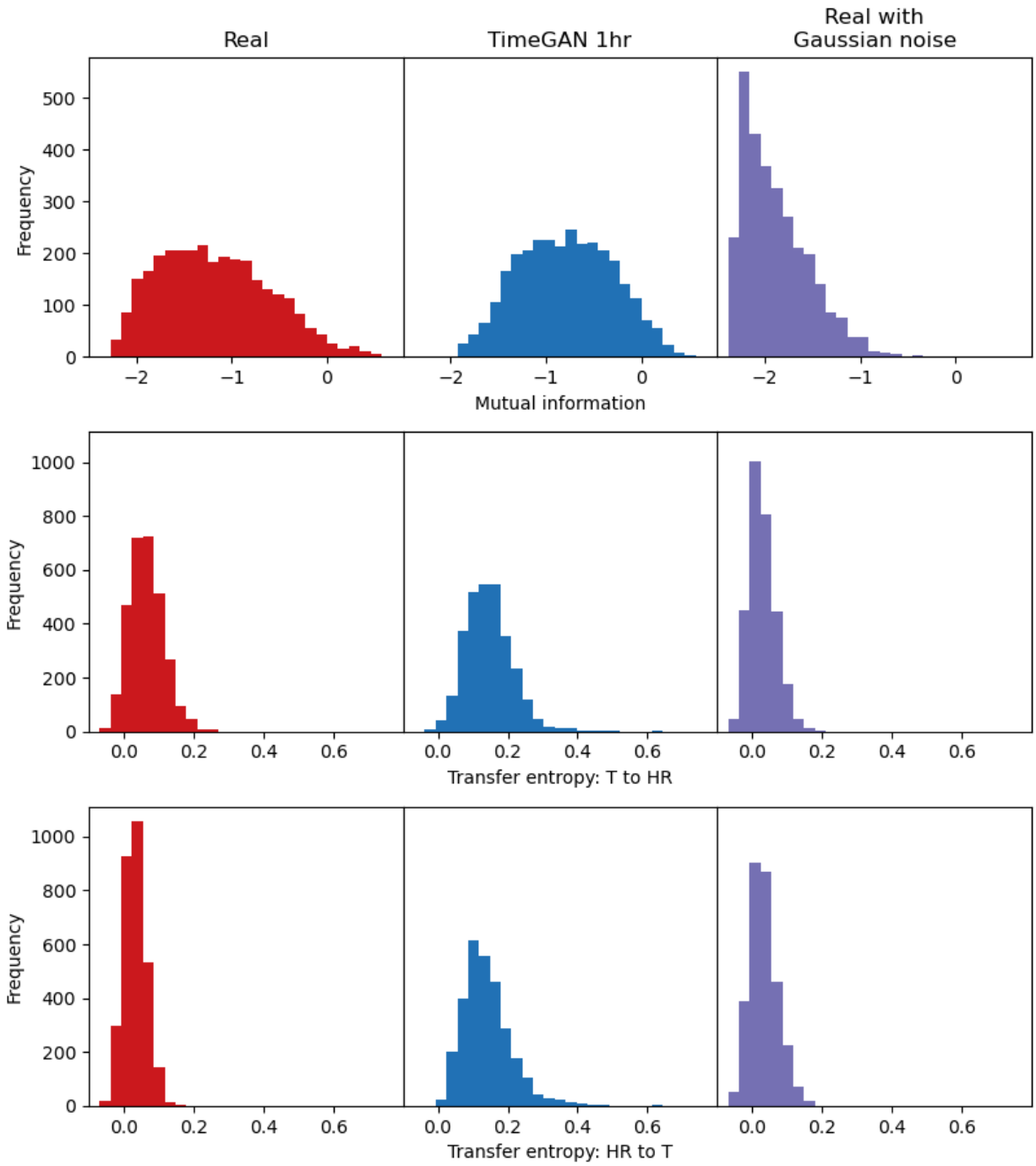

Figure 5.6: Histograms for mutual information and transfer entropy values. These were evaluated for heart rate and temperature time-series (real and synthetic). In the middle column, the TimeGAN synthetic dataset contained 1hr synthetic observation segments, which were concatenated to 6hr observations using a nearest-neighbouring approach.

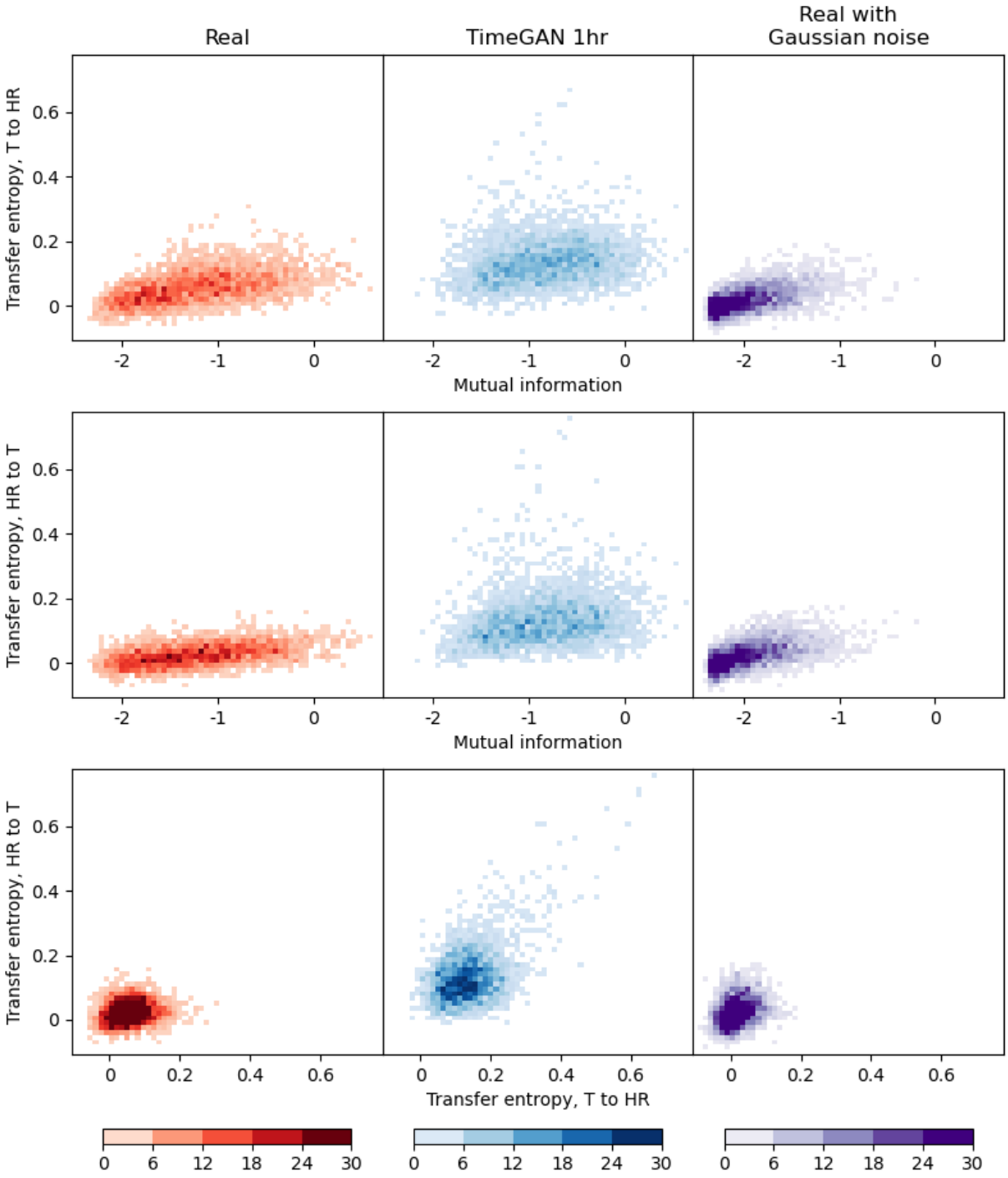

Figure 5.7: 2d histograms for mutual information and transfer entropy values. These were evaluated for heart rate and temperature time-series (real and synthetic), and shown as heatmaps. In the middle column, the TimeGAN synthetic dataset contained 1hr synthetic observation segments, which were concatenated to 6hr observations using a nearest-neighbouring approach.

be symptomatic of causal forcing of one variable on the other and result in higher values of mutual information and transfer entropy. There is certainly real physiological interaction between temperature and heart rate, but this periodicity with peaks occurring at 10-12 minute intervals in not realistic. The other artefact that may bias the information-theoretic measures are the artificial discontinuities at every hour. The KSG estimation uses nearest neighbour based counts for short 'past information' sequences within each observation, and regular concurrent discontinuities may have some effects on this estimation.

In contrast, mutual information and, to a lesser extent, transfer entropy were reduced for synthetic observations with added Gaussian noise. I had previously investigated the effect of Gaussian noise on transfer entropy estimation (and other causal indices) in Section 2.2, and showed that this reduced the value of transfer entropy. This was an expected result, since this estimates the amount of causal influence exerted by the 'recent history' of one variable on the 'present value' of the other, and adding independent noise at each timestamp to both time-series dilutes this signal.

In Figure 5.7, there appeared to be moderate positive correlation between mutual information and both transfer entropies in the real dataset, but little to no correlation between the transfer entropies. As each synthetic observation is defined with respect to a neighbouring real observation, I calculated the Pearson correlation coefficient between all observation pairs (matched across datasets), in each information-theoretic measure. As expected, there were moderately strong correlations between the real and Gaussian noise datasets for each measure. However, there were zero correlations in between the real dataset and synthetic 1hr TimeGAN dataset, despite many of the real and synthetic observations looking reasonably similar. In addition, there was much stronger correlation between transfer entropies for synthetic TimeGAN 1hr synthetic dataset than for the real dataset, which was unusual (there should be strong causal influence in one direction only, as per Chapters 2 and 3).

These results highlighted that (i) noisy resampling of real observations may perform well with regard to fidelity metrics [183] but results in weaker information-theoretic relationships between variables, and (ii) current state-of-the-art generative deep learning architectures do not accurately learn information-theoretic relationships between multivariate time-series and further work is needed to develop architectures that are capable of implicitly learning these relationships.

5.4 Utility: downstream epidemiology with Sepsis-3

I approached the third main contribution of this chapter via the insights I had gained while working on a sepsis epidemiology project, which had involved identifying sepsis incidence in ICU and describing trajectories of sepsis status during ICU stay. Sepsis is a

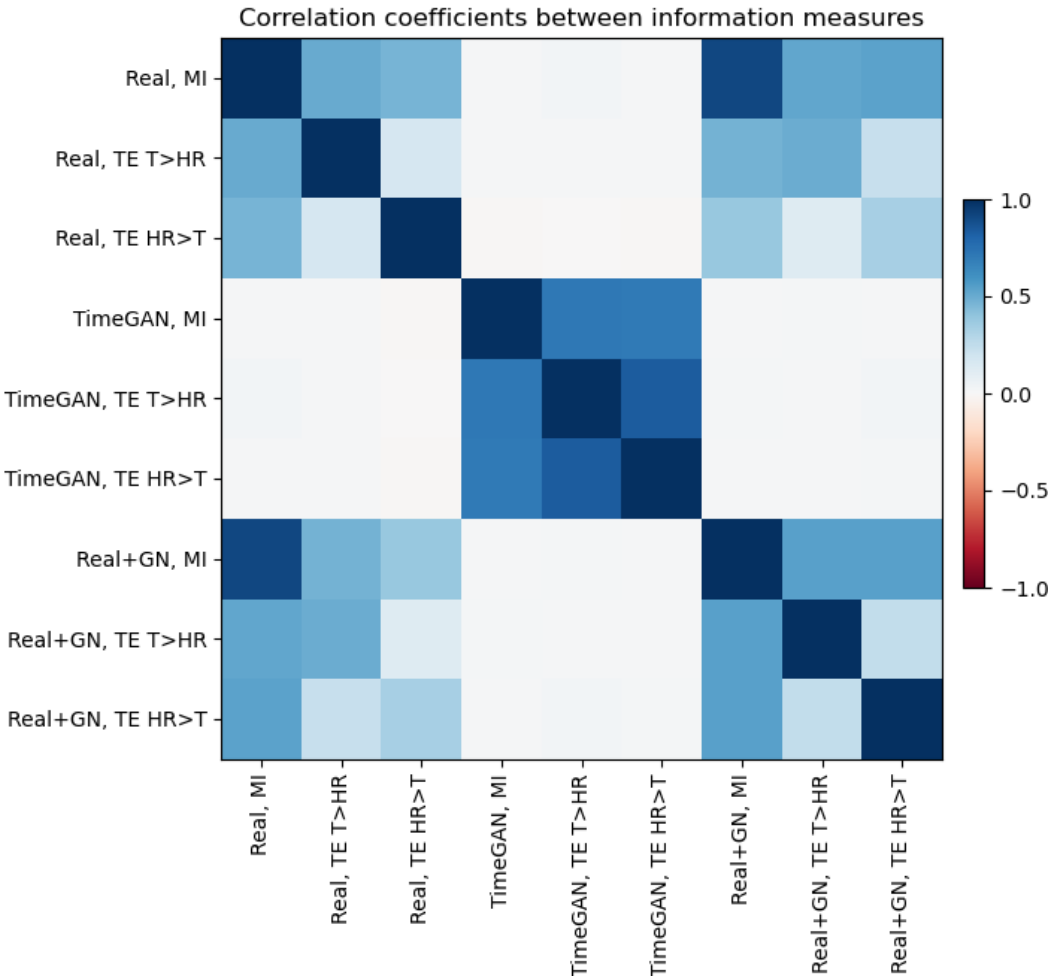

Figure 5.8: Pearson correlation between mutual information and transfer entropy pairs, across the three datasets. Real+GN: Real data with Gaussian noise, MI: mutual information, TE: transfer entropy, HR>T: heart rate to temperature, T>HR: temperature to heart rate.

major cause of mortality [216] but has proven notoriously difficult to define, even in the data-rich ICU environment. Predicting onset of sepsis is an important research question [217] but, with multiple competing notions (systematic inflammatory response syndrome, septicaemia, severe sepsis), there has not been an accepted gold-standard definition of sepsis. In 2016, an expert international taskforce recommended the Sepsis-3 criteria, defining sepsis as life-threatening organ dysfunction caused by dysregulated host response to infection [9]. These criteria defined organ dysfunction as an increase in Sequential Organ Failure Assessment (SOFA) score [218] of at least 2 points. The guidelines do not explicitly state how to define suspected infection, which is generally dependent on the practice and demographics of an individual ICU. Previous definitions of sepsis did not have a consistent link to mortality, which was deemed an essential element of the definition, for there to be confidence in its validity. This was a driving factor in the data-driven construction of the Sepsis-3 criteria.

Though there has been some opposition to the Sepsis-3 definition within the clinical community [219], Sepsis-3 has been adopted by a number of studies that operationalise these criteria within large-scale electronic health record databases, in an effort to describe sepsis epidemiology [12, 141]. Major hindrances to comprehensively studying sepsis epidemiology, and consequently to developing downstream prediction models, include the scarcity of accessible de-identified ICU records, which prevents widespread comparison between ICUs, and the lack of common and consistent data structure and labelling between current large-scale databases. The ability to generate privacy-preserving synthetic data that reliably replicates sepsis epidemiology could facilitate more extensive sepsis research, especially if combined with transfer learning to capture between-centre differences. Therefore, I investigated whether TimeGAN (as a current leading generative model for time-series) could reproduce descriptive findings from the real AmsterdamUMCdb dataset.

Sequential Organ Failure Assessment. The SOFA score is a points-based summary of a patient's organ function (or dysfunction), subdivided into six scores for different physiological systems: cardiovascular, respiratory, coagulation, renal, liver and central nervous system (Table 5.3). Each system is graded from 0 to 4, which are determined via thresholds applied to one or several physiological variables and are typically updated at least once per day. The total SOFA score is a daily sum of the maximum (non-missing) component scores within that day. If pre-ICU data is unavailable, then the SOFA score is assumed to be 0 [9]. If at least 3 SOFA components are missing during on any given day, then the total score is marked as invalid. Missing SOFA scores on the day of ICU death are set to the maximum.

Sepsis and septic shock. Following the framework of [141], I defined infection as an escalation in antibiotic treatment, i.e. an increase in the spectrum or number of

Score	Measurement	0	1	2	3	4
Respiration	PaO_2/FiO_2 ratio, mmHg	≥ 400	< 400	< 300	< 200*	< 100*
Coagulation	Platelets, $10^3/mm^3$	≥ 150	< 150	< 100	< 50	< 20
Liver	Bilirubin, μmol/l	≤ 20	> 20	> 33	> 102	> 204
Cardiovascular	MAP, mmHg	≥ 70	< 70			
or	Dobutamine, μg/kg·min			> 0		
or	Dopamine, μg/kg·min			≤ 5	> 5	
or	Epinephrine, μg/kg·min				≤ 0.1	> 0.1
or	Norepinephrine, μg/kg·min				≤ 0.1	> 0.1
CNS	Glasgow Coma Scale	15-16	13-14	10-12	6-9	< 6
Renal	Creatinine, μmol/l	< 110	< 170	< 299	≤ 440	> 440
or	Urine output, ml/day	≥ 500			< 500	< 200

Table 5.3: SOFA score summary, taken from [218]. Each score is the highest possible (i.e. if a patient had MAP < 70, dopamine $\leq 5\mu$g/kg·min and norepinephrine $\leq \mu$g/kg·min, then the cardiovascular score is 3). The cardiovascular drugs are vasopressors, so cardiovascular SOFA ≥ 2 forms part of the definition of septic shock. Glasgow Coma Scale measures the ocular, oral and motoric response to stimuli, grading these up to 4, 5 and 6 respectively. CNS: central nervous system, MAP: mean arterial pressure. * with ventilatory support.

antibiotics, with at least one intravenous course. I worked with Chris Williams to form a curated list of antibiotics (in consultation with Ari Ercole, plus Patrick Thoral and Paul Elbers from Amsterdam UMC). Some antibiotic administration is prophylactic, with the antibiotics given routinely to (almost) all admissions or to the subset of patients arriving from cardiothoracic or elective surgery. In particular, Amsterdam UMC practices selective digestive decontamination in ICU, which includes a four-day course of cefotaxime. If an infection is suspected during this period, then cefotaxime may be switched to ceftriaxone, a similar antibiotic. This was context-specific knowledge about Amsterdam UMC that a generative model must either be explicitly told or learn by itself.

Sepsis was identified when the SOFA score increased by ≥ 2 on consecutive days with antibiotic escalation on either day, or when the SOFA score was at least 2 points higher on the day after antibiotic escalation compared to the day before. Septic shock is a particularly serious subset of sepsis, in which metabolic and circulatory dysfunction leads to much higher mortality. This is defined in the Sepsis-3 criteria as sepsis accompanied by administration of vasopressors (cardiovascular SOFA score of at least 2) and a lactate level > 2mmol/l.

In the absence of clear criteria to define a reduction in sepsis status (i.e. from septic shock to sepsis without shock), Chris and I defined the following conditions under which the sepsis status was assumed to remain unchanged on consecutive days: (i) a sepsis episode continued for as long as the current SOFA score was higher than a SOFA baseline

and the patient remained on antibiotics, and (ii) a septic shock episode continued until the lactate level decreased below 2 or vasopressors were discontinued. We defined the baseline SOFA score for a sepsis episode as the smaller SOFA score when the SOFA increase of ≥ 2 occurred, or as 0 on admission. Lastly, we defined ICU mortality as death whilst in ICU or within 24hrs of discharge from ICU to another ward.

5.4.1 Increasing the synthetic data granularity

To evaluate the ability of TimeGAN to capture sepsis epidemiology, I formed three datasets at different levels of variable and temporal granularity. These were as follows:

- **Dataset #1.** This contained categorical SOFA component scores, a binary antibiotic escalation variable (as a proxy for suspected infection), the daily max lactate value and a binary indicator for ICU death. Each real observation contained daily time-series data for one patient during their ICU stay, for a maximum of 14 days after admission. The observations were of variable length up to $T = 14$, depending on the ICU length of stay.
- **Dataset #2.** This contained daily summaries of each of the variables involved in SOFA score calculation (listed in Table 5.3), the maximum antibiotic rank and number of antibiotics at maximum rank (used to calculate antibiotic escalation), a binary indicator for whether any antibiotics were given intravenously, the daily max lactate value and a binary indicator for ICU death. As before, each real observation contained daily time-series data for one patient during their ICU stay and the observations were of variable length, up to $T = 14$.
- **Dataset #3.** This contained hourly physiological variables, binary indicators for hourly administration of 32 antibiotics used to treat infection, the hourly max lactate value and a binary indicator for ICU death. The physiological variables involved all of those in Table 5.3, except with PaO2 and FiO2 as separate variables instead of the ratio between them, and with hourly urine output instead of daily. I also included additional variables that were not involved in the Sepsis-3 criteria but were useful to describe the epidemiology, as follows: heart rate, mechanical ventilation, gender, age (categorised) and admission type (including whether the admission was for cardiothoracic surgery). Each real observation contained hourly time-series for one patient, for a maximum of 24hrs after ICU admission. Observations were of variable length, up to $T = 24$.

The synthetic datasets associated with each dataset were similarly labelled (e.g. Synthetic #1). Additionally, there were five synthetic datasets associated with Dataset #3, which were labelled (a-e). Each came from a different initialisation of the same TimeGAN

generative model (i.e. identical architecture and hyperparameters, different learned weights).

The real datasets contained a mixture of continuous and categorical variables, some of which were immutable for each patient (i.e. demographics, admission category and outcome). Additionally, some variables had a high proportion of missing data. I decided to handle both issues naïvely in preprocessing and post-processing, rather than adjust the TimeGAN model architecture to handle this. This involved the following:

- I set thresholds for each continuous variable at the 1^{st} and 99^{th} percentiles, q_1 and q_{99}. For most variables, I set outliers to be equal to these threshold (i.e. the smallest 1% of values were set to the 1^{st} percentile). For a small number of variables, including mean arterial pressure and heart rate, I set these outlier values to NaN, and treated them as missing data.
- For Dataset #3, I imputed a subset of variables that were not typically recorded every hour, using last observation carried forward (LOCF) imputation. This included creatinine, platelets, bilirubin, PaO_2, FiO_2 and GCS scores.
- For all datasets, I defined a fixed implausible value for each variable, with any missing data set to this value. For some variables lower values indicate organ dysfunction, e.g. platelets. This is reversed for others, with high values indicating organ dysfunction, e.g. bilirubin. For the former, I set the imputed NaN value to be approximately equal to $q_1 - (q_{99} - q_1)/9$. For the latter, I set the imputed NaN value to be approximately equal to $q_{99} + (q_{99} - q_1)/9$. This meant that, after min-max scaling the dataset, the minimum (or maximum) value for normal data was 0.1 (or 0.9), with NaN value 0 (or 1), when low (or high) values were indicative of organ dysfunction.
- For categorical variables, I imputed missing data as -1, in cases where there was any instance of missing data in the real dataset. If there was no missing data, then I did not create this additional category. I then treated categorical variables as continuous variables, rather than e.g. adding a soft-max activation layer in the generative model. When min-max scaling these categorical variables, I set the min value to be the smallest categorical value -0.5 and the max value to be the largest +0.5. This meant that if there were n categories, each category would have an interval of length $1/(n+1)$ associated with it, i.e. the k^{th} category value would be the centre of the interval $[k/(n+1), (k+1)/(n+1)]$.
- After model training and generation of synthetic dataset, I inverted the min-max scaling. For immutable demographic variables, I took the mean value across the synthetic observation. I then rounded all categorical variables. Any value below or above the thresholds q_1 and q_{99} were determined to be 'missing' in the synthetic dataset. For Synthetic #3, I performed the same LOCF imputations as before.

For both Synthetic #2 and #3, I calculated SOFA component scores from the physiological variables. Additional post-processing included identifying prophylactic antibiotics according to the same process as the real dataset, e.g. post-elective surgery antibiotics in the first 24hr, post-cardiac vancomycin, and cefotaxime.

5.4.2 Comparing real and synthetic sepsis epidemiology

Datasets #1 and #2 contained all the data required to identify septic shock and sepsis without shock according to the Sepsis-3 criteria, during each day of ICU stay. For every day up to 14 days after ICU admission, I assigned each patient to one of five categories according to their sepsis and outcome status. These were: no sepsis, sepsis without shock, septic shock, discharged from ICU and died. Grouping patients by their sepsis status at admission, I highlighted the differences between the 'sepsis trajectories' for real and synthetic datasets (Figure 5.9). In the real AmsterdamUMCdb dataset, the majority of patients did not have sepsis at admission. Of these (bottom row, left), only a small proportion developed sepsis later in their ICU stay, while almost half were discharged within 48hrs and most discharged within 14 days. Of the patients with sepsis at admission, about 1 in 3 had septic shock. The mortality of patients with septic shock at admission (second row, left) was much higher than among other patients, and fewer patients were discharged within 14 days. The number of patients with continued septic shock gradually decreased over 14 days and, in some instances, patients with septic shock at admission were downgraded to sepsis without shock or to no sepsis after only 24hrs. Among patients who had sepsis without shock at admission (third row, left), a large number no longer had sepsis after 24hr. The number in this category then decreased by the next day, but this was likely due to the fact a subset of these patients showed considerable improvement over the first 48hrs, going from sepsis without shock to no sepsis to discharged on consecutive days.

The synthetic dataset from the model trained on the SOFA component scores (Synthetic #1) had a reasonably comparable number of sepsis and septic shock patients on admission (Figure 5.9, first row, middle). However, very few synthetic patients with septic shock at admission (second row, middle) progressed to sepsis without shock or to no sepsis, even though a significant proportion ended up discharged. The same was true for sepsis without shock at admission. The other notable point with this synthetic dataset was the much higher overall ICU mortality, particularly in patients admitted to ICU without sepsis (fourth row, middle). In the second synthetic dataset (Synthetic #2), which was trained on daily physiological variables, there were several concerning patterns. Firstly, there was almost no incidence of septic shock, either at admission or throughout the whole 14 day period. Secondly, there were zero ICU deaths among patients with sepsis at admission. As the Sepsis-3 criteria was designed to ensure the relationship between sepsis and mortality

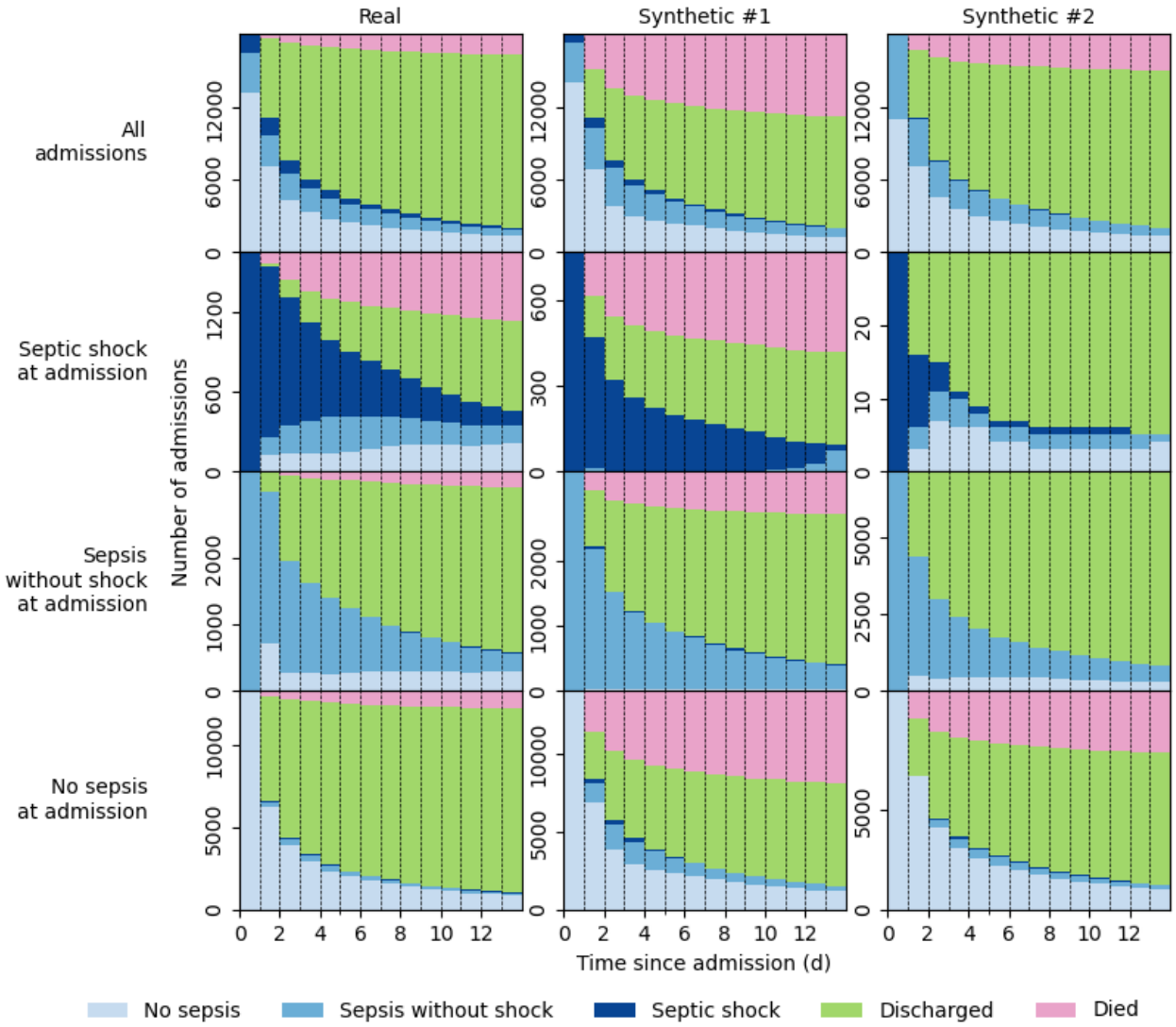

Figure 5.9: Trajectories of ICU admissions for the real dataset and two synthetic datasets. This is shown for up to 14 days in ICU and stratified by admission sepsis status. The left column was from the real AmsterdamUMCdb data; the middle column was derived from a synthetic dataset trained on categorical SOFA component scores for each day, alongside antibiotic escalation, max lactate and ICU death; and the column on the right was derived from a synthetic dataset trained on the daily summaries of the continuous variables whose thresholding provides the SOFA component scores, alongside the maximum antibiotic rank, max lactate and ICU death. For each day, a patient belonged to one of five mutually-exclusive categories (no sepsis, sepsis without shock, septic shock, discharged, died). The conditions for moving to another category were defined in the main text. Patients were further grouped (in rows) by their sepsis status upon ICU admission.

was reflected within the criteria, this synthetic dataset was clearly insufficient.

I delved deeper into the sepsis epidemiology with Synthetic #3a. Tables 5.4 and 5.5 summarised the demographics, admission category, first 24hr physiology and outcomes for real AmsterdamUMCdb dataset and the synthetic data from dataset Synthetic #3a. In this table, patients were again grouped by sepsis status at admission. I highlighted cells when the percentage difference for each variable (of the median or of the sum) between the real and synthetic datasets was large in absolute value. Some univariate distributions were well-preserved in the synthetic dataset, even across the sepsis admission categories, e.g. gender, max heart rate, minimum mean arterial pressure and PaO_2. However, the mortality in each group was again completely at odds with the real data (and the findings of multiple similar sepsis epidemiology studies, such as [12, 141]). The age profile of admissions was also somewhat different in the synthetic dataset, while a much higher proportion of septic shock patients were emergency surgical admissions rather than emergency medical. Finally, the length of stay was significantly higher in the synthetic data in all categories.

I had observed a much lower than expected ICU mortality for patients with septic shock in two of these first three synthetic datasets (Synthetic #1, #2 and #3a), which alone was a significant enough flaw to render these datasets unviable for research purposes. However, I decided to re-train an additional four TimeGAN models with the same architecture, hyperparameters and training data as Synthetic #3a, to see if this was consistently reproduced (Synthetic #3b-e). This turned out not to be the case, but it instead raised concerns about the replicability of the TimeGAN model in this setting. Table 5.6 shows the number of admissions, number of females and ICU mortality for the real dataset and each of the synthetic datasets described in this section. Again, I divided this table into sepsis status at admission categories. While the TimeGAN model generally managed to reproduce the overall ICU mortality, the ICU mortality in each admission category varied wildly across synthetic datasets. In contrast to earlier synthetic datasets (Synthetic #2 and #3a), which had unrealistic low mortality for patients with septic shock, the additional synthetic datasets all overestimated the mortality in this cohort. The number of patient admitted with septic shock also varied significantly between the synthetic datasets. Additionally, in all but one of the synthetic counterparts to Dataset #3, an unexpectedly large majority of patients with septic shock at admission were female, again deviating hugely from the real AmsterdamUMCdb data. In other groups, the picture was slightly more consistent, but it is clear that any associations the TimeGAN generative model has learnt does not include key physiologically-grounded causal relationships (i.e. males are more likely to be in ICU and to develop sepsis, while patients with septic shock have much higher mortality than general ICU admissions). In fairness, this was not a straightforward chain of relationships for the generative model to understand from within the data, as sepsis status was entangled within a set of physiological variables and subsequently disentangled

	Septic shock		Sepsis without shock	
	Real	Synthetic #3a	Real	Synthetic #3a
No. of admissions	1661	314	3317	6218
Female, n (%)	611 (36.8)	122 (38.9)	1230 (37.1)	2348 (37.8)
Age 18-39, n (%)	205 (12.3)	0 (0)	420 (12.7)	1220 (19.6)
Age 40-49, n (%)	177 (10.7)	67 (21.3)	302 (9.1)	1239 (19.9)
Age 50-59, n (%)	251 (15.1)	91 (29.0)	546 (16.5)	618 (9.9)
Age 60-69, n (%)	395 (23.8)	55 (17.5)	833 (25.1)	1304 (21.0)
Age 70-79, n (%)	410 (24.7)	88 (28.0)	867 (26.1)	1798 (28.9)
Age 80+, n (%)	223 (13.4)	13 (4.1)	349 (10.5)	39 (0.6)
Admission cat., n (%)				
Elective surgical	0 (0)	0 (0)	0 (0)	0 (0)
Emergency surgical	314 (18.9)	155 (49.4)	349 (10.5)	518 (8.3)
Emergency medical	1347 (81.1)	159 (50.6)	2968 (89.5)	5668 (91.2)
Cardiothoracic	15 (0.9)	0 (0)	48 (1.4)	32 (0.5)
First 24hr physiology				
Max heart rate	121 (104-136)	113 (101-118)	108 (92-126)	103 (95-114)
Min MAP, mmHg	53 (44-61)	59 (54-65)	60 (53-66)	63 (61-67)
Max FiO2	0.69 (0.51-0.91)	0.68 (0.60-0.75)	0.51 (0.41-0.80)	0.61 (0.58-0.65)
Min PaO2, mmHg	71 (61-84)	62 (59-67)	77 (66-94)	71 (65-76)
Min PaO2:FiO2 ratio	132 (88-198)	97 (83-125)	180 (118-247)	117 (105-135)
Min GCS	11 (3-15)	3 (3-3)	13 (7-15)	7 (4-10)
Max creatinine, µmol/L	132 (93-204)	350 (223-480)	93 (71-125)	154 (112-384)
Min platelets	145 (78-216)	36 (32-49)	163 (101-250)	98 (36-148)
Max bilirubin, µmol/L	13.0 (8.0-26.0)	36.1 (26.4-46.6)	10.0 (6.0-17.0)	19.3 (9.7-32.6)
Max SOFA score	11 (8-13)	19 (17-20)	7 (5-9)	12 (8-14)
Vasopressors, n (%)	1661 (100.0)	314 (100.0)	1744 (52.6)	3677 (59.1)
Mechanical vent., n (%)	1531 (92.2)	314 (100.0)	2704 (81.5)	6195 (99.6)
Outcomes				
Antibiotic esc., n (%)	1661 (100.0)	314 (100.0)	3317 (100.0)	6218 (100.0)
ICU length of stay, h	140 (49-339)	438 (332-536)	67 (25-204)	199 (126-440)
ICU mortality, n (%)	634 (38.2)	16 (5.1)	380 (11.5)	8 (0.1)

Table 5.4: Summary characteristics of ICU admissions by sepsis status for the real dataset and for a synthetic dataset (continued in Table 5.5). Values are median (IQR) unless otherwise stated. The TimeGAN generative model for the synthetic dataset was trained on hourly measurements (up to the first 24hr in ICU) for all physiological variables used sepsis status definition, as well as outcomes, demographic categories and admission type. The table was colour-coded to highlight differences between the real and synthetic data, by the percentage increase (or decrease) (of the median or of n) from real to synthetic, according to the following categories -20% to 20%, 20% to 60%, 60% to 100%, >100%, -20% to -60%, -60% to -100%, <-100%.

	Other		Missing data, n (%)	
	Real	Synthetic #3a	Real	Synthetic #3a
No. of admissions	13243	11689	0 (0)	0 (0)
Female, n (%)	4035 (30.5)	3801 (32.5)	416 (2.3)	5 (0.0)
Age 18-39, n (%)	1040 (7.9)	68 (0.6)	0 (0)	0 (0)
Age 40-49, n (%)	1008 (7.6)	723 (6.2)	0 (0)	0 (0)
Age 50-59, n (%)	2237 (16.9)	1564 (13.4)	0 (0)	0 (0)
Age 60-69, n (%)	3790 (28.6)	3509 (30.0)	0 (0)	0 (0)
Age 70-79, n (%)	3880 (29.3)	5785 (49.5)	0 (0)	0 (0)
Age 80+, n (%)	1288 (9.7)	40 (0.3)	0 (0)	0 (0)
Admission cat., n (%)				
Elective surgical	7397 (55.9)	7988 (68.3)	0 (0)	99 (0.5)
Emergency surgical	1078 (8.1)	1060 (9.1)	0 (0)	99 (0.5)
Emergency medical	4768 (36.0)	2574 (22.0)	0 (0)	99 (0.5)
Cardiothoracic	5755 (43.5)	4651 (39.8)	0 (0)	0 (0)
First 24hr physiology				
Max heart rate	99 (87-113)	96 (85-112)	15 (0.1)	0 (0)
Min MAP, mmHg	59 (51-66)	61 (49-65)	6 (0.0)	0 (0)
Max FiO2	0.50 (0.41-0.60)	0.60 (0.45-0.73)	587 (3.2)	3 (0.0)
Min PaO2, mmHg	80 (69-97)	67 (59-116)	579 (3.2)	5 (0.0)
Min PaO2:FiO2 ratio	205 (148-271)	119 (82-273)	587 (3.2)	5 (0.0)
Min GCS	15 (11-15)	7 (4-10)	5517 (30.3)	12614 (69.2)
Max creatinine, µmol/L	87 (71-110)	154 (119-187)	348 (1.9)	32 (0.2)
Min platelets	148 (108-202)	102 (58-183)	321 (1.8)	0 (0)
Max bilirubin, µmol/L	10.0 (6.0-15.0)	4.3 (2.1-9.8)	9178 (50.4)	6576 (36.1)
Max SOFA score	6 (4-8)	7 (5-8)	0 (0)	0 (0)
Vasopressors, n (%)	8849 (66.8)	5082 (43.5)	0 (0)	0 (0)
Mechanical vent., n (%)	11738 (88.6)	9881 (84.5)	0 (0)	0 (0)
Outcomes				
Antibiotic esc., n (%)	109 (0.8)	410 (3.5)	0 (0)	310 (1.7)
ICU length of stay, h	24 (21-71)	60 (43-98)	0 (0)	43 (0.2)
ICU mortality, n (%)	1256 (9.5)	2757 (23.6)	0 (0)	0 (0)

Table 5.5: Summary characteristics of ICU admissions by sepsis status for the real dataset and for a synthetic dataset (continued from Table 5.4). Values are median (IQR) unless otherwise stated. The TimeGAN generative model for the synthetic dataset was trained on hourly measurements (up to the first 24hr in ICU) for all physiological variables used sepsis status definition, as well as outcomes, demographic categories and admission type. The table was colour-coded to highlight differences between the real and synthetic data, by the percentage increase (or decrease) (of the median or of n) from real to synthetic, according to the following categories -20% to 20%, 20% to 60%, 60% to 100%, >100%, -20% to -60%, -60% to -100%, <-100%.

in post-processing using the Sepsis-3 definition. However, it does illustrate how far a leading generative model currently is from reproducing synthetic medical time-series data that is both realistic and useful for clinical research.

	Overall		
	Admissions	Female, n (%)	ICU mortality, n (%)
Real	18221	5876 (33.0)	2270 (12.5)
Synthetic #1	18221	n/a	7824 (42.9)
Synthetic #2	18221	n/a	3485 (19.1)
Synthetic #3a	18221	6271 (34.4)	2781 (15.3)
Synthetic #3b	18221	4364 (24.0)	2344 (12.9)
Synthetic #3c	18221	8529 (46.8)	1910 (10.5)
Synthetic #3d	18221	9602 (52.7)	2295 (12.6)
Synthetic #3e	18221	8390 (46.0)	3036 (16.7)
	Septic shock		
	Admissions	Female, n (%)	ICU mortality, n (%)
Real	1661	611 (36.8)	634 (38.2)
Synthetic #1	314	n/a	766 (51.3)
Synthetic #2	30	n/a	0 (0)
Synthetic #3a	314	122 (38.9)	16 (5.1)
Synthetic #3b	867	678 (78.2)	522 (60.2)
Synthetic #3c	1879	926 (85.8)	823 (76.3)
Synthetic #3d	2151	1881 (87.5)	1063 (49.4)
Synthetic #3e	783	739 (94.4)	613 (78.3)
	Sepsis without shock		
	Admissions	Female, n (%)	ICU mortality, n (%)
Real	3317	1230 (37.1)	380 (11.5)
Synthetic #1	3377	n/a	714 (21.1)
Synthetic #2	7181	n/a	0 (0)
Synthetic #3a	6218	2348 (37.8)	8 (0.1)
Synthetic #3b	7587	2854 (37.6)	1769 (23.3)
Synthetic #3c	4032	1586 (39.3)	87 (2.2)
Synthetic #3d	5313	1635 (30.8)	286 (5.4)
Synthetic #3e	8249	4309 (52.2)	1691 (20.5)
	No sepsis		
	Admissions	Female, n (%)	ICU mortality, n (%)
Real	13243	4035 (30.5)	1256 (9.5)
Synthetic #1	14078	n/a	6717 (47.7)
Synthetic #2	11010	n/a	3485 (31.7)
Synthetic #3a	11689	3801 (32.5)	2757 (23.6)
Synthetic #3b	9767	832 (8.5)	53 (0.5)
Synthetic #3c	13110	6017 (45.9)	1000 (7.6)
Synthetic #3d	10757	6086 (56.6)	946 (8.8)
Synthetic #3e	9189	3342 (36.4)	732 (8.0)

Table 5.6: Summary of gender (where included) and ICU mortality by sepsis status for the real dataset and all synthetic datasets. In practice, there should be a higher incidence of sepsis and of septic shock among males and a higher ICU mortality for septic shock, as observed in the real AmsterdamUMCdb dataset (and in similar studies). This was often not observed consistently in the synthetic datasets and there was huge variability for gender and ICU mortality within synthetic datasets from generative models that were identically trained (Synthetic #3a-e).

CHAPTER 6

Conclusion

This thesis described wide-ranging methodology for uncovering and understanding structure within medical time-series data. This included causal influence estimation, modular multilevel time-series modelling, Bayesian model evaluation, artefact detection in physiological waveforms, and assessment of synthetic data. Although the theoretical work in this thesis was motivated and accompanied by applications to intensive care time-series, it is largely domain-agnostic. In particular, the integrated likelihood approach in Chapter 3 is relevant to any multilevel linear models, and the discussion and evaluation of synthetic data in Chapter 5 can be abstracted to more general synthetic datasets (including tabular and imaging data).

6.1 Summary

The key contributions and results from this work were as follows:

- The focus of **Chapter 2** was the estimation of causal influence in a bivariate temporal system. I performed an in-depth qualitative and quantitative review of causal influence indices, including a novel sensitivity analysis to evaluate the impact of real-world data issues. The underlying motivation behind this analysis was to understand whether Covid-19 affected the auto-regulation of physiological subsystems, and therefore resulted in decreased causal influence between physiological time-series variables. Having highlighted the usefulness and consistency of transfer entropy in describing causal influence, I applied information-theoretic measures (entropy, mutual information and transfer entropy) to a set of physiological time-series variables (arterial blood pressure, heart rate and temperature), for a cohort of ICU patients from the Dutch Data Warehouse for Covid-19. Finally, I visualised information-theoretic trajectories by estimating these measures within successive 24hr windows, up to 14 days in ICU. This initial data visualisation showed that

there was generally stronger causal influence from temperature to each cardiovascular variable, than in the opposite direction. Additionally, averaged across the cohort, many of the transfer entropy trajectories appeared to remain mostly constant during ICU stay, while mutual information tended to decrease over time.

- In **Chapter 3**, I continued working with these information-theoretic trajectories with the aim of providing further insights into physiological regulation in ICU, particularly for Covid-19 patients. I used two flexible multilevel time-series models to describe these trajectories. I sought to compare between the Covid-19 cohort and a similar cohort of ICU patients with sepsis and respiratory dysfunction, from AmsterdamUMCdb. To make comparisons between multiple models with respect to the same dataset (in this instance, the combined dataset of both cohorts), I developed a hybrid approach for reliable estimation of the Bayesian model evidence in high-dimensional settings, using semi-conjugate priors and integrated likelihoods. I provided analytical solutions for integrated likelihoods in single-level and multilevel linear models (the latter of which were novel), and used these in Markov chain Monte Carlo estimation of the model evidence. I showed that my approach yielded improvements in terms of bias and variance, when compared to MCMC methods using the full likelihood. Finally, I used this approach to compare trajectory models for each ICU cohort and for the combined dataset. These results provided some evidence to support a clinical hypothesis of brainstem dysfunction and impaired autoregulation for patients in ICU with Covid-19.
- In **Chapter 4**, I switched the focus of the thesis from statistical modelling to deep learning. Generative deep learning of time-series data is useful not only for learning latent representations (i.e. of the patient state), but also for generating realistic synthetic observations. I introduced a novel fully unsupervised framework for artefact detection in physiological waveforms, called DeepClean [23]. The DeepClean framework used a variational autoencoder, trained to generate artefact-free synthetic observations. This was followed by a post-processing automatic threshold to identify artefacts within real observations, using the distance between matching real-synthetic observation pairs. In addition to artefact detection, this framework also provided a method for data imputation (i.e. replacing an artefactual real observation with the matching synthetic observation), and suggested the conditions under which this appears to be successful (i.e. when the corresponding latent representation has sufficient probability).
- In **Chapter 5**, I summarised the overarching goals of synthetic data generation: privacy, fidelity and utility. I then examined an aspect of each in detail, testing the capabilities of synthetic medical time-series datasets using a state-of-the-art generative deep learning model for time-series, TimeGAN. Firstly, I showed that an

observation-wide identifiability score was dependent on the size of real and synthetic datasets, using a geometric example. I then developed this identifiability score into a property of the underlying generative model. Next, I evaluated whether the TimeGAN model implicitly preserved information-theoretic measures between multivariate time-series. Finally, I investigating whether a simplified synthetic ICU dataset (which was generated by a model that was not explicitly trained on sepsis incidence) could be used for downstream sepsis epidemiology research, and showed that the resulting synthetic datasets had inconsistent behaviour and failed to capture and replicate fundamental relationships between sepsis incidence and ICU mortality. I highlighted several flaws in the TimeGAN model performance, illustrating that there are many technical challenges that must be addressed before large-scale synthetic medical datasets can become widely-used.

6.2 Limitations and future work

Many of the limitations of my work in this thesis provided natural avenues for future research. I have outlined some of these below:

Chapter 2. The main limitation in this chapter was the difficulty in interpretation of the absolute value of any causal influence index, without needing to rely on comparison relative to other time-series data. I showed that some indices were very dependent on the time-series length, so reliable comparison between datasets can also only be performed for time-series of similar lengths. Non-parametric significance tests exist for standard linear Granger causality, e.g. [51–53, 55]. This is perhaps the most well-established causal influence method but has previously been shown to perform inconsistently for nonlinear systems [50]. Similar non-parametric tests need to be introduced for a wider range of causal influence indices. I also highlighted data issues that may introduce biases in results and interpretation (including rounding, missingness and observation noise), but there still remains some unanswered questions about how best to address or alleviate these issues in practice.

There is not a consensus in the literature about optimal hyperparameter selection in causal influence indices. Dynamical systems theory provides universal methods for identifying suitable embedding hyperparameters [80–82], but other authors have suggested domain-specific or empirical approaches are better [75, 83]. The set of causal influence indices included in my quantitative review shared a common univariate embedding step that featured constant time-delay lags, and it is perhaps unlikely that an optimal multivariate mixed embedding will be of this form.

Similarly, most estimation algorithms cannot reliably estimate causal influence in

sparse data settings, or when time-series are recorded non-simultaneously or at irregular time intervals. Sparse data is difficult to handle without implicit or explicit regularisation. Often, a typical workflow for such data may involve preprocessing to transform the data into a multivariate series with constant time intervals. However, imputation can result in significant and poorly quantified biases propagating through to causal influence estimation, and more work is needed to explicitly factor these biases into uncertainty quantification within a causal influence framework. However, some causal influence indices could be translated into a Bayesian framework with prior beliefs, in particular by exploiting the links between information theory and approximate Bayesian computation. Bayesian methods may also be useful for data that is recorded with non-constant frequency, but more work is needed to develop a consistent framework that allows causal influence estimation in this setting.

This analysis could be extended to multivariate settings (both low-dimensional and high-dimensional). Model misspecification in multivariate systems is a common issue, i.e. when potential confounding variables are omitted, omission of confounding variables can create spurious false-positive causal relationships [44]. There may also be redundancy across multiple variables which provide similar information to the effect variable, or sets of variables that interact in a synergistic manner such that their combined causal influence is greater than the 'sum of their parts'. Consequently, results from bivariate indices cannot be definitively interpreted as the existence of a fundamental direct causal relationship between two variables [75], without further knowledge of possible confounding relationships. By considering a bivariate system in isolation of possible confounders, the separability assumption (i.e. the causal variable alone contains unique information about future values of the effect variable) cannot be verified. An avenue for further work is therefore to advance this quantitative analysis beyond a bivariate setting by including possible confounding variables, in line with conditional extensions to Granger causality [34, 58, 220, 221] and transfer entropy [91].

There are two reasons why I did not investigate high-dimensional approaches like PCMCI [36] more thoroughly within this thesis. Firstly, these algorithms assess conditional independence between pairs of variables in a multivariate time-series (i.e. testing edges within the fully-connected graph). The conditional independence test is similar to the Granger causality conditional independence statement. Bivariate causal influence indices should not necessarily be neglected, since it may be that different tests of bivariate causal influence can be used in this stage of the algorithm. This is further complicated because analysing pairwise conditional independence statements for high-dimensional data is computationally expensive, so estimation of bivariate causal relationships must be efficient. Secondly, these methods are designed for high-dimensional settings where many variables are recorded at uniform frequency. This is not really the case for physiological time-series

data from intensive care. My initial plan was to investigate just two physiological signals (temperature and heart rate), though this was later expanded to include arterial blood pressure recordings as well, so I focused solely on bivariate methods in Chapter 2. Only a small number of physiological signals are routinely recorded in intensive care and most other variables, e.g. laboratory results, are infrequently and irregularly recorded. As such, high-dimensional methods such as PCMCI may not be the most suitable for this setting. However, PCMCI is also useful for identifying suitable values of embedding hyperparameters, including τ and m. As mentioned above, this is something that has been not been explored fully within this thesis.

In terms of the application to physiological time-series from intensive care, more work is needed to understand the trajectories and relationships shown in Figures 2.11 and 2.10. This was expanded upon further in Chapter 3. As noted by one of my examiners, analysis of causal influence may not be necessary, in order to test the clinical hypothesis that severe disease disrupts causal relationships between physiological systems. In other words, it is perhaps sufficient only to quantify changes in temporal correlations between the physiological variables. During my viva, my examiners and I discussed comparing the findings from this chapter with simpler cross-correlation based indices. Unfortunately, time-limited access to the raw data from the Dutch Data Warehouse database meant that I was unable to revisit this during thesis corrections.

Chapter 3. Likewise, model misspecification is also an issue in the multilevel models discussed in Chapter 3. The model that best describes the data (and achieves the highest model evidence) may not necessarily be the most useful for a given research objective, e.g. if the goal is to perform inference on certain covariates, then these covariates should not be naïvely excluded on the basis of model evidence comparison. On a similar note, the choice of suitable priors, e.g. weakly-informative or narrow, is often related to specific previous knowledge or is otherwise often poorly-justified. A clear discussion of what constitutes good scientific practice in this regard is needed (for instance, post-hoc maximising the model evidence with respect to prior distribution hyperparameters is akin to p-hacking). Multilevel models can also be extended to higher-dimensional settings using the Kolmogorov-Arnold representation behind the generalised additive model framework. There are also many alternatives to this regression model framework, including hidden Markov models and recurrent neural networks, which can be adapted to include multilevel structure. Given possible model misspecification in the application of information-theoretic indices to physiological time-series data, it is likely that other models are more suited to in this instance. However, time constraints meant the only models that I investigated in this chapter were univariate non-linear functions of time.

I showed that model evidence estimation using an integrated likelihood (alongside

SMC on variance parameters) performed better than using the full likelihood (with SMC on all model parameters). One point that remained unclear was whether there were any circumstances (e.g. relating to the size and structure of the data) under which this is not the case, i.e. if sampling from a highly-nonlinear integrated likelihood is sometimes more challenging than sampling from a high-dimensional product of simple distributions. Establishing the conditions under which the integrated likelihood approach performs better than the full likelihood approach will provide more clarity about which approach is more appropriate under a given research question.

I limited the theoretical work in this chapter to general linear models (Gaussian likelihood function and identity link function) with semi-conjugate Gaussian priors for model covariates β. However, this framework can be generalised to other exponential family likelihoods with conjugate priors, including to generalised linear models with discrete outcome variables, e.g. logistic regression. Most generalised linear model forms (exponential, gamma, Poisson, categorical, multinomial) have corresponding semi-conjugate or fully-conjugate prior distributions, which, if suitable given previous or assumed knowledge of model covariates, can be used in a similar integrated likelihood approach.

Chapter 4. In this chapter, I discussed the relationship between representation learning and information theory, and highlighted how latent representations can help to guide imputation when segments of the time-series contain waveform artefacts. ICU multimodality monitoring is incredibly high-dimensional, which can make it difficult for clinical staff to parse a complete snapshot of the patient state. In particular, humans tend to struggle at visualising multivariate trajectories of time-series variables. I believe that representation learning has a key role to play in the future of personalised medicine, by exploiting short and long-term temporal relationships and providing concise summaries of the patient state and trajectory.

The DeepClean framework works particularly well for structured, quasi-periodic waveforms. One of the key assumptions was that the waveform was recorded at constant frequency and that each observation had equal length. The structure of high-frequency ABP waveforms means that it is unlikely that an additional preprocessing steps to standardise waveform timestamps (e.g. using linear interpolation) would result in a significant loss in artefact detection performance. For other types of data, the same framework (a probabilistic autoencoder model with appropriate preprocessing and an automatic thresholding during post-processing) could be used to identify artefacts but these might require tailored autoencoder network architectures. The artefact detection framework can be extended to multivariate artefact detection with relatively straightforward adjustments to the generative deep learning network architecture. In a multivariate physiological waveform, artefactual segments within one waveform may help to identify events that

should perhaps invalidate all of the waveforms as similarly artefactual, e.g. extreme patient movement.

An interesting future direction would be to extend the artefact detection framework to sparse time-series measurements, e.g. daily laboratory measurements. Most generative models are incapable of handling these types of data, and may require a more explicitly Bayesian approach due to the measurement sparsity. When I spent some time shadowing Dr Ari Ercole and his colleagues on the ICU ward, they described having an internal sense of the 'trustworthiness' and 'seriousness' of various data observations. Replicating this process automatically using Bayesian generative deep learning could prove extremely useful and would perhaps align well with the human intuition, since this intuition is in some sense Bayesian as well.

Towards the end of the chapter, I discussed some of the motivations behind the model architecture and the use of ABP waveform as a test case for model evaluation, without explicitly mentioning limitations that these decisions imposed on my findings. In particular, further work is needed to extend and evaluate the DeepClean framework to more irregular ABP waveforms (e.g. patients with arrhythmias, such as atrial fibrillation or multiple ectopics) and to establish the generalisability to previously unseen waveforms from other patients. Similarly, the 'ground-truth' manual annotation was not completely objective, because artefacts are not unambiguously defined. As a result of these considerations around the properties of ABP morphology, a true gold-standard is lacking, which in turn imposes limitations on rigorous evaluation of model performance.

Chapter 5. In this chapter, time-series length posed issues in terms of synthetic data fidelity. Estimates of information-theoretic measures were not reliable when the time-series length was insufficiently short but, at the other extreme, the generative model was unable to create synthetic observations that were both long and realistic to the human eye. More work is needed to create generative deep learning architectures to achieve this. On a related note, structured representation learning approaches, which explicitly build in known causal structure and information within a deep learning framework, may help to improve synthetic dataset utility, for instance the sepsis epidemiology task on which the TimeGAN model was unsuccessful.

I believe this final chapter contains several elements for future research around synthetic time-series data, and for investigating important unanswered questions surrounding the use of large-scale synthetic medical datasets. A key component of synthetic data generation should be to establish statistical guarantees that enable data holders to make informed decisions about when and why synthetic medical datasets can and should be published, and to minimise risks to various stakeholders within the medical community (primarily the patients, clinicians and researchers). Communicating the risks and rewards of sharing

real de-anonymised medical datasets is a challenging task in itself, and the abstraction from real to synthetic datasets will often be harder for stakeholder groups to understand, so future work on synthetic datasets also needs to clearly identify tangible benefits to these stakeholders. While working on this chapter, I realised that the pathway to full open-access publication of large-scale synthetic ICU datasets requires navigating many theoretical and technical obstacles, most of which are currently poorly understood or still completely unknown. While some of these challenges, particularly those relating to patient privacy concerns, will overlap with similar questions about real ICU datasets, there are also other issues that are unique to the synthetic data generation itself. My aim for this final chapter was to highlight some of the successes and failures of current state-of-the-art generative deep learning models, and I showed that there is much still to do to understand and develop generative deep learning models in clinical applications.

Bibliography

[1] J.-L. Vincent. The coming era of precision medicine for intensive care. *Crit. Care*, 21(Suppl 3):314, December 2017.

[2] M. J. H. Aries, M. Czosnyka, K. P. Budohoski, *et al.* Continuous determination of optimal cerebral perfusion pressure in traumatic brain injury. *Crit. Care Med.*, 40(8):2456–2463, August 2012.

[3] S. N. Karmali, A. Sciusco, S. M. May, and G. L. Ackland. Heart rate variability in critical care medicine: a systematic review. *Intensive Care Med Exp*, 5(1):33, December 2017.

[4] S. M. Bishop, S. I. Yarham, V. U. Navapurkar, D. K. Menon, and A. Ercole. Multifractal analysis of hemodynamic behavior: intraoperative instability and its pharmacological manipulation. *Anesthesiology*, 117(4):810–821, October 2012.

[5] A. Müller, J. F. Kraemer, T. Penzel, *et al.* Causality in physiological signals. *Physiol. Meas.*, 37(5):46–72, May 2016.

[6] L. Gao, P. Smielewski, M. Czosnyka, and A. Ercole. Early asymmetric Cardio-Cerebral causality and outcome after severe traumatic brain injury. *J. Neurotrauma*, 34(19):2743–2752, October 2017.

[7] E. Beqiri, P. Smielewski, C. Robba, *et al.* Feasibility of individualised severe traumatic brain injury management using an automated assessment of optimal cerebral perfusion pressure: the COGiTATE phase II study protocol. *BMJ Open*, 9(9):e030727, September 2019.

[8] A. I. R. Maas, D. K. Menon, P. D. Adelson, *et al.* Traumatic brain injury: integrated approaches to improve prevention, clinical care, and research. *Lancet Neurol.*, 16(12):987–1048, December 2017.

[9] M. Singer, C. S. Deutschman, C. W. Seymour, *et al.* The third international consensus definitions for sepsis and septic shock (Sepsis-3). *JAMA*, 315(8):801–810, February 2016.

[10] D. K. Menon and A. Ercole. Critical care management of traumatic brain injury. *Handb. Clin. Neurol.*, 140:239–274, 2017.

[11] A. I. R. Maas, D. K. Menon, E. W. Steyerberg, *et al.* Collaborative European NeuroTrauma effectiveness research in traumatic brain injury (CENTER-TBI): a prospective longitudinal observational study. *Neurosurgery*, 76(1):67–80, January 2015.

[12] A. E. W. Johnson, J. Aboab, J. D. Raffa, *et al.* A comparative analysis of sepsis identification methods in an electronic database. *Crit. Care Med.*, 46(4):494–499, April 2018.

[13] T. J. Pollard, A. E. W. Johnson, J. D. Raffa, *et al.* The eICU collaborative research database, a freely available multi-center database for critical care research. *Sci Data*, 5:180178, September 2018.

[14] S. Harris, S. Shi, D. Brealey, *et al.* Critical care health informatics collaborative (CCHIC): Data, tools and methods for reproducible research: A multi-centre UK intensive care database. *Int. J. Med. Inform.*, 112:82–89, April 2018.

[15] P. J. Thoral, J. M. Peppink, R. H. Driessen, *et al.* Sharing ICU patient data responsibly under the Society of Critical Care Medicine/European Society of Intensive Care Medicine Joint Data Science collaboration: The Amsterdam University Medical Centers Database (AmsterdamUMCdb) example. *Crit. Care Med.*, 49(6):e563–e577, June 2021.

[16] L. M. Fleuren, T. A. Dam, M. Tonutti, *et al.* The Dutch Data Warehouse, a multicenter and full-admission electronic health records database for critically ill COVID-19 patients. *Crit. Care*, 25(1):304, August 2021.

[17] L. M. Fleuren, D. P. de Bruin, M. Tonutti, *et al.* Large-scale ICU data sharing for global collaboration: the first 1633 critically ill COVID-19 patients in the dutch data warehouse. *Intensive Care Med.*, 47(4):478–481, April 2021.

[18] M. Hüser, A. Kündig, W. Karlen, V. De Luca, and M. Jaggi. Forecasting intracranial hypertension using multi-scale waveform metrics. *arXiv*, February 2019. Preprint at `https://arxiv.org/abs/1902.09499`.

[19] L. Anthony Celi, R. G. Mark, D. J. Stone, and R. A. Montgomery. 'Big Data' in the Intensive Care Unit: Closing the Data Loop. *Am. J. Respir. Crit. Care Med.*, 187(11):1157–1160, June 2013.

[20] S. H. Haddad and Y. M. Arabi. Critical care management of severe traumatic brain injury in adults. *Scand. J. Trauma Resusc. Emerg. Med.*, 20:12, February 2012.

[21] T. Edinburgh, S. J. Eglen, and A. Ercole. Causality indices for bivariate time series data: A comparative review of performance. *Chaos*, 31(8):083111, August 2021.

[22] T. Edinburgh, A. Ercole, and S. Eglen. Bayesian model selection for multilevel models using integrated likelihoods. *PLoS One*, 18(2):e0280046, February 2023.

[23] T. Edinburgh, P. Smielewski, M. Czosnyka, *et al.* DeepClean: Self-supervised artefact rejection for intensive care waveform data using deep generative learning. *Acta Neurochir. Suppl.*, 131:235–241, 2021.

[24] S. J. Yong. Persistent brainstem dysfunction in Long-COVID: A hypothesis. *ACS Chem. Neurosci.*, 12(4):573–580, February 2021.

[25] M. Marshall. COVID and the brain: researchers zero in on how damage occurs. *Nature*, 595(7868):484–485, July 2021.

[26] C. W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–438, 1969.

[27] J. Runge. Causal network reconstruction from time series: From theoretical assumptions to practical estimation. *Chaos*, 28(7):075310, July 2018.

[28] J. Pearl and D. Mackenzie. *The Book of Why: The New Science of Cause and Effect*. Penguin, USA, 1st edition, April 2018.

[29] J. Berrevoets, K. Kacprzyk, Z. Qian, and M. van der Schaar. Causal deep learning. *arXiv*, March 2023. Preprint at `https://arxiv.org/abs/2303.02186`.

[30] Y. Mehta and D. Arora. Newer methods of cardiac output monitoring. *World J. Cardiol.*, 6(9):1022–1029, September 2014.

[31] C. A. Sims. Money, income, and causality. *Am. Econ. Rev.*, 62(4):540–552, 1972.

[32] T. Schreiber. Measuring information transfer. *Phys. Rev. Lett.*, 85(2):461–464, July 2000.

[33] C. W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–438, 1969.

[34] J. Geweke. Inference and causality in economic time series models. In *Handbook of Econometrics*, volume 2, pages 1101–1144. Elsevier, January 1984.

[35] D. D. Zhang, H. F. Lee, C. Wang, *et al.* The causality analysis of climate change and large-scale human crisis. *Proc. Natl. Acad. Sci. U.S.A.*, 108(42):17296–17301, October 2011.

[36] J. Runge, P. Nowack, M. Kretschmer, S. Flaxman, and D. Sejdinovic. Detecting and quantifying causal associations in large nonlinear time series datasets. *Sci Adv*, 5(11):eaau4996, November 2019.

[37] J. Runge, S. Bathiany, E. Bollt, *et al.* Inferring causation from time series in earth system sciences. *Nat. Commun.*, 10(1):2553, June 2019.

[38] C. M. Gray, P. König, A. K. Engel, and W. Singer. Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which reflects global stimulus properties. *Nature*, 338(6213):334–337, March 1989.

[39] A. K. Seth, A. B. Barrett, and L. Barnett. Granger causality analysis in neuroscience and neuroimaging. *J. Neurosci.*, 35(8):3293–3297, February 2015.

[40] M. Eichler. Graphical modelling of multivariate time series. *Probab. Theory Related Fields*, 153(1):233–268, June 2012.

[41] J. Aldrich. Correlations genuine and spurious in Pearson and Yule. *Stat. Sci.*, 10(4):364–376, November 1995.

[42] G. Sugihara, R. May, H. Ye, *et al.* Detecting causality in complex ecosystems. *Science*, 338(6106):496–500, October 2012.

[43] K. Hlaváčková-Schindler, M. Paluš, M. Vejmelka, and J. Bhattacharya. Causality detection based on information-theoretic approaches in time series analysis. *Phys. Rep.*, 441(1):1–46, March 2007.

[44] M. Eichler. Causal inference with multiple time series: principles and problems. *Philos. Trans. A Math. Phys. Eng. Sci.*, 371(1997):20110613, August 2013.

[45] A. Papana, C. Kyrtsou, D. Kugiumtzis, and C. Diks. Simulation study of direct causality measures in multivariate time series. *Entropy*, 15(7):2635–2661, July 2013.

[46] S. Palachy. Inferring causality in time series data - towards data science. `https://towardsdatascience.com/inferring-causality-in-time-series-data-b8b75fe52c46`, November 2019. Accessed on Aug 28, 2020.

[47] M. Lungarella, K. Ishiguro, Y. Kuniyoshi, and N. Otsu. Methods for quantifying the causal structure of bivariate time-series. *Int. J. Bifurcat. Chaos*, 17(03):903–921, March 2007.

[48] I. Vlachos and D. Kugiumtzis. Nonuniform state-space reconstruction and coupling detection. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.*, 82(1.2):016207, July 2010.

[49] L. Barnett, A. B. Barrett, and A. K. Seth. Granger causality and transfer entropy are equivalent for Gaussian variables. *Phys. Rev. Lett.*, 103(23):238701, December 2009.

[50] Y. Chen, S. L. Bressler, and M. Ding. Frequency decomposition of conditional granger causality and application to multivariate neural field potential data. *J. Neurosci. Methods*, 150(2):228–237, January 2006.

[51] H. Y. Toda and T. Yamamoto. Statistical inference in vector autoregressions with possibly integrated processes. *J. Econom.*, 66(1):225–250, March 1995.

[52] D. Giles. Testing for Granger causality. `https://davegiles.blogspot.com/2011/04/testing-for-granger-causality.html`, April 2011. Accessed on Aug 28, 2020.

[53] C. Hiemstra and J. D. Jones. Testing for linear and nonlinear Granger causality in the stock price-volume relation. *J. Finance*, 49(5):1639–1664, December 1994.

[54] Z. Bai, Y. Hui, D. Jiang, *et al.* A new test of multivariate nonlinear causality. *PLoS One*, 13(1):e0185155, January 2018.

[55] C. Diks and V. Panchenko. A new statistic and practical guidelines for nonparametric Granger causality testing. *J. Econ. Dyn. Control*, 30(9-10):1647–1669, 2006.

[56] C. Diks and M. Wolski. Nonlinear Granger causality: Guidelines for multivariate analysis: Multivariate nonlinear Granger causality. *J. Appl. Econ.*, 31(7):1333–1351, November 2016.

[57] N. Ancona, D. Marinazzo, and S. Stramaglia. Radial basis function approach to nonlinear Granger causality of time series. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.*, 70(5 Pt 2):056221, November 2004.

[58] Y. Chen, G. Rangarajan, J. Feng, and M. Ding. Analyzing multiple nonlinear time series with extended Granger causality. *Phys. Lett. A*, 324(1):26–35, April 2004.

[59] D. Marinazzo, M. Pellicoro, and S. Stramaglia. Kernel method for nonlinear granger causality. *Phys. Rev. Lett.*, 100(14):144103, April 2008.

[60] U. Feldmann and J. Bhattacharya. Predictability improvement as an asymmetrical measure of interdependence in bivariate time series. *Int. J. Bifurcat. Chaos*, 14(02):505–514, February 2004.

[61] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, July 1948.

[62] S. Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Stat.*, 22(1):79–86, 1951.

[63] E. T. Jaynes. Information theory and statistical mechanics. *Phys. Rev.*, 106(4):620–630, May 1957.

[64] R. Steuer, J. Kurths, C. O. Daub, J. Weise, and J. Selbig. The mutual information: detecting and evaluating dependencies between variables. *Bioinformatics*, 18 Suppl 2:231–40, October 2002.

[65] L. F. Kozachenko and N. N. Leonenko. Sample estimate of the entropy of a random vector. *Problemy Peredachi Informatsii*, 23(2):9–16, 1987.

[66] T. B. Berrett, R. J. Samworth, and M. Yuan. Efficient multivariate entropy estimation via k-nearest neighbour distances. *Annals of Statistics*, 47(1):288–318, February 2019.

[67] A. Kaiser and T. Schreiber. Information transfer in continuous processes. *Physica D*, 166(1):43–62, June 2002.

[68] M. Paluš and M. Vejmelka. Directionality of coupling from bivariate time series: How to avoid false causalities and missed connections. *Phys. Rev. E*, 75(5):056211, May 2007.

[69] A. Kraskov, H. Stögbauer, and P. Grassberger. Estimating mutual information. *Physical Review E*, 69(6):066138, June 2004.

[70] G. Gómez-Herrero, W. Wu, K. Rutanen, *et al.* Assessing coupling dynamics from an ensemble of time series. *Entropy*, 17(4):1958–1970, April 2015.

[71] R. Marschinski and H. Kantz. Analysing the information flow between financial time series. *The European Physical Journal B - Condensed Matter and Complex Systems*, 30(2):275–281, November 2002.

[72] M. Paluš. Coarse-grained entropy rates for characterization of complex time series. *Physica D*, 93(1):64–77, May 1996.

[73] M. Paluš, V. Komárek, Z. Hrncír, and K. Sterbová. Synchronization as adjustment of information rates: detection from bivariate time series. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.*, 63(4.2):046211, April 2001.

[74] F. Takens. Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence, Warwick 1980*, pages 366–381. Springer Berlin Heidelberg, 1981.

[75] J. Arnhold, P. Grassberger, K. Lehnertz, and C. E. Elger. A robust method for detecting interdependences: application to intracranially recorded EEG. *Physica D*, 134(4):419–430, December 1999.

[76] J. Bhattacharya, E. Pereda, and H. Petsche. Effective detection of coupling in short and noisy bivariate data. *IEEE Trans. Syst. Man Cybern. B Cybern.*, 33(1):85–95, February 2003.

[77] R. Q. Quiroga, J. Arnhold, and P. Grassberger. Learning driver-response relationships from synchronization patterns. *Phys. Rev. E*, 61(5):5142–5148, May 2000.

[78] D. Mønster, R. Fusaroli, K. Tylén, A. Roepstorff, and J. F. Sherson. Inferring causality from noisy time series data. *arXiv*, March 2016. Preprint at `https://arxiv.org/abs/1603.01155`.

[79] A. T. Clark, H. Ye, F. Isbell, *et al.* Spatial convergent cross mapping to detect causal relationships from short time series. *Ecology*, 96:1174–1181, May 2015.

[80] I. Vlachos and D. Kugiumtzis. State space reconstruction from multiple time series. In *Topics on Chaotic Systems*, pages 378–387. World Scientific Publishing Co., May 2009.

[81] M. B. Kennel, R. Brown, and H. D. Abarbanel. Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Phys. Rev. A*, 45(6):3403–3411, March 1992.

[82] A. M. Fraser and H. L. Swinney. Independent coordinates for strange attractors from mutual information. *Phys. Rev. A Gen. Phys.*, 33(2):1134–1140, February 1986.

[83] T. Schreiber. Interdisciplinary application of nonlinear time series methods. *Phys. Rep.*, 308(1):1–64, January 1999.

[84] C. Hsiao. Autoregressive modeling and causal ordering of economic variables. *J. Econ. Dyn. Control*, 4:243–259, November 1982.

[85] J. Runge, J. Heitzig, V. Petoukhov, and J. Kurths. Escaping the curse of dimensionality in estimating multivariate transfer entropy. *Phys. Rev. Lett.*, 108(25):258701, June 2012.

[86] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *J. R. Stat. Soc. Series B Stat. Methodol.*, 63(2):411–423, May 2001.

[87] J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

[88] P. Hall and E. J. Hannan. On stochastic complexity and nonparametric density estimation. *Biometrika*, 75(4):705–714, December 1988.

[89] M. Hénon. A two-dimensional mapping with a strange attractor. *Commun. Math. Phys.*, 50(1):69–77, 1976.

[90] T. Edinburgh. Bivariate causality indices review: code, data and figures, March 2021. `https://doi.org/10.5281/zenodo.4639395`.

[91] J. T. Lizier. JIDT: An information-theoretic toolkit for studying the dynamics of complex systems. *Frontiers in Robotics and AI*, 1:11, 2014.

[92] J. Park, C. Smith, G. Sugihara, and E. Deyle. EDM: Empirical dynamic modelling ('pyEDM'). Python package version 1.7.0., 2020. `https://github.com/SugiharaLab`.

[93] W. B. Cannon. Organization for physiological homeostasis. *Physiol. Rev.*, 9(3):399–431, July 1929.

[94] H. Modell, W. Cliff, J. Michael, *et al.* A physiologist's view of homeostasis. *Adv. Physiol. Educ.*, 39(4):259–266, December 2015.

[95] G. J. Tortora and B. H. Derrickson. *Principles of Anatomy and Physiology.* John Wiley & Sons, May 2018.

[96] E. A. Tansey and C. D. Johnson. Recent advances in thermoregulation. *Adv. Physiol. Educ.*, 39(3):139–148, September 2015.

[97] C. L. Tan and Z. A. Knight. Regulation of body temperature by the nervous system. *Neuron*, 98(1):31–48, April 2018.

[98] J. S. Shahoud, T. Sanvictores, and N. R. Aeddula. *Physiology, Arterial Pressure Regulation.* StatPearls Publishing, August 2022.

[99] M. Noda and T. Matsuda. Central regulation of body fluid homeostasis. *Proc. Jpn. Acad. Ser. B Phys. Biol. Sci.*, 98(7):283–324, 2022.

[100] P. Sterling. Allostasis: a new paradigm to explain arousal pathology. In *Handbook of life stress cognition and health*, pages 629–649. John Wiley & Sons, January 1988.

[101] P. Sterling. Allostasis: a model of predictive regulation. *Physiol. Behav.*, 106(1):5–15, April 2012.

[102] D. S. Ramsay and S. C. Woods. Clarifying the roles of homeostasis and allostasis in physiological regulation. *Psychol. Rev.*, 121(2):225–247, April 2014.

[103] M. Stumvoll, P. A. Tataranni, N. Stefan, B. Vozarova, and C. Bogardus. Glucose allostasis. *Diabetes*, 52(4):903–909, April 2003.

[104] G. D. James. The adaptive value and clinical significance of allostatic blood pressure variation. *Curr. Hypertens. Rev.*, 15(2):93–104, February 2019.

[105] A. Ercole, P. J. Hutchinson, and J. D. Pickard. Brainstem death and prolonged disorders of consciousness. In *Oxford Textbook of Medicine*. Oxford University Press, January 2020.

[106] A. L. Goldberger and B. J. West. Applications of nonlinear dynamics to clinical cardiology. *Ann. N. Y. Acad. Sci.*, 504:195–213, 1987.

[107] J. P. Saul, P. Albrecht, R. D. Berger, and R. J. Cohen. Analysis of long term heart rate variability: methods, 1/f scaling and implications. *Comput. Cardiol.*, 14:419–422, 1988.

[108] S. M. Pincus and A. L. Goldberger. Physiological time-series analysis: what does regularity quantify? *Am. J. Physiol.*, 266(4 Pt 2):H1643–56, April 1994.

[109] S. M. Pincus. Approximate entropy as a measure of system complexity. *Proc. Natl. Acad. Sci. U. S. A.*, 88(6):2297–2301, March 1991.

[110] H. Goldstein. *Multilevel Models in Educational and Social Research*. Charles Griffin & Co; Oxford University Press, 1987.

[111] A. H. Leyland and H. Goldstein. *Multilevel Modelling of Health Statistics*. Wiley series in probability and statistics. Wiley, 2001.

[112] T. Edinburgh, S. J. Eglen, P. Thoral, P. Elbers, and A. Ercole. Sepsis-3 criteria in AmsterdamUMCdb: open-source code implementation. *GigaByte*, 2022:45, March 2022.

[113] A. Gelman and J. Hill. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press, December 2006.

[114] H. Akaike. Information theory and an extension of the maximum likelihood principle. In B. N. Petrov and F. Csáki, editors, *2nd International Symposium on Information Theory*, pages 267–281, Budapest, Hungary, 1973. Akadémiai Kiadó.

[115] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *J. R. Stat. Soc.*, 57(1):289–300, January 1995.

[116] J. Neyman, E. S. Pearson, and K. Pearson. IX. On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231(694-706):289–337, February 1933.

[117] S. S. Wilks. The large-sample distribution of the likelihood ratio for testing composite hypotheses. *The Annals of Mathematical Statistics*, 9(1):60–62, March 1938.

[118] O'Hagan and Anthony. *Kendall's Advanced Theory of Statistics, Vol 2B: Bayesian Inference*. Arnold, 1994.

[119] A. Gelman. Multilevel (hierarchical) modeling: What it can and cannot do. *Technometrics*, 48(3):432–435, August 2006.

[120] T. Hastie and R. Tibshirani. Generalized additive models. *Stat. Sci.*, 1(3):297–310, August 1986.

[121] A. N. Kolmogorov. On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. *Doklady Akademii Nauk*, 114:953–956, 1957.

[122] D. Zhang, X. Lin, and M. Sowers. Semiparametric regression for periodic longitudinal hormone data from multiple menstrual cycles. *Biometrics*, 56(1):31–39, March 2000.

[123] D. Zhang and X. Lin. Hypothesis testing in semiparametric additive mixed models. *Biostatistics*, 4(1):57–74, January 2003.

[124] S. J. Taylor and B. Letham. Forecasting at scale. *Am. Stat.*, 72(1):37–45, January 2018.

[125] K. A. Bollen and P. J. Curran. Autoregressive latent trajectory (ALT) models a synthesis of two traditions. *Sociol. Methods Res.*, 32(3):336–383, February 2004.

[126] Y. Xiong, H. J. Kim, and V. Singh. Mixed effects neural networks (menets) with applications to gaze estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7743–7752, 2019.

[127] M.-N. Tran, N. Nguyen, D. Nott, and R. Kohn. Bayesian deep net GLM and GLMM. *J. Comput. Graph. Stat.*, 29(1):97–113, January 2020.

[128] H. Jeffreys. *The Theory of Probability*. OUP Oxford, August 1998.

[129] R. E. Kass and A. E. Raftery. Bayes factors. *J. Am. Stat. Assoc.*, 90(430):773–795, June 1995.

[130] T. Kloek and H. K. van Dijk. Bayesian estimates of equation system parameters: An application of integration by Monte Carlo. *Econometrica*, 46(1):1–19, January 1978.

[131] J. S. Liu and R. Chen. Sequential Monte Carlo methods for dynamic systems. *J. Am. Stat. Assoc.*, 93(443):1032–1044, September 1998.

[132] J. L. Foulley, M. San Cristobal, D. Gianola, and S. Im. Marginal likelihood and Bayesian approaches to the analysis of heterogeneous residual variances in mixed linear Gaussian models. *Comput. Stat. Data Anal.*, 13(3):291–305, April 1992.

[133] P. J. Heagerty and S. L. Zeger. Marginalized multilevel models and likelihood inference. *Stat. Sci.*, 15(1):1–19, February 2000.

[134] P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, December 1995.

[135] B. P. Carlin and S. Chib. Bayesian Model Choice via Markov Chain Monte Carlo Methods. *J. R. Stat. Soc. Series B Stat. Methodol.*, 57(3):473–484, August 1995.

[136] A. Gelman and X.-L. Meng. Simulating normalizing constants: from importance sampling to bridge sampling to path sampling. *SSO Schweiz. Monatsschr. Zahnheilkd.*, 13(2):163–185, May 1998.

[137] J. Annis, N. J. Evans, B. J. Miller, and T. J. Palmeri. Thermodynamic integration and steppingstone sampling methods for estimating bayes factors: A tutorial. *J. Math. Psychol.*, 89:67–86, April 2019.

[138] S. Bathelmé. Priors of convenience. `https://dahtah.wordpress.com/2012/08/22/priors-of-convenience`, August 2012. Accessed on Oct 10, 2022.

[139] T. Edinburgh, A. Ercole, and S. J. Eglen. Source code for "Bayesian model selection for multilevel models using integrated likelihoods". `https://doi.org/10.5281/zenodo.7314381`, November 2022.

[140] ESICM. 3rd Critical Care Datathon. `https://www.esicm.org/events/datathon-2021/`, May 2021. Accessed on Sep 22, 2021.

[141] A. D. Shah, N. S. MacCallum, S. Harris, *et al.* Descriptors of sepsis using the Sepsis-3 criteria: A cohort study in critical care units within the U.K. national institute for health research critical care health informatics collaborative. *Crit. Care Med.*, July 2021.

[142] M. Tang, E. V. Slud, and R. M. Pfeiffer. Goodness of fit tests for linear mixed models. *J. Multivar. Anal.*, 130:176–193, September 2014.

[143] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis (3rd Edition)*. Chapman and Hall/CRC, 2013.

[144] Q. Li, R. G. Mark, and G. D. Clifford. Artificial arterial blood pressure artifact models and an evaluation of a robust blood pressure and heart rate estimator. *Biomed. Eng. Online*, 8:13, July 2009.

[145] F. Scalzo and X. Hu. Semi-supervised detection of intracranial pressure alarms using waveform dynamics. *Physiol. Meas.*, 34(4):465–478, April 2013.

[146] M. C. Chambrin. Alarms in the intensive care unit: how can the number of false alarms be reduced? *Crit. Care*, 5(4):184–188, August 2001.

[147] A. M. Sullivan, H. Xia, J. C. Mc Bride, and X. Zhao. Reconstruction of missing physiological signals using artificial neural networks. *Comput. Cardiol.*, 37:317–320, October 2010.

[148] M. Megjhani, A. Alkhachroum, K. Terilli, *et al.* An active learning framework for enhancing identification of non-artifactual intracranial pressure waveforms. *Physiol. Meas.*, 40(1):015002, January 2019.

[149] J. X. Sun, A. T. Reisner, and R. G. Mark. A signal abnormality index for arterial blood pressure waveforms. In *Comput. Cardiol.*, pages 13–16, October 2006.

[150] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):15:1–15:58, July 2009.

[151] J. An and S. Cho. Variational autoencoder based anomaly detection usingreconstruction probability. *Special Lecture on IE*, 2:1–18, December 2015.

[152] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In *Parallel distributed processing: explorations in the microstructure of cognition, Vol. 1: Foundations*, pages 318–362. MIT Press, January 1986.

[153] D. P. Kingma and M. Welling. An introduction to variational autoencoders. *arXiv*, June 2019. Preprint at `https://arxiv.org/abs/1906.02691`.

[154] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On Calibration of Modern Neural Networks. *arXiv*, June 2017. Preprint at `https://arxiv.org/abs/1706.04599`.

[155] A. Makhzani and B. Frey. k-sparse autoencoders. *arXiv*, December 2013. Preprint at https://arxiv.org/abs/1312.5663.

[156] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 1096–1103. ACM, July 2008.

[157] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 11:3371–3408, Dec 2010.

[158] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML11, pages 833–840. Omnipress, 2011.

[159] D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. *arXiv*, December 2013. Preprint at https://arxiv.org/abs/1312.6114v10.

[160] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey. Adversarial autoencoders. *arXiv*, November 2015. Preprint at https://arxiv.org/abs/1511.05644.

[161] F. Huszar. Is maximum likelihood useful for representation learning? https://www.inference.vc/maximum-likelihood-for-representation-learning-2/, May 2017. Accessed on Oct 23, 2019.

[162] I. Higgins, L. Matthey, A. Pal, *et al.* b-VAE: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2(5):6, April 2017.

[163] C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther. Ladder variational autoencoders. *arXiv*, February 2016. Preprint at https://arxiv.org/abs/1602.02282.

[164] S. R. Bowman, L. Vilnis, O. Vinyals, *et al.* Generating sentences from a continuous space. *arXiv*, November 2015. Preprint at https://arxiv.org/abs/1511.06349.

[165] A. A. Alemi, B. Poole, I. Fischer, *et al.* Fixing a broken ELBO. *arXiv*, November 2017. Preprint at https://arxiv.org/abs/1711.00464.

[166] X. Chen, D. P. Kingma, T. Salimans, *et al.* Variational lossy autoencoder. *arXiv*, November 2016. Preprint at https://arxiv.org/abs/1611.02731.

[167] P. Dayan, G. E. Hinton, R. M. Neal, and R. S. Zemel. The Helmholtz machine. *Neural Comput.*, 7(5):889–904, September 1995.

[168] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18(7):1527–1554, July 2006.

[169] D. J. Rezende and S. Mohamed. Variational inference with normalizing flows. *arXiv*, May 2015. Preprint at `https://arxiv.org/abs/1505.05770`.

[170] Y. Burda, R. Grosse, and R. Salakhutdinov. Importance weighted autoencoders. *arXiv*, September 2015. Preprint at `https://arxiv.org/abs/1509.00519`.

[171] G. W. Cottrell and P. Munro. Principal components analysis of images via back propagation. In *Visual Communications and Image Processing '88: Third in a Series*, volume 1001, pages 1070–1077. International Society for Optics and Photonics, October 1988.

[172] A. F. Mejia, M. B. Nebel, A. Eloyan, B. Caffo, and M. A. Lindquist. PCA leverage: outlier detection for high-dimensional functional magnetic resonance imaging data. *Biostatistics*, 18(3):521–536, February 2017.

[173] E. J. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *arXiv*, December 2009. Preprint at `https://arxiv.org/abs/0912.3599`.

[174] P. P. Markopoulos, G. N. Karystinos, and D. A. Pados. Some options for L1-Subspace signal processing. *arXiv*, September 2013. Preprint at `https://arxiv.org/abs/1309.1194`.

[175] P. P. Markopoulos, S. Kundu, S. Chamadia, and D. A. Pados. Efficient L1-Norm Principal-Component analysis via bit flipping. *arXiv*, October 2016. Preprint at `https://arxiv.org/abs/1610.01959`.

[176] B. R. Reddy and I. S. Murthy. ECG data compression using fourier descriptors. *IEEE Trans. Biomed. Eng.*, 33(4):428–434, April 1986.

[177] A. Ercole. Attenuation in invasive blood pressure measurement systems. *Br. J. Anaesth.*, 96(5):560–562, May 2006.

[178] M. Slotani. Tolerance regions for a multivariate normal population. *Ann. Inst. Stat. Math.*, 16(1):135–153, December 1964.

[179] S.-B. Lee, H. Kim, Y.-T. Kim, *et al.* Artifact removal from neurophysiological signals: impact on intracranial and arterial pressure monitoring in traumatic brain injury. *J. Neurosurg.*, 132(6):1952–1960, May 2019.

[180] J. Yoon, D. Jarrett, and M. van der Schaar. Time-series generative adversarial networks. *Advances in Neural Information Processing Systems*, 33:5509–5519, December 2019.

[181] J. Yoon, L. N. Drumright, and M. van der Schaar. Anonymization through data synthesis using generative adversarial networks (ADS-GAN). *IEEE J Biomed Health Inform*, 24(8):2378–2388, August 2020.

[182] C. Esteban, S. L. Hyland, and G. Rätsch. Real-valued (medical) time series generation with recurrent conditional GANs. *arXiv*, June 2017. Preprint at `https://arxiv.org/abs/1706.02633`.

[183] A. M. Alaa, B. van Breugel, E. Saveliev, and M. van der Schaar. How faithful is your synthetic data? Sample-level metrics for evaluating and auditing generative models. *arXiv*, February 2021. Preprint at `https://arxiv.org/abs/2102.08921`.

[184] M. Alzantot, S. Chakraborty, and M. Srivastava. SenseGen: A deep learning architecture for synthetic sensor data generation. In *2017 IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 188–193, March 2017.

[185] T. Salimans, I. Goodfellow, W. Zaremba, *et al.* Improved techniques for training GANs. *arXiv*, June 2016. Preprint at `https://arxiv.org/abs/1606.03498`.

[186] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni. Modeling tabular data using conditional GAN. *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 32:7335–7345, December 2019.

[187] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. *Proceedings of the 34th International Conference on Machine Learning*, 70:214–223, July 2017.

[188] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf. Wasserstein Auto-Encoders. *arXiv*, November 2017. Preprint at `https://arxiv.org/abs/1711.01558`.

[189] M. S. M. Sajjadi, O. Bachem, M. Lucic, O. Bousquet, and S. Gelly. Assessing generative models via precision and recall. *arXiv*, May 2018. Preprint at `https://arxiv.org/abs/1806.00035`.

[190] M. F. Naeem, S. J. Oh, Y. Uh, Y. Choi, and J. Yoo. Reliable fidelity and diversity metrics for generative models. *arXiv*, February 2020. Preprint at `https://arxiv.org/abs/2002.09797`.

[191] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *J. Mach. Learn. Res.*, 13(25):723–773, March 2012.

[192] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. *arXiv*, June 2017. Preprint at `https://arxiv.org/abs/1706.08500`.

[193] L. Theis, A. van den Oord, and M. Bethge. A note on the evaluation of generative models. *arXiv*, November 2015. Preprint at `https://arxiv.org/abs/1511.01844`.

[194] D. Xu, S. Yuan, L. Zhang, and X. Wu. FairGAN: Fairness-aware generative adversarial networks. In *2018 IEEE International Conference on Big Data*, pages 570–575, December 2018.

[195] M. Zameshina, O. Teytaud, F. Teytaud, *et al.* Fairness in generative modeling. *arXiv*, October 2022. Preprint at `https://arxiv.org/abs/2210.03517`.

[196] T. Liu, A. J. Chan, B. van Breugel, and M. van der Schaar. Practical approaches for fair learning with multitype and multivariate sensitive attributes. *arXiv*, November 2022. Preprint at `https://arxiv.org/abs/2211.06138`.

[197] K. El Emam. Seven ways to evaluate the utility of synthetic data. *IEEE Secur. Priv.*, 18(4):56–59, July 2020.

[198] M. Hittmeir, A. Ekelhart, and R. Mayer. On the utility of synthetic data: An empirical evaluation on machine learning tasks. In *Proceedings of the 14th International Conference on Availability, Reliability and Security*, number 29 in Ares '19, pages 1–6. Association for Computing Machinery, August 2019.

[199] Z. Lin, A. Jain, C. Wang, G. Fanti, and V. Sekar. Using GANs for sharing networked time series data: Challenges, initial promise, and open questions. *arXiv*, September 2019. Preprint at `https://arxiv.org/abs/1603.01155`.

[200] K. El Emam, L. Mosquera, and R. Hoptroff. *Practical Synthetic Data Generation: Balancing Privacy and the Broad Availability of Data.* O'Reilly Media, Inc., May 2020.

[201] Z. Azizi, C. Zheng, L. Mosquera, *et al.* Can synthetic data be a proxy for real clinical trial data? a validation study. *BMJ Open*, 11(4):e043497, April 2021.

[202] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. *Technical Report, Computer Science Laboratory, SRI International*, April 1998.

[203] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data*, 1(1):3–es, March 2007.

[204] K. El Emam, E. Jonker, L. Arbuckle, and B. Malin. A systematic review of re-identification attacks on health data. *PLoS One*, 6(12):e28071, December 2011.

[205] D. Chen, N. Yu, Y. Zhang, and M. Fritz. GAN-Leaks: A taxonomy of membership inference attacks against generative models. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 343–362. Association for Computing Machinery, November 2020.

[206] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. *arXiv*, October 2016. Published at `https://arxiv.org/abs/1610.05820`.

[207] G. Loukides, J. C. Denny, and B. Malin. The disclosure of diagnosis codes can breach research participants' privacy. *J. Am. Med. Inform. Assoc.*, 17(3):322–327, May 2010.

[208] C. Dwork. Differential privacy. In *Automata, Languages and Programming*, pages 1–12. Springer Berlin Heidelberg, 2006.

[209] V. Cheng, V. M. Suriyakumar, N. Dullerud, S. Joshi, and M. Ghassemi. Can you fake it until you make it? Impacts of differentially private synthetic data on downstream classification fairness. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 149–160. Association for Computing Machinery, March 2021.

[210] L. Xie, K. Lin, S. Wang, F. Wang, and J. Zhou. Differentially private generative adversarial network. *arXiv*, February 2018. Preprint at `https://arxiv.org/abs/1802.06739`.

[211] G. J. J. van Den Burg and C. K. I. Williams. On memorization in probabilistic deep generative models. In *Advances in Neural Information Processing Systems 34 proceedings (NeurIPS 2021)*. Neural Information Processing Systems, December 2021.

[212] B. Efron. Bootstrap methods: Another look at the jackknife. *Annals of Statistics*, 7(1):1–26, January 1979.

[213] A. Yale, S. Dash, R. Dutta, *et al.* Generation and evaluation of privacy preserving synthetic health data. *Neurocomputing*, 416:244–255, November 2020.

[214] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.* Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27:2672–2680, June 2014.

[215] J. Jordon, J. Yoon, and M. van der Schaar. PATE-GAN: Generating synthetic data with differential privacy guarantees. *International conference on learning representations*, September 2018.

[216] K. E. Rudd, S. C. Johnson, K. M. Agesa, *et al.* Global, regional, and national sepsis incidence and mortality, 1990-2017: analysis for the global burden of disease study. *Lancet*, 395(10219):200–211, January 2020.

[217] L. M. Fleuren, T. L. T. Klausch, C. L. Zwager, *et al.* Machine learning for the prediction of sepsis: a systematic review and meta-analysis of diagnostic test accuracy. *Intensive Care Med.*, 46(3):383–400, March 2020.

[218] J. L. Vincent, R. Moreno, J. Takala, *et al.* The SOFA (Sepsis-related Organ Failure Assessment) score to describe organ dysfunction/failure. on behalf of the working group on Sepsis-Related problems of the European Society of Intensive Care Medicine. *Intensive Care Med.*, 22(7):707–710, July 1996.

[219] M. Sartelli, Y. Kluger, L. Ansaloni, *et al.* Raising concerns about the Sepsis-3 definitions. *World J. Emerg. Surg.*, 13:6, January 2018.

[220] E. Siggiridou and D. Kugiumtzis. Granger causality in multivariate time series using a time-ordered restricted vector autoregressive model. *IEEE Trans. Signal Process.*, 64(7):1759–1773, April 2016.

[221] S. Guo, A. K. Seth, K. M. Kendrick, C. Zhou, and J. Feng. Partial Granger causality—eliminating exogenous inputs and latent variables. *Journal of Neuroscience Methods*, 172(1):79–93, April 2008.

[222] M. Stimberg. CODECHECK certificate 2021-001. `https://doi.org/10.5281/zenodo.4720843`, April 2021.

[223] R. Brunner, K. Ikegwu, J. Trauger, and T. Trauger. PyIF. `https://github.com/lcdm-uiuc/PyIF`, October 2019. Accessed on Sep 05, 2020.

[224] T. Edinburgh, A. Ercole, and S. J. Eglen. Bayesian model selection for multilevel models using integrated likelihoods: open-access code. `https://doi.org/10.5281/zenodo.7314381`, November 2022.

[225] T. Edinburgh, S. J. Eglen, P. Thoral, P. Elbers, and A. Ercole. Supporting data for "Sepsis-3 criteria in AmsterdamUMCdb: open-source code implementation", February 2022.

APPENDIX A

INFORMATION AND CAUSAL INFLUENCE

A.1 Information-theoretic indices and linear processes with Gaussian noise

The linear process with Gaussian noise is defined as:

$$\begin{aligned} x_{t+1} &= b_x x_t + \lambda y_t + \epsilon_{x,t}, \qquad & \epsilon_{x,t} &\sim N(0, \sigma_x^2) \\ y_{t+1} &= b_y y_t + \epsilon_{y,t}, & \epsilon_{y,t} &\sim N(0, \sigma_y^2) \end{aligned} \tag{A.1}$$

Closed-form analytic expressions for information-theoretic measures can be derived for X and Y from this system [67]. However, in [67], the authors only implicitly defined entropy, mutual information and transfer entropy in this system, so I have extended on their work to provide full analytic solutions here. Denoting the covariance matrix of vector $\boldsymbol{v} \in \mathbb{R}^m$ as C, with entries $C_{i,j}(\boldsymbol{v}) = c(v_i, v_j) = \mathbb{E}[v_i v_j] - \mathbb{E}[v_i]\,\mathbb{E}[v_j]$, the information-theoretic measures $H(X)$, $I(X,Y)$ and $\mathrm{TE}_{Y\to X}$ are:

$$H(X) = \frac{m}{2} + \frac{m}{2}\log(2\pi) + \frac{1}{2}\log\det C(\boldsymbol{x}_t) \tag{A.2}$$

$$I(X,Y) = \frac{1}{2}\log\frac{\det C(\boldsymbol{x}_t)\det C(\boldsymbol{y}_t)}{\det C(\boldsymbol{z}_t)} \tag{A.3}$$

$$\mathrm{TE}_{Y\to X} = \frac{1}{2}\log\frac{\det C\left(\begin{pmatrix}\boldsymbol{x}_t \\ x_{t+1}\end{pmatrix}\right)\det C(\boldsymbol{z}_t)}{\det C\left(\begin{pmatrix}\boldsymbol{z}_t \\ x_{t+1}\end{pmatrix}\right)\det C(\boldsymbol{x}_t)} \tag{A.4}$$

In Section 2.2, I estimated transfer entropy for $m = 1$ and $\tau = 1$. I have derived full algebraic expressions for this choice of hyperparameters here, but the same calculations can

be extended to other hyperparameter values. Defining $u = (1 - b_y^2)/\sigma_y^2,\ v = 1 - b_x^2, w = 1 - b_x b_y$, elements of the covariance matrices are as follows:

$$
\begin{aligned}
c(y_t, y_t) &= c(y_{t+1}, y_{t+1}) = c(b_y y_t + \epsilon_{y,t},\ b_y y_t + \epsilon_{y,t}) = b_y^2 c(y_t, y_t) + \sigma_y^2 = \frac{\sigma_y^2}{1 - b_y^2} = \frac{1}{u} \\
c(y_t, y_{t+1}) &= c(y_t, b_y y_t + \epsilon_{y,t}) = b_y c(y_t, y_t) = \frac{b_y \sigma_y^2}{1 - b_y^2} = \frac{b_y}{u} \\
c(x_t, y_t) &= c(x_{t+1}, y_{t+1}) = c(b_x x_t + \lambda y_t + \epsilon_{x,t},\ b_y y_t + \epsilon_{y,t}) = b_x b_y c(x_t, y_t) + \lambda b_y c(y_t, y_t) \\
&= \frac{1}{1 - b_x b_y} \lambda b_y c(y_t, y_t) = \frac{1}{1 - b_x b_y} \lambda b_y \frac{\sigma_y^2}{1 - b_y^2} = \frac{\lambda b_y}{uw} \\
c(x_{t+1}, y_t) &= c(b_x x_t + \lambda y_t + \epsilon_{x,t},\ y_t) = b_x c(x_t, y_t) + \lambda c(y_t, y_t) \\
&= b_x \frac{\lambda b_y}{1 - b_x b_y} \frac{\sigma_y^2}{1 - b_y^2} + \lambda \frac{\sigma_y^2}{1 - b_y^2} = \frac{\lambda}{uw} \\
c(x_t, y_{t+1}) &= c(x_t, b_y y_t + \epsilon_{y,t}) = b_y c(x_t, y_t) = b_y \frac{\lambda b_y}{1 - b_x b_y} \frac{\sigma_y^2}{1 - b_y^2} = \frac{\lambda b_y^2}{uw} \\
c(x_t, x_t) &= c(x_{t+1}, x_{t+1}) = c(b_x x_t + \lambda y_t + \epsilon_{x,t}, b_x x_t + \lambda y_t + \epsilon_{x,t}) \\
&= b_x^2 c(x_t, x_t) + 2\lambda b_x c(x_t, y_t) + \lambda^2 c(y_t, y_t) + \sigma_x^2 \\
&= \frac{1}{1 - b_x^2} \left(2\lambda b_x \frac{\lambda b_y}{1 - b_x b_y} \frac{\sigma_y^2}{1 - b_y^2} + \lambda^2 \frac{\sigma_y^2}{1 - b_y^2} + \sigma_x^2 \right) \\
&= \frac{1}{1 - b_x^2} \left(\sigma_x^2 + \frac{\lambda^2 (1 + b_x b_y) \sigma_y^2}{(1 - b_x b_y)(1 - b_y^2)} \right) = \frac{1}{uvw} (uw\sigma_x^2 + \lambda^2 (1 + b_x b_y)) \\
c(x_t, x_{t+1}) &= c(x_t, b_x x_t + \lambda y_t + \epsilon_{x,t}) = b_x c(x_t, x_t) + \lambda c(x_t, y_t) \\
&= \frac{b_x}{1 - b_x^2} \left(\sigma_x^2 + \frac{\lambda^2 (1 + b_x b_y) \sigma_y^2}{(1 - b_x b_y)(1 - b_y^2)} \right) + \frac{\lambda^2 b_y}{1 - b_x b_y} \frac{\sigma_y^2}{1 - b_y^2} \\
&= \frac{1}{1 - b_x^2} \left(b_x \sigma_x^2 + \frac{\lambda^2 (b_x + b_y) \sigma_y^2}{(1 - b_x b_y)(1 - b_y^2)} \right) = \frac{1}{uvw} (b_x uw\sigma_x^2 + \lambda^2 (b_x + b_y))
\end{aligned}
$$

These calculations hold under an implicit stationarity assumption. In reality, initial states x_0 and y_0 can have a small effect, but this becomes negligible if the time-series do not diverge and if a large number of early states (called transients) are discarded. For example, for $b_y < 1$:

$$
c(y_t, y_t) = \sum_{k=0}^{t} b_y^{2k} \sigma_y^2 = \frac{\sigma_y^2}{1 - b_y^2} + \sum_{k=t+1}^{\infty} b_y^{2k} \sigma_y^2 = \frac{\sigma_y^2}{1 - b_y^2} + R_{y,t}, \quad R_{y,t} = \sum_{k=t+1}^{\infty} b_y^{2k} \sigma_y^2 \to 0
$$

The full covariance determinants of the different subspaces are:

$$\begin{aligned}
\det C\left(\begin{pmatrix} x_t \\ x_{t+1} \end{pmatrix}\right) &= c(x_t, x_t)c(x_{t+1}, x_{t+1}) - c(x_t, x_{t+1})^2 \\
&= \frac{1}{(uvw)^2}\Big(u^2w^2\sigma_x^4(1-b_x^2) + 2\lambda^2\sigma_x^2uw(1 + b_xb_y - b_xb_y - b_x^2) \\
&\qquad + \lambda^4(1 + 2b_xb_y + b_x^2b_y^2 - b_x^2 - 2b_xb_y - b_y^2)\Big) \\
&= \frac{1}{uvw^2}(uw^2\sigma_x^4 + 2\lambda^2w\sigma_x^2 + \lambda^4\sigma_y^2) \\
\det C\left(\begin{pmatrix} y_t \\ y_{t+1} \end{pmatrix}\right) &= c(y_t, y_t)c(y_{t+1}, y_{t+1}) - c(y_t, y_{t+1})^2 \\
&= \frac{1}{u^2}(1 - b_y^2) = \sigma_y^2c(y_t, y_t) \\
\det C\left(\begin{pmatrix} x_t \\ y_t \end{pmatrix}\right) &= c(x_t, x_t)c(y_t, y_t) - c(x_t, y_t)^2 \\
&= \frac{1}{u^2vw^2}(uw^2\sigma_x^2 + \lambda^2w + \lambda^2wb_xb_y - \lambda^2vb_y^2) \\
&= \frac{1}{u^2vw^2}(uw^2\sigma_x^2 + \lambda^2(1 + b_xb_y)(1 - b_xb_y) - \lambda^2b_y^2(1 - b_x^2)) \\
&= \frac{1}{u^2vw^2}(uw^2\sigma_x^2 + \lambda^2(1 - b_y^2)) = \frac{1}{uvw^2}(w^2\sigma_x^2 + \lambda^2\sigma_y^2) \\
\det C\left(\begin{pmatrix} x_t \\ y_t \\ x_{t+1} \end{pmatrix}\right) &= c(x_t, x_t)c(y_t, y_t)c(x_{t+1}, x_{t+1}) + 2c(x_t, x_{t+1})c(x_{t+1}, y_t)c(x_t, y_t) \\
&\quad - c(x_t, x_{t+1})^2c(y_t, y_t) - c(x_t, x_t)c(x_{t+1}, y_t)^2 - c(x_{t+1}, x_{t+1})c(x_t, y_t)^2 \\
&= \frac{1}{u^3v^2w^3}\Big(w(uw\sigma_x^2 + \lambda^2(1 + b_xb_y))^2 + 2\lambda^2b_yv(b_xuw\sigma_x^2 + \lambda^2(b_x + b_y)) \\
&\qquad -\lambda^2b_y^2v(uw\sigma_x^2 + \lambda^2(1 + b_xb_y)) - w(b_xuw\sigma_x^2 + \lambda^2(b_x + b_y))^2 \\
&\qquad\qquad -\lambda^2v(uw\sigma_x^2 + \lambda^2(1 + b_xb_y))\Big) \\
&= \frac{1}{u^3v^2w^3}(u^2vw^3\sigma_x^4 + A\lambda^2uw\sigma_x^2 + B\lambda^4) \\
&= \frac{1}{u^3v^2w^3}(u^2vw^3\sigma_x^4 + \lambda^2u^2vw\sigma_x^2\sigma_y^2) = \frac{\sigma_x^2}{uvw^2}(w^2\sigma_x^2 + \lambda^2\sigma_y^2) \\
&= \sigma_x^2 \det C\left(\begin{pmatrix} x_t \\ y_t \end{pmatrix}\right)
\end{aligned}$$

where:

$$\begin{aligned}
A &= 2w(2 - w) + 2(1 - w)v - (1 - u\sigma_y^2)v - 2w(1 - v + 1 - w) - v = uv\sigma_y^2 \\
B &= w(1 - b_x^2)(1 - b_y^2) + v(-1 + b_xb_y)(1 - b_y^2) = uvw\sigma_y^2 - uvw\sigma_y^2 = 0
\end{aligned}$$

$$\begin{aligned}
\det C\left(\begin{pmatrix} x_t \\ y_t \\ y_{t+1} \end{pmatrix}\right) &= c(x_t,x_t)c(y_t,y_t)c(y_{t+1},y_{t+1}) + 2c(x_t,y_t)c(y_t,y_{t+1})c(x_t,y_{t+1}) \\
&\quad - c(x_t,y_{t+1})^2 c(y_t,y_t) - c(x_t,x_t)c(y_t,y_{t+1})^2 - c(y_{t+1},y_{t+1})c(x_t,y_t)^2 \\
&= \frac{1}{u^3w^2}(uw^2c(x_t,x_t) + 2\lambda^2 b_y^4 - \lambda^2 b_y^4 - b_y^2 uw^2 c(x_t,x_t) - \lambda^2 b_y^2) \\
&= \frac{1}{u^3w^2}(1-b_y^2)(uw^2c(x_t,x_t) - \lambda^2 b_y^2) = \frac{\sigma_y^2}{u}c(x_t,x_t) - \sigma_y^2\left(\frac{\lambda b_y}{uw}\right)^2 \\
&= \sigma_y^2(c(x_t,x_t)c(y_t,y_t) - c(x_t,y_t)^2) = \sigma_y^2 \det C\left(\begin{pmatrix} x_t \\ y_t \end{pmatrix}\right)
\end{aligned}$$

From this, entropy, mutual information and transfer entropy are defined as:

$$\begin{aligned}
H(Y) &= \frac{1}{2} + \frac{1}{2}\log 2\pi c(y_t,y_t) = \frac{1}{2} + \frac{1}{2}\log\frac{2\pi\sigma_y^2}{1-b_y^2} \\
H(X) &= \frac{1}{2} + \frac{1}{2}\log 2\pi c(x_t,x_t) = \frac{1}{2} + \frac{1}{2}\log\frac{2\pi}{1-b_x^2}\left(\sigma_x^2 + \frac{\lambda^2(1+b_xb_y)\sigma_y^2}{(1-b_xb_y)(1-b_y^2)}\right) \\
I(X,Y) &= \frac{1}{2}\log\frac{c(x_t,x_t)c(y_t,y_t)}{c(x_t,x_t)c(y_t,y_t) - c(x_t,y_t)^2} = \frac{1}{2}\log\frac{(uw^2\sigma_x^2 + \lambda^2(1-b_x^2b_y^2))/(u^2vw^2)}{(uw^2\sigma_x^2 + \lambda^2(1-b_y^2))/(u^2vw^2)} \\
&= \frac{1}{2}\log\frac{\sigma_x^2(1-b_y^2)(1-b_xb_y)^2 + \lambda^2\sigma_y^2(1-b_x^2b_y^2)}{\sigma_x^2(1-b_y^2)(1-b_xb_y)^2 + \lambda^2\sigma_y^2(1-b_y^2)} \\
\mathrm{TE}_{X\to Y} &= \frac{1}{2}\log\frac{\det C\big((x_t,y_t)^T\big)\det C\big((y_t,y_{t+1})^T\big)}{c(y_t,y_t)\det C\big((x_t,y_t,y_{t+1})^T\big)} = \frac{1}{2}\log\frac{\det C\big((x_t,y_t)^T\big)\sigma_y^2 c(y_t,y_t)}{c(y_t,y_t)\sigma_y^2\det C\big((x_t,y_t)^T\big)} \\
&= 0 \\
\mathrm{TE}_{Y\to X} &= \frac{1}{2}\log\frac{\det C\big((x_t,y_t)^T\big)\det C\big((x_t,x_{t+1})^T\big)}{c(x_t,x_t)\det C\big((x_t,y_t,x_{t+1})^T\big)} \\
&= \frac{1}{2}\log\frac{\det C\big((x_t,y_t)^T\big)\det C\big((x_t,x_{t+1})^T\big)}{c(x_t,x_t)\sigma_x^2\det C\big((x_t,y_t)^T\big)} = \frac{1}{2}\log\frac{\det C\big((x_t,x_{t+1})^T\big)}{\sigma_x^2 c(x_t,x_t)} \\
&= \frac{1}{2}\log\frac{uw^2\sigma_x^4 + 2\lambda^2 w\sigma_x^2 + \lambda^4\sigma_y^2}{uw^2\sigma_x^4 + \lambda^2 w(2-w)\sigma_x^2} \\
&= \frac{1}{2}\log\frac{\sigma_x^4(1-b_y^2)(1-b_xb_y)^2 + 2\lambda^2\sigma_x^2\sigma_y^2(1-b_xb_y) + \lambda^4\sigma_y^4}{\sigma_x^4(1-b_y^2)(1-b_xb_y)^2 + \lambda^2\sigma_x^2\sigma_y^2(1-b_x^2b_y^2)}
\end{aligned}$$

When λ is small, then both $I(X,Y)$ and $\mathrm{TE}_{Y\to X}$ are approximately quadratic in λ:

$$\begin{aligned}
I(X,Y) &= \frac{1}{2}\frac{\sigma_y^2(1-b_x^2b_y^2) - \sigma_y^2(1-b_y^2)}{\sigma_x^2(1-b_y^2)(1-b_xb_y)^2}\lambda^2 + O(\lambda^4) = \frac{\sigma_y^2}{2\sigma_x^2}\frac{b_y^2(1-b_x^2)}{(1-b_y^2)(1-b_xb_y)^2}\lambda^2 + O(\lambda^4) \\
\mathrm{TE}_{Y\to X} &= \frac{1}{2}\frac{2\sigma_x^2\sigma_y^2(1-b_xb_y) - \sigma_x^2\sigma_y^2(1-b_x^2b_y^2)}{\sigma_x^4(1-b_y^2)(1-b_xb_y)^2}\lambda^2 + O(\lambda^4) = \frac{\sigma_y^2}{2\sigma_x^2}\frac{1}{(1-b_y^2)}\lambda^2 + O(\lambda^4)
\end{aligned}$$

These covariance calculations can similarly be extended to arbitrary integer-valued lags τ:

$$
\begin{aligned}
c(y_t, y_{t+\tau}) &= b_y c(y_t, y_{t+\tau-1}) = b_y^2 c(y_t, y_{t+\tau-2}) = \frac{\sigma_y^2 b_y^\tau}{1-b_y^2} \\
c(x_t, y_{t+\tau}) &= b_y c(x_t, y_{t+\tau-1}) = b_y^\tau c(x_y, y_t) = \frac{\sigma_y^2 b_y^{\tau+1}\lambda}{(1-b_y^2)(1-b_x b_y)} \\
c(x_t, x_{t+\tau}) &= b_x c(x_t, x_{t+\tau-1}) + \lambda c(x_t, y_{t+\tau-1}) \\
&= \frac{\sigma_x^2 b_x^\tau}{1-b_x^2} + \frac{\lambda^2\sigma_y^2}{(1-b_x^2)(1-b_y^2)(1-b_x b_y)}\Big((1-b_x^2)\sum_{k=0}^{\tau} b_y^k b_x^{\tau-k} + b_x^{\tau+1}(b_x+b_y)\Big) \\
c(y_t, x_{t+\tau}) &= b_x c(y_t, x_{t+\tau-1}) + \lambda c(y_t, y_{t+\tau-1}) + c(y_t, \epsilon_{x,t+\tau-1}) \\
&= \frac{\lambda\sigma_y^2}{(1-b_y^2)(1-b_x b_y)}\Big(b_x^\tau b_y + (1-b_x b_y)\sum_{k=0}^{\tau-1} b_y^k b_x^{\tau-k}\Big)
\end{aligned}
$$

Clearly, a full expression for the coarse-grained transinformation rate (CTIR) is a much more complicated than transfer entropy. However, it is possible to construct covariance matrices which have elements that are one of the above four expressions, then numerically compute determinants in each case, before summing over a range of τ values to calculate CTIR.

A.2 Hyperparameters, computational cost and Ulam lattice figures

In Table A.1, I provide hyperparameter choices for each causal influence index, across different simulated systems. Table A.2 shows computation times for each causal influence index across all simulated model systems, averaged over 10 random initialisations of the simulated system for each value of λ. For practical reasons, some of the indices were combined because they shared internal computations. TE (H)/ETE (H) was the fastest in almost all cases, even though this calculation included 10 reshuffled computations for ETE (H). Several methods had extremely long computational times in UL simulations with $T = 10^5$, particularly CTIR and PI, but this was distorted by difficulties in computation when the system was in a synchronised state. I observed a marked difference in computational cost for NLGC when using k-means for clustering instead of fuzzy c-means, and this was one of the reasons that I argued in favour of using k-means for estimating NLGC. Figure A.1 shows correlations between methods and the additional tests relating to common data issues, for the Ulam lattice. These correlations are between of the 'baseline' $T = 10^3$ Ulam lattice system and the additional sensitivity analysis tests, for the net directed index $r_{X\to Y}$ values and all values of λ. Figures A.2 and A.3 show the results of these experiments in full (alongside Figure 2.4 in the main text).

Method	Parameter	LP	UL		HU			HB(I)	HB(NI)
	T	10^4	10^3	10^5	10^3	10^4	10^5	10^4	10^4
All ($*$)	m	2	1	1	2	2	2	2	2
EGC	L	20	100	100	100	100	100	100	100
	δ	0.8	0.5	0.2	0.5	0.3	0.2	0.6	0.6
NLGC	P	10	50	50	50	50	100	10	10
PI	R	10	1	1	1	1	1	1	1
	$\tau_{\max}$	20	5	5	5	5	5	5	5
$\text{SI}^{(1)}$	R	10	20	20	20	20	20	20	20
$\text{SI}^{(2)}$	R	30	20	20	20	20	20	100	100

Method	Parameter	All simulations
All	τ	1
NLGC	σ	0.05
PI	m	1
	h	1
TE (H)	m	1
	N	8
ETE (H)	S	10
	l	1
TE (KSG)	m	1
	k	4
CTIR	k	4
CCM	$T_{\max}$	T
	$n_{T'}$	40
	δ_ρ	0.05

Table A.1: Causal influence indices hyperparameter values for all simulations. I followed the hyperparameter values in [47]. The indices are as follows (where GC is Granger causality): extended GC (EGC), nonlinear GC (NLGC), predictability improvement (PI), transfer entropy (TE), effective transfer entropy (ETE), coarse-grained transinformation rate (CTIR), similarity indices (SI) and convergent cross mapping (CCM). TE (H) denotes estimation using a histogram binning partition, and TE (KSG) denotes Kraskov-Stögbauer-Grassberger estimation. The lower half of the table are parameters shared in all simulated systems. The embedding dimension was generally $m = 2$ for methods ($*$), except for the Ulam lattice or when otherwise specified for a particular method in the lower half of the table. CTIR is the only method that does not involve m or τ.

Method	LP	UL		HU			HB(I)	HB(NI)
	$T = 10^4$	10^3	10^5	10^3	10^4	10^5	10^4	10^4
EGC	0.193	0.131	4.095	0.070	0.227	1.216	0.250	0.291
NLGC	0.693	0.314	7.488	0.462	1.536	27.718	0.319	0.320
PI	0.555	0.047	21.323	0.051	0.535	6.166	0.535	0.675
ETE (H)	0.041	0.032	1.060	0.013	0.039	0.370	0.039	0.040
TE (KSG)	0.277	0.021	12.461	0.018	0.216	3.097	0.200	0.264
CTIR	8.729	0.181	140.876	0.156	1.860	25.541	1.714	2.317
$SI^{(1,2)}$	3.030	0.086	108.029	0.118	2.856	88.200	3.301	3.463
CCM	0.069	0.028	4.500	0.029	0.065	0.740	0.068	0.101

Table A.2: Computational requirements of each causal influence index on all simulated systems. The table reports the time in seconds for each computation, averaged over 10 independent runs and all values of λ. The most efficient method is highlighted. ETE (H) includes the computation of TE (H) as well. Both SI values were computed concurrently, so the computational time listed is for both indices combined. The computations were done in a high performance CPU computing cluster using SkyLake 6140 with 18 core 2.3GHz processors and 384GB of RAM.

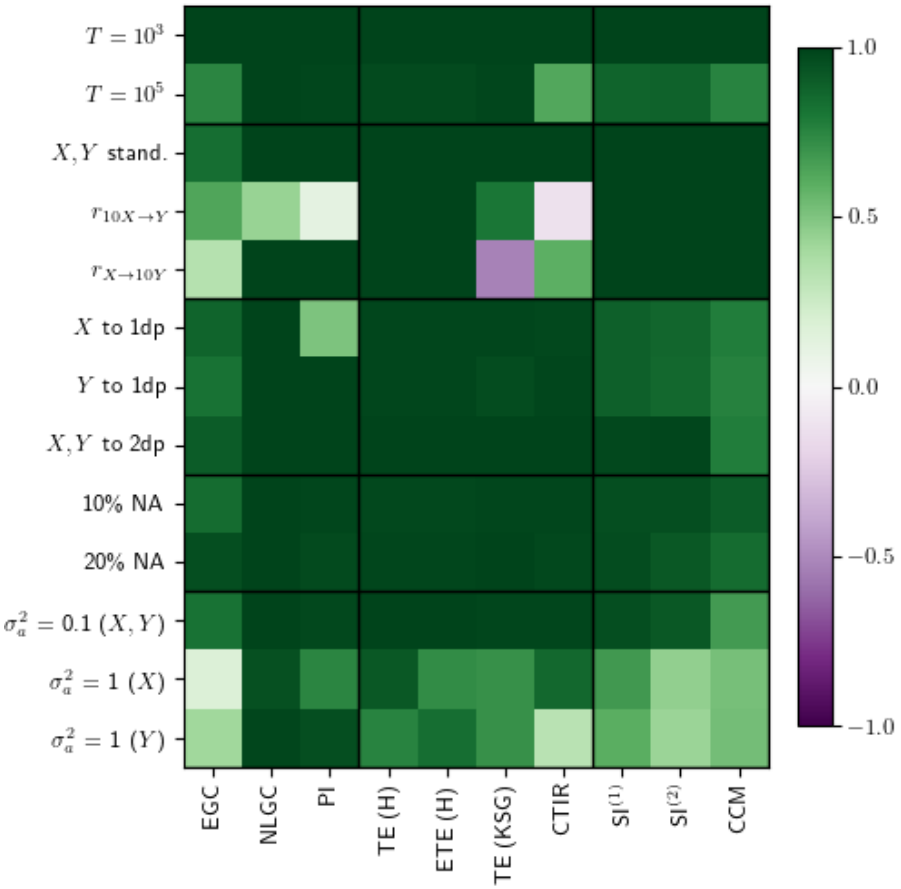

Figure A.1: Pearson correlations in $r_{X \to Y}$ values for data sensitivity tests. This is for the base case of Ulam lattice simulation with $T = 10^3$, and for the subsequent tests, which included the effects of data size, scaling, rounding error, missing data and Gaussian noise.

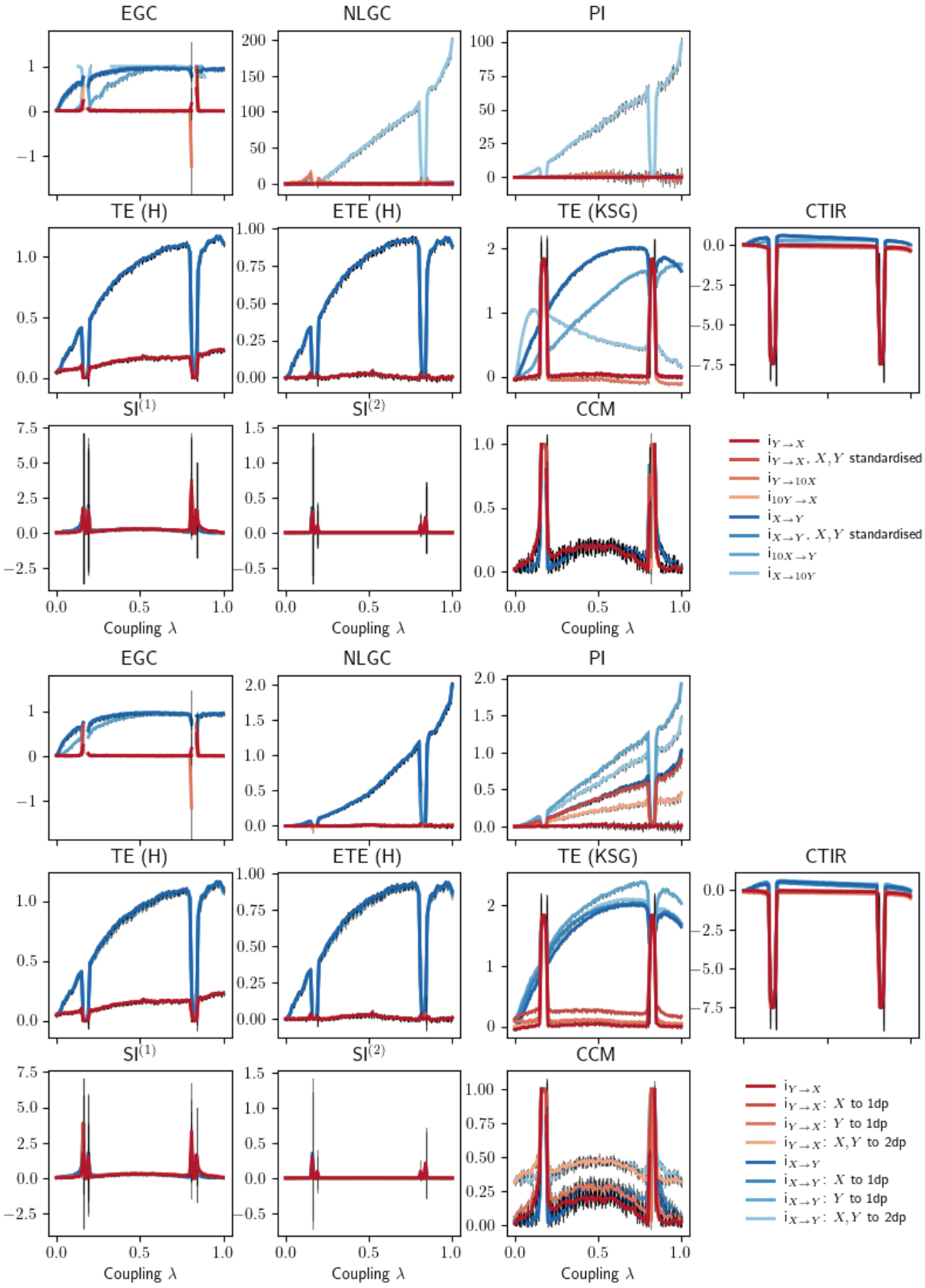

Figure A.2: Ulam lattice simulation results for data scaling (top) and rounding error (bottom). The Ulam lattice simulation had $T = 10^3$ data points and unidirectional ($X \to Y$) coupling. The Gaussian noise was added to the variables in brackets after simulation. Error bars are for one standard deviation from the mean values, after 10 independent Ulam lattice simulations.

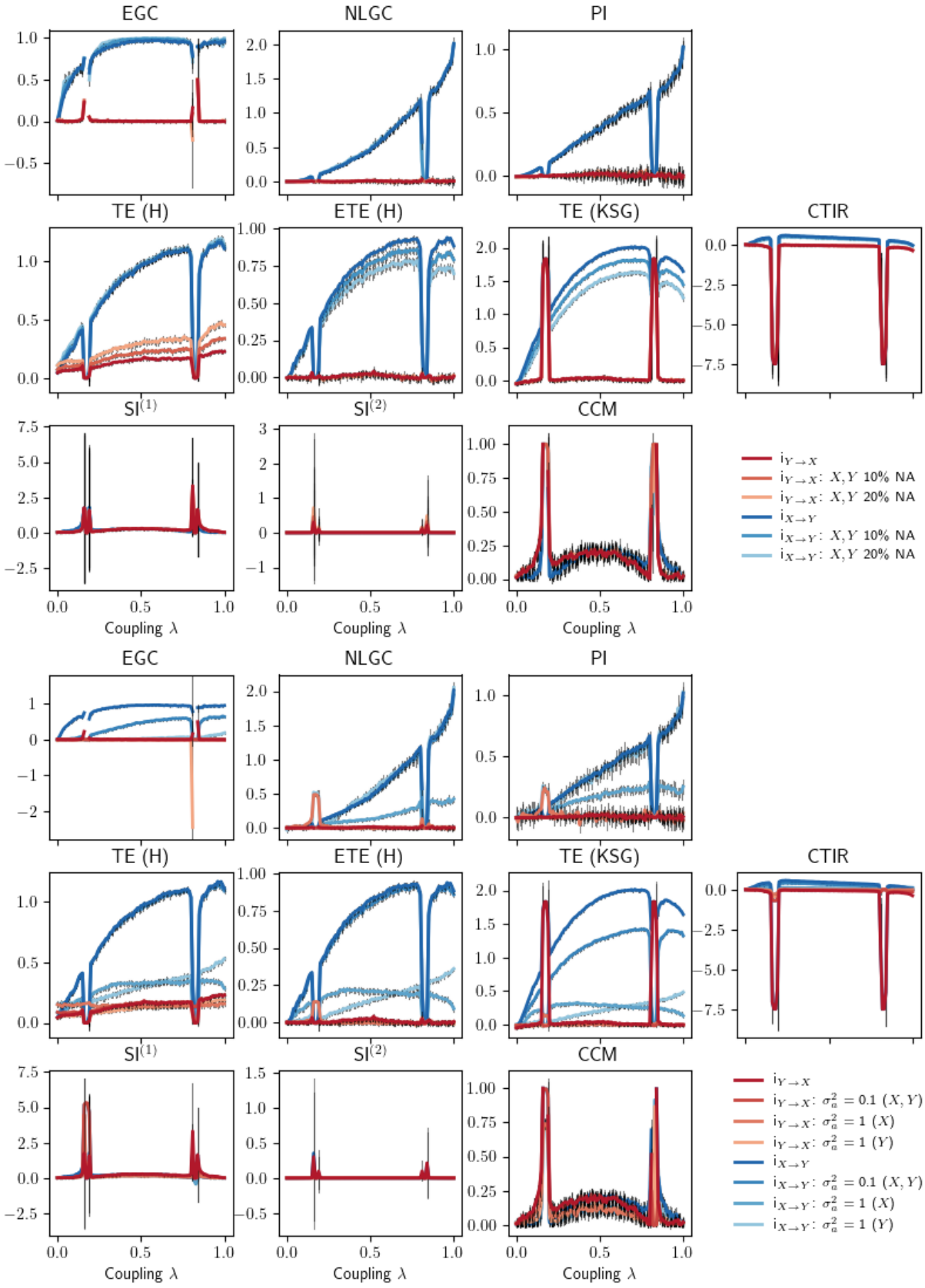

Figure A.3: Ulam lattice simulation results for missing data (top) and Gaussian noise (bottom). The Ulam lattice simulation had $T = 10^3$ data points and unidirectional $(X \to Y)$ coupling. Error bars are for one standard deviation from the mean values, after 10 independent Ulam lattice simulations.

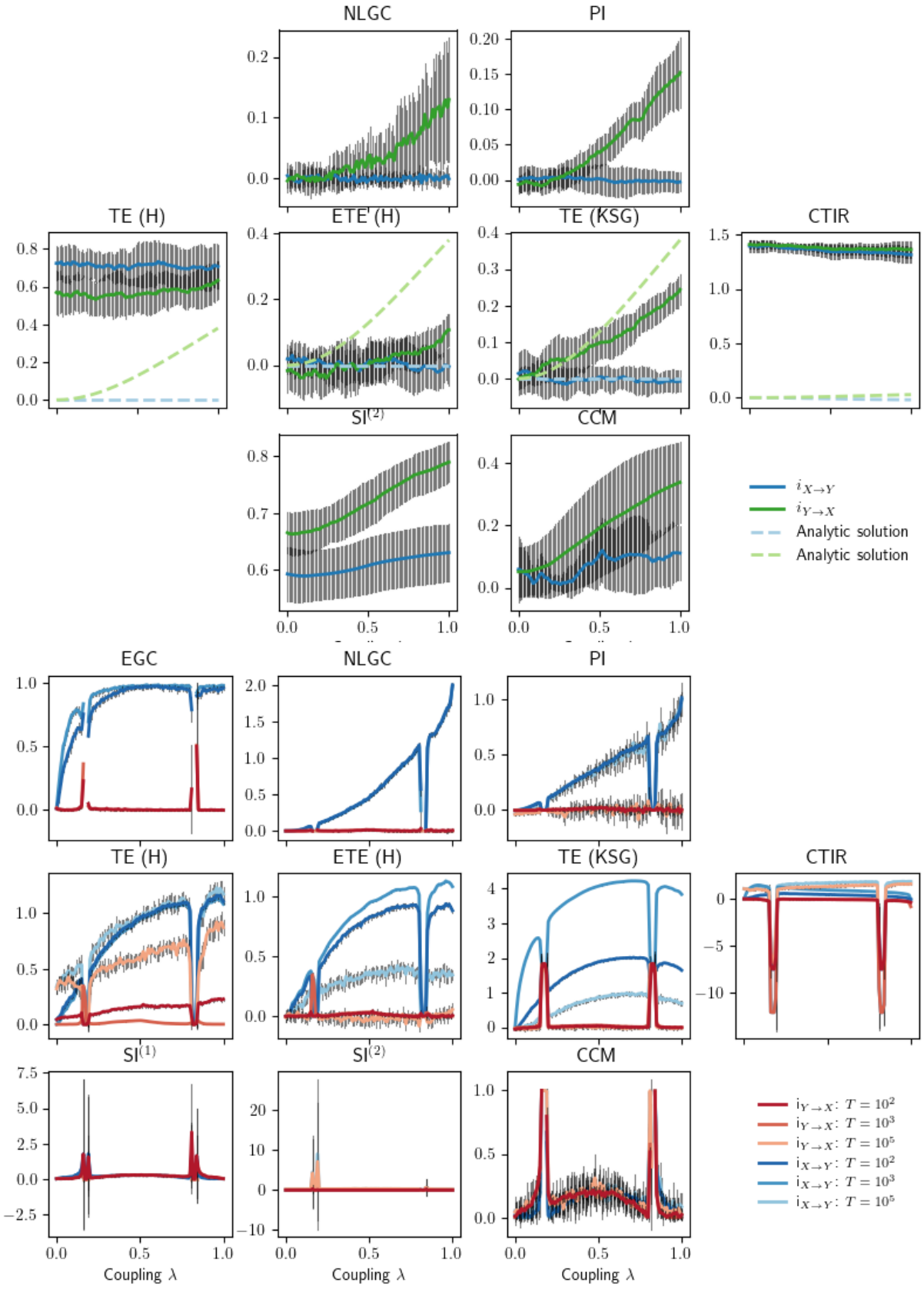

Figure A.4: Linear process and Ulam lattice simulation results for $T = 10^2$ data points. The Ulam lattice figure also includes results with $T = 10^3$ and $T = 10^3$ data points, to mirror Figure 2.4. In the linear process, two indices (EGC and $\mathrm{SI}^{(1)}$ returned NA values for all values of the coupling strength λ. Error bars are for one standard deviation from the mean values, after 10 independent simulations.

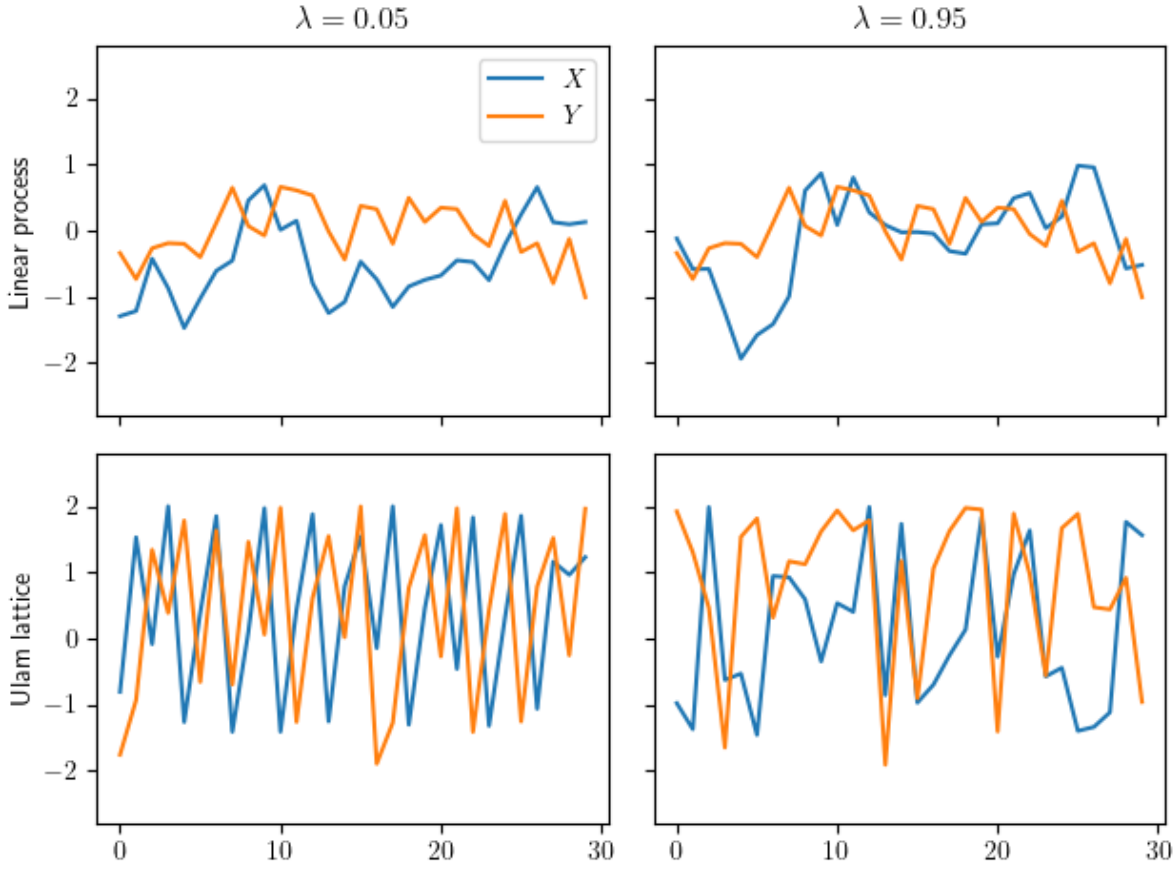

Figure A.5: Example transients from linear process and Ulam lattice experiments. These are sequences of length $T = 30$, taken after discarding initial $T = 10^4$ transients. For both experiments, low and high values of the coupling parameter λ are included. When $\lambda = 0.95$, there is coupling from Y to X in the linear process and from X to Y in the Ulam lattice.

APPENDIX B

MATHEMATICAL DETAILS FOR MULTILEVEL MODELS

B.1 Cyclic cubic regression spline for GAMs

The cyclic cubic regression spline defines the basis functions for a generalised additive model, as described in Section 3.1.3. The function $f(t) = \sum_{j=1}^{d} f_j(t)\beta_j$ is a piece-wise cubic polynomial, described in terms of (changepoint) knots τ_j, $j = 1, \ldots, d+1$. The coefficients β_j are the (unknown) value of the function $f(t)$ at these knots, i.e. $\beta_j = f(\tau_j)$. The first basis function $f_1(t)$ is:

$$\begin{aligned} f_1(t) = &\frac{(\tau_2 - t)}{h_j} \mathbb{1}\{\tau_1 \leq t \leq \tau_2\} + \frac{(t - \tau_d)}{h_d} \mathbb{1}\{\tau_d \leq t \leq \tau_{d+1}\} \\ &+ \sum_{k=1}^{d-1} \frac{(\tau_{k+1} - t)^3 - h_k^2(\tau_{k+1} - t)}{6h_k} \mathbb{1}\{\tau_k \leq t \leq \tau_{k+1}\} F_{k1} \\ &+ \sum_{k=2}^{d} \frac{(t - \tau_{k-1})^3 - h_{k-1}^2(t - \tau_{k-1})}{6h_{k-1}} \mathbb{1}\{\tau_{k-1} \leq t \leq \tau_k\} F_{k1} \end{aligned}$$

For $j = 2, \ldots, d$, the basis functions $f_j(t)$ are defined as:

$$\begin{aligned} f_j(t) = &\frac{(\tau_{j+1} - t)}{h_j} \mathbb{1}\{\tau_j \leq t \leq \tau_{j+1}\} + \frac{(t - \tau_{j-1})}{h_{j-1}} \mathbb{1}\{\tau_{j-1} \leq t \leq \tau_j\} \\ &+ \sum_{k=1}^{d-1} \frac{(\tau_{k+1} - t)^3 - h_k^2(\tau_{k+1} - t)}{6h_k} \mathbb{1}\{\tau_k \leq t \leq \tau_{k+1}\} F_{kj} \\ &+ \sum_{k=2}^{d} \frac{(t - \tau_{k-1})^3 - h_{k-1}^2(t - \tau_{k-1})}{6h_{k-1}} \mathbb{1}\{\tau_{k-1} \leq t \leq \tau_k\} F_{kj} \end{aligned}$$

where $h_k = (\tau_{k+1} - \tau_k)$, the $d \times d$ matrices F, B and D are defined as $F = B^{-1}D$:

$$B_{ij} = \frac{1}{6} \times \begin{cases} 2(h_1 + h_{d-1}) & i = j & i = 1 \\ 2(h_{i-1} + h_i) & i = j & i = 2, \ldots, d \\ h_{i-1} & i = j + 1 & i = 2, \ldots, d \\ h_i & i = j - 1 & i = 1, \ldots, d-1 \\ h_{d-1} & i = 1,\ j = d & \\ h_{d-1} & j = 1,\ i = d & \\ 0 & \text{otherwise} & \end{cases}$$

$$D_{ij} = \begin{cases} -1/h_1 - 1/h_{d-1} & i = j & i = 1 \\ -1/h_{i-1} - 1/h_i & i = j & i = 2, \ldots, d \\ 1/h_{i-1} & i = j + 1 & i = 2, \ldots, d \\ 1/h_i & i = j - 1 & i = 1, \ldots, d-1 \\ 1/h_{d-1} & i = 1,\ j = d & \\ 1/h_{d-1} & j = 1,\ i = d & \\ 0 & \text{otherwise} & \end{cases}$$

B and D are constructed such that $B\delta = D\beta$, where $\delta_j = f''(\tau_j)$ are the second derivatives at the knots. The cyclic condition means that $f(\tau_{d+1}) = f(\tau_1)$ and $f''(\tau_{d+1}) = f''(\tau_1)$. A smoothing penalty term, $\int_{\tau_1}^{\tau_d} f''(t)^2 dt$, can also be defined in terms of these matrices, i.e. $\int_{\tau_1}^{\tau_d} f''(t)^2 dt = \beta^T D^T B^{-1} D\beta = \beta^T S\beta$.

B.2 Integrated likelihoods for multilevel models

In Section 3.2, I presented the integrated likelihoods for multilevel (two-level and three-level) models. In this section, I provide detailed derivations of these results.

Two-level model. For the basic two-level linear model (Equation 3.2):

$$\begin{aligned} p(\mathcal{D}|\mathcal{M}_{21}, \sigma_y^2, \sigma_\eta^2) = \int_{\mathbb{R}^d} \int_{\mathbb{R}^n} & \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\beta - \mu)^T \Sigma^{-1} (\beta - \mu)\right) \times \\ & \prod_j \frac{1}{(2\pi\sigma_\eta^2)^{1/2}} \exp\left(-\frac{\eta_j^2}{2\sigma_\eta^2}\right) \times \\ & \prod_{i,j} \frac{1}{(2\pi\sigma_y^2)^{1/2}} \exp\left(-\frac{(y_{ij} - \beta^T x_{ij} - \eta_j)^2}{2\sigma_y^2}\right) d\eta d\beta \\ = \int_{\mathbb{R}^d} & \frac{1}{\sigma_y^m |\Sigma|^{1/2} (2\pi)^{(m+d)/2}} \exp\left(-\frac{1}{2}(\beta - \mu)^T \Sigma^{-1} (\beta - \mu)\right) \times \\ & \prod_j \left[\frac{1}{\sqrt{2\pi\sigma_\eta^2}} \int_{-\infty}^{\infty} \exp\left(-\frac{\eta_j^2}{2\sigma_\eta^2} - \frac{1}{2\sigma_y^2} \sum_i (y_{ij} - \beta^T x_{ij} - \eta_j)^2\right) d\eta_j \right] d\beta \end{aligned}$$

where $\prod_{i,j}(2\pi\sigma_y^2)^{-1/2} = (2\pi\sigma_y^2)^{-\sum_j n_j/2} = (2\pi\sigma_y^2)^{-n/2}$. Completing the square in η_j for the integrand in the square brackets:

$$\begin{aligned}\frac{\eta_j^2}{\sigma_\eta^2}+\frac{1}{\sigma_y^2}\sum_i(y_{ij}-\beta^T x_{ij}-\eta_j)^2 \;=\; & \frac{\sigma_y^2+n_j\sigma_\eta^2}{\sigma_y^2\sigma_\eta^2}\left(\eta_j-\frac{\sigma_\eta^2}{\sigma_y^2+n_j\sigma_\eta^2}\sum_i(y_{ij}-\beta^T x_{ij})\right)^2 \\ & +\frac{1}{\sigma_y^2}\sum_i(y_{ij}-\beta^T x_{ij})^2-\frac{1}{\sigma_y^2}\frac{\sigma_\eta^2}{\sigma_y^2+n_j\sigma_\eta^2}(\sum_i(y_{ij}-\beta^T x_{ij}))^2\end{aligned}$$

Then:

$$\begin{aligned}&\frac{1}{\sqrt{2\pi\sigma_\eta^2}}\int_{-\infty}^{\infty}\exp\left(-\frac{\eta_j^2}{2\sigma_\eta^2}-\frac{1}{2\sigma_y^2}\sum_i(y_{ij}-\beta^T x_{ij}-\eta_j)^2\right)d\eta_j \\ &\quad=\sqrt{\frac{\sigma_y^2}{\sigma_y^2+n_j\sigma_\eta^2}}\exp\left(-\frac{1}{2\sigma_y^2}\left(\sum_i(y_{ij}-\beta^T x_{ij})^2-\frac{\sigma_\eta^2}{\sigma_y^2+n_j\sigma_\eta^2}(\sum_i(y_{ij}-\beta^T x_{ij}))^2\right)\right)\end{aligned}$$

Rearranging for β:

$$\begin{aligned}&(\beta-\mu)^T\Sigma^{-1}(\beta-\mu)+\frac{1}{\sigma_y^2}\sum_{i,j}(y_{ij}-\beta^T x_{ij})^2-\frac{1}{\sigma_y^2}\sum_j\left(\frac{\sigma_\eta^2}{\sigma_y^2+n_j\sigma_\eta^2}(\sum_i(y_{ij}-\beta^T x_{ij}))^2\right) \\ &=(\beta-\hat\mu)^T\hat\Sigma^{-1}(\beta-\hat\mu)+\mu^T\Sigma^{-1}\mu+\frac{1}{\sigma_y^2}\sum_{i,j}y_{ij}^2-\frac{1}{\sigma_y^2}\sum_j\left(\frac{\sigma_\eta^2}{\sigma_y^2+n_j\sigma_\eta^2}(\sum_i y_{ij})^2\right)-\hat\mu^T\hat\Sigma^{-1}\hat\mu\end{aligned}$$

Finally, this gives the integrated likelihood

$$\begin{aligned}p(\mathcal{D}|\mathcal{M}_{21},\sigma_y^2,\sigma_\eta^2) = \; & \frac{|\hat\Sigma|^{1/2}}{(2\pi\sigma_y^2)^{m/2}|\Sigma|^{1/2}}\prod_j\left(\sqrt{\frac{\sigma_y^2}{\sigma_y^2+n_j\sigma_\eta^2}}\right)\times \\ & \exp\left(-\frac{1}{2}\left(\mu^T\Sigma^{-1}\mu+\frac{1}{\sigma_y^2}\sum_{i,j}y_{ij}^2-\hat\mu^T\hat\Sigma^{-1}\hat\mu\right.\right. \\ & \left.\left.-\frac{1}{\sigma_y^2}\sum_j\left(\frac{\sigma_\eta^2}{\sigma_y^2+n_j\sigma_\eta^2}(\sum_i y_{ij})^2\right)\right)\right)\end{aligned}$$

where the posterior mean and covariance are as defined in Section 3.2:

$$\begin{aligned}\hat\Sigma^{-1}(\sigma_y^2,\sigma_\eta^2) &= \Sigma^{-1}+\frac{1}{\sigma_y^2}\sum_{i,j}x_{ij}x_{ij}^T-\frac{1}{\sigma_y^2}\sum_j\left(\frac{\sigma_\eta^2}{\sigma_y^2+n_j\sigma_\eta^2}(\sum_i x_{ij})(\sum_k x_{kj}^T)\right) \\ \hat\mu(\sigma_y^2,\sigma_\eta^2) &= \hat\Sigma\left(\Sigma^{-1}\mu+\frac{1}{\sigma_y^2}\sum_{i,j}x_{ij}y_{ij}-\frac{1}{\sigma_y^2}\sum_j\left(\frac{\sigma_\eta^2}{\sigma_y^2+n_j\sigma_\eta^2}(\sum_i y_{ij})(\sum_k x_{kj})\right)\right)\end{aligned}$$

Two-level model with group-varying coefficients. In the more general case (Equation 3.4), the steps are very similar to the above:

$$
\begin{aligned}
p(\mathcal{D}|\mathcal{M}_{22}, \sigma_y^2, \phi) = \int_{\mathbb{R}^d} \int_{\mathbb{R}^{n \times d'}} & \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\beta-\mu)^T\Sigma^{-1}(\beta-\mu)\right) \times \\
& \prod_j \frac{1}{(2\pi)^{d'/2}|\Sigma_\eta(\phi)|^{1/2}} \exp\left(-\frac{1}{2}\eta_j^T\Sigma_\eta^{-1}(\phi)\eta_j\right) \times \\
& \prod_{i,j} \frac{1}{(2\pi\sigma_y^2)^{1/2}} \exp\left(-\frac{(y_{ij}-\beta^T x_{ij}-\eta_j^T z_{ij})^2}{2\sigma_y^2}\right) d\eta d\beta \\
= \int_{\mathbb{R}^{d+nd'}} & \frac{1}{\sigma_y^m|\Sigma|^{1/2}|\Sigma_\eta(\phi)|^{n/2}(2\pi)^{(m+d+nd')/2}} \exp\left(-\frac{1}{2}(\beta-\mu)^T\Sigma^{-1}(\beta-\mu)\right) \times \\
& \exp\left(-\frac{1}{2}\sum_j \eta_j^T\Sigma_\eta^{-1}(\nu)\eta_j - \frac{1}{2\sigma_y^2}\sum_i (y_{ij}-\beta^T x_{ij}-\eta_j^T z_{ij})^2\right) d\eta d\beta
\end{aligned}
$$

Then:

$$
\begin{aligned}
& (\beta-\mu)^T\Sigma^{-1}(\beta-\mu) + \sum_j \eta_j^T\Sigma_\eta^{-1}\eta_j + \frac{1}{\sigma_y^2}\sum_{i,j}(y_{ij}-\beta^T x_{ij}-\eta_j^T z_{ij})^2 \\
& = (\beta-\mu)^T\Sigma^{-1}(\beta-\mu) + \sum_j \left(\eta_j^T\left(\Sigma_\eta^{-1} + \frac{1}{\sigma_y^2}\sum_i z_{ij}z_{ij}^T\right)\eta_j - \frac{1}{\sigma_y^2}\eta_j^T\left(\sum_i z_{ij}(y_{ij}-\beta^T x_{ij})\right)\right. \\
& \qquad \left. - \frac{1}{\sigma_y^2}\left(\sum_i (y_{ij}-\beta^T x_{ij})z_{ij}^T\right)\eta_j + \frac{1}{\sigma_y^2}\sum_i (y_{ij}-\beta^T x_{ij})^2\right) \\
& = (\beta-\mu)^T\Sigma^{-1}(\beta-\mu) + \sum_j \left((\eta_j-\hat{\mu}_{\eta,j})^T\hat{\Sigma}_{\eta,j}^{-1}(\eta_j-\hat{\mu}_{\eta,j}) \frac{1}{\sigma_y^2}\sum_i (y_{ij}-\beta^T x_{ij})^2 - \hat{\mu}_{\eta,j}^T\hat{\Sigma}_{\eta,j}^{-1}\hat{\mu}_{\eta,j}\right) \\
& = (\beta-\hat{\mu})^T\hat{\Sigma}^{-1}(\beta-\hat{\mu}) + \sum_j \left((\eta_j-\hat{\mu}_{\eta,j})^T\hat{\Sigma}_{\eta,j}^{-1}(\eta_j-\hat{\mu}_{\eta,j})\right) \\
& \quad + \frac{1}{\sigma_y^2}\sum_{i,j} y_{ij}^2 - \frac{1}{\sigma_y^4}\sum_j \left((\sum_i z_{ij}^T y_{ij})\hat{\Sigma}_{\eta,j}(\sum_k z_{kj}y_{kj})\right) + \mu^T\Sigma^{-1}\mu - \hat{\mu}^T\hat{\Sigma}^{-1}\hat{\mu}
\end{aligned}
$$

with the posterior means and covariances:

$$
\begin{aligned}
\hat{\Sigma}_{\eta,j}^{-1} &= \Sigma_\eta^{-1}(\nu) + \frac{1}{\sigma_y^2}\sum_i z_{ij}z_{ij}^T, \qquad \hat{\mu}_{\eta,j} = \hat{\Sigma}_{\eta,j}\left(\frac{1}{\sigma_y^2}\sum_i z_{ij}(y_{ij}-\beta^T x_{ij})\right) \\
\hat{\Sigma}^{-1} &= \Sigma^{-1} + \frac{1}{\sigma_y^2}\sum_{i,j} x_{ij}x_{ij}^T - \frac{1}{\sigma_y^4}\sum_j \left((\sum_i x_{ij}z_{ij}^T)\hat{\Sigma}_{\eta,j}(\sum_k z_{kj}x_{kj}^T)\right) \\
\hat{\mu} &= \hat{\Sigma}\left(\Sigma^{-1}\mu + \frac{1}{\sigma_y^2}\sum_{i,j} x_{ij}y_{ij} - \frac{1}{\sigma_y^4}\sum_j \left((\sum_i x_{ij}z_{ij}^T)\hat{\Sigma}_{\eta,j}(\sum_k z_{kj}y_{kj})\right)\right)
\end{aligned}
$$

This integrated likelihood is:

$$p(\mathcal{D}|\mathcal{M}_{22},\sigma_y^2,\phi) = \frac{|\hat{\Sigma}|^{1/2}}{(2\pi\sigma_y^2)^{m/2}|\Sigma|^{1/2}|\Sigma_\eta|^{n/2}}\prod_j |\hat{\Sigma}_{\eta,j}|^{1/2} \times \exp\Bigg(-\frac{1}{2}\Bigg(\mu^T\Sigma^{-1}\mu + \frac{1}{\sigma_y^2}\sum_{i,j} y_{ij}^2 - \hat{\mu}^T\hat{\Sigma}^{-1}\hat{\mu} - \frac{1}{\sigma_y^4}\sum_j\Bigg((\sum_i z_{ij}^T y_{ij})\hat{\Sigma}_{\eta,j}(\sum_k z_{kj}y_{kj})\Bigg)\Bigg)\Bigg)$$

Three-level hierarchical structure. Finally, for the three-level hierarchical structure (Equation 3.5):

$$\begin{aligned} p(\mathcal{D}|\mathcal{M}_{31},\sigma_y^2,\sigma_\eta^2,\sigma_\zeta^2) &= \int_{\mathbb{R}^{d+m+K}} \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}}\exp\Bigg(-\frac{1}{2}(\beta-\mu)^T\Sigma^{-1}(\beta-\mu)\Bigg) \times \\ &\quad \prod_k \frac{1}{(2\pi\sigma_\zeta^2)^{1/2}}\exp\Bigg(-\frac{\zeta_k^2}{2\sigma_\zeta^2}\Bigg)\prod_{j,k}\frac{1}{(2\pi\sigma_\eta^2)^{1/2}}\exp\Bigg(-\frac{\eta_{jk}^2}{2\sigma_\eta^2}\Bigg) \times \\ &\quad \prod_{i,j,k}\frac{1}{(2\pi\sigma_y^2)^{1/2}}\exp\Bigg(-\frac{(y_{ijk}-\beta^T x_{ijk}-\eta_{jk}-\zeta_k)^2}{2\sigma_y^2}\Bigg)d\zeta d\eta d\beta \\ &= \int_{\mathcal{R}^{m+d+K}} \frac{1}{\sigma_y^m\sigma_\eta^n\sigma_\zeta^K|\Sigma|^{1/2}(2\pi)^{(n+d+m+K)/2}}\exp\Bigg(-\frac{1}{2}Z\Bigg)d\zeta d\eta d\beta \end{aligned}$$

where:

$$Z = (\beta-\mu)^T\Sigma^{-1}(\beta-\mu) + \sum_k\Bigg[\frac{1}{\sigma_\zeta^2}\zeta_k^2 + \frac{1}{\sigma_\eta^2}\sum_j\eta_{jk}^2 + \frac{1}{\sigma_y^2}\sum_{i,j}(y_{ijk}-\beta^T x_{ijk}-\eta_{jk}-\zeta_k)^2\Bigg]$$

The expression in square brackets is in the same form as the simple multilevel model:

$$\begin{aligned} \frac{1}{\sigma_\zeta^2}\zeta_k^2 + \frac{1}{\sigma_\eta^2}\sum_j\eta_{jk}^2 + \frac{1}{\sigma_y^2}\sum_{i,j}(a_{ijk}-\zeta_k)^2 &= \frac{\sigma_y^2+\sigma_\zeta^2 m_k}{\sigma_y^2\sigma_\zeta^2}\Bigg(\zeta_k - \frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2 m_k}\sum_{i,j}a_{ijk}\Bigg)^2 \\ &\quad + \frac{1}{\sigma_\eta^2}\sum_j\eta_{jk}^2 + \frac{1}{\sigma_y^2}\sum_{i,j}a_{ijk}^2 \\ &\quad - \frac{1}{\sigma_y^2}\frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2 m_k}(\sum_{i,j}a_{ijk})^2 \end{aligned}$$

where I have introduced the following for readability:

$$a_{ijk} = (y_{ijk}-\beta^T x_{ijk}-\eta_{jk}),\ b_{ijk} = (y_{ijk}-\beta^T x_{ijk})$$

For a given k, the last two lines of the previous expression are:

$$\begin{aligned}
&\frac{1}{\sigma_\eta^2}\sum_j \eta_{jk}^2 + \frac{1}{\sigma_y^2}\sum_{i,j}(b_{ijk}-\eta_{jk})^2 - \frac{1}{\sigma_y^2}\frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2 m_k}\left(\sum_{i,j}(b_{ijk}-\eta_{jk})\right)^2 \\
&= \frac{1}{\sigma_\eta^2}\sum_j \eta_{jk}^2 + \frac{1}{\sigma_y^2}\sum_j\left((\sum_i b_{ijk}^2) - 2\eta_{jk}(\sum_i b_{ijk}) + m_{jk}\eta_{jk}^2\right) \\
&\quad - \frac{1}{\sigma_y^2}\frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2 m_k}\left(\sum_j\left((\sum_i b_{ijk}) - m_{jk}\eta_{jk}\right)\right)^2 \\
&= \sum_{j,l}\eta_{jk}\eta_{lk}\left(\frac{1}{\sigma_\eta^2}\delta_{jl} + \frac{1}{\sigma_y^2}m_{jk}\delta_{jl} - \frac{1}{\sigma_y^2}\frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2 m_k}m_{jk}m_{lk}\right) + \frac{1}{\sigma_y^2}\sum_{i,j} b_{ijk}^2 \\
&\quad - 2\sum_j \eta_{jk}\left(\frac{1}{\sigma_y^2}(\sum_i b_{ijk}) - \frac{1}{\sigma_y^2}\frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2 m_k}(\sum_{i,l} b_{ilk})m_{jk}\right) - \frac{1}{\sigma_y^2}\frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2 m_k}(\sum_{i,j} b_{ijk})^2 \\
&= \eta_k^T\hat{\Sigma}_{\eta,k}^{-1}\eta_k - 2\eta_k^T\hat{\Sigma}_{\eta,k}^{-1}\hat{\mu}_{\eta,k} + \frac{1}{\sigma_y^2}\sum_{i,j} b_{ijk}^2 - \frac{1}{\sigma_y^2}\frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2 m_k}(\sum_{i,j} b_{ijk})^2 \\
&= (\eta_k-\hat{\mu}_{\eta,k})^T\hat{\Sigma}_{\eta,k}^{-1}(\eta_k-\hat{\mu}_{\eta,k}) - \hat{\mu}_{\eta,k}^T\hat{\Sigma}_{\eta,k}^{-1}\hat{\mu}_{\eta,k} + \frac{1}{\sigma_y^2}\sum_{i,j} b_{ijk}^2 - \frac{1}{\sigma_y^2}\frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2 m_k}(\sum_{i,j} b_{ijk})^2
\end{aligned}$$

where $\eta_k^T = (\eta_{1k},\ldots,\eta_{n_k k})$ and:

$$(\hat{\Sigma}_{\eta,k}^{-1})_{jl} = \underbrace{\frac{1}{\sigma_\eta^2}\delta_{jl} + \frac{1}{\sigma_y^2}m_{jk}\delta_{jl}}_{A_{jl}} - \underbrace{\frac{1}{\sigma_y^2}\frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2 m_k}m_{jk}m_{lk}}_{B_{jl}} = A_{jl} - B_{jl}$$

$$(\hat{\Sigma}_{\eta,k}^{-1}\hat{\mu}_{\eta,k})_l = \frac{1}{\sigma_y^2}\sum_i b_{ilk} - \frac{1}{\sigma_y^2}\frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2 m_k}(\sum_{i,j} b_{ijk})m_{lk}$$

For matrices A and B, where A and $A \pm B$ are invertible and B has rank 1 (as is the case with the outer product above), then the inverse of $A \pm B$ is:

$$A \pm B)^{-1} = A^{-1} \mp \frac{1}{1 \pm \mathrm{Tr}(BA^{-1})}A^{-1}BA^{-1}$$

In this case,

$$A_{jl} = \left(\frac{1}{\sigma_\eta^2} + \frac{1}{\sigma_y^2}m_{jk}\right)\delta_{jl},\ \ (A^{-1})_{jl} = \left(\frac{\sigma_\eta^2\sigma_y^2}{\sigma_y^2+\sigma_\eta^2 m_{jk}}\right)\delta_{jl}$$

$$\mathrm{Tr}(BA^{-1}) = \sum_{j,l}\left(\frac{1}{\sigma_y^2}\frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2 m_k}m_{jk}m_{lk}\right)\left(\frac{\sigma_\eta^2\sigma_y^2}{\sigma_y^2+\sigma_\eta^2 m_{jk}}\delta_{jl}\right) = \frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2 m_k}\sum_j \frac{\sigma_\eta^2}{\sigma_y^2+\sigma_\eta^2 m_{jk}}m_{jk}^2$$

Putting this together, we have the following:

$$(\hat{\Sigma}_{\eta,k})_{jl} = \frac{\sigma_y^2\sigma_\eta^2}{\sigma_y^2+\sigma_\eta^2 m_{jk}}\left(\delta_{jl} + \frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2\sum_p(\sigma_\eta^2 m_{pk}/(\sigma_y^2+\sigma_\eta^2 m_{pk}))}\frac{\sigma_\eta^2 m_{lk}}{\sigma_y^2+\sigma_\eta^2 m_{lk}} m_{jk}\right) \quad \text{(B.1)}$$

$$\begin{aligned}
(\hat{\mu}_{\eta,k})_j &= \sum_l (\hat{\Sigma}_{\eta,k})_{jl}(\hat{\Sigma}_{\eta,k}^{-1}\hat{\mu}_{\eta,k})_l \\
&= \sum_l \left(\frac{\sigma_y^2\sigma_\eta^2}{\sigma_y^2+\sigma_\eta^2 m_{jk}}\left(\delta_{jl} + \frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2\sum_p(\sigma_\eta^2 m_{pk}/(\sigma_y^2+\sigma_\eta^2 m_{pk}))}\frac{\sigma_\eta^2 m_{lk}}{\sigma_y^2+\sigma_\eta^2 m_{lk}} m_{jk}\right)\times \right. \\
&\qquad \left. \left(\frac{1}{\sigma_y^2}(\sum_i b_{ilk}) - \frac{1}{\sigma_y^2}\frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2 m_k}(\sum_{q,p} b_{qpk}) m_{lk}\right)\right) \\
&= \frac{\sigma_\eta^2}{\sigma_y^2+\sigma_\eta^2 m_{jk}}(\sum_i b_{ijk}) \; + \; \frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2\sum_l(\sigma_y^2 m_{lk}/(\sigma_y^2+\sigma_\eta^2 m_{lk}))}\times \\
&\qquad \frac{\sigma_\eta^2 m_{jk}}{\sigma_y^2+\sigma_\eta^2 m_{jk}}\left((\sum_q \frac{\sigma_\eta^2 m_{pk}}{\sigma_y^2+\sigma_\eta^2 m_{pk}}\sum_q b_{qpk}) - (\sum_{q,p} b_{qpk})\right) \\
&= \frac{\sigma_\eta^2}{\sigma_y^2+\sigma_\eta^2 m_{jk}}(\sum_i b_{ijk}) \; - \; \frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2\sum_l(\sigma_y^2 m_{lk}/(\sigma_y^2+\sigma_\eta^2 m_{lk}))}\times \qquad \text{(B.2)} \\
&\qquad (\sum_p \frac{\sigma_y^2}{\sigma_y^2+\sigma_\eta^2 m_{pk}}\sum_q b_{qpk})\frac{\sigma_\eta^2 m_{jk}}{\sigma_y^2+\sigma_\eta^2 m_{jk}}
\end{aligned}$$

$$\begin{aligned}
\hat{\mu}_{\eta,k}^T\hat{\Sigma}_{\eta,k}^{-1}\hat{\mu}_{\eta,k} &= \frac{1}{\sigma_y^2}\sum_j\left((\sum_i b_{ijk}) - \frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2 m_k}(\sum_{q,p} b_{qpk}) m_{jk}\right)(\hat{\Sigma}_{\eta,k}^{-1}\hat{\mu}_{\eta,k})_j \\
&= \frac{1}{\sigma_y^2}(\sum_j \frac{\sigma_\eta^2}{\sigma_y^2+\sigma_\eta^2 m_{jk}}(\sum_i b_{ijk})^2) - \frac{1}{\sigma_y^2}\frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2 m_k}(\sum_{q,p} b_{qpk})(\sum_j \frac{\sigma_\eta^2 m_{jk}}{\sigma_y^2+\sigma_\eta^2 m_{jk}}\sum_i b_{ijk}) \\
&\quad - \frac{1}{\sigma_y^2}\frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2\sum_l(\sigma_\eta^2 m_{lk}/(\sigma_y^2+\sigma_\eta^2 m_{lk}))}(\sum_p \frac{\sigma_y^2}{\sigma_y^2+\sigma_\eta^2 m_{pk}}\sum_q b_{qpk})\times \\
&\quad \left((\sum_j \frac{\sigma_\eta^2 m_{jk}}{\sigma_y^2+\sigma_\eta^2 m_{jk}}\sum_i b_{ijk}) - \frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2 m_k}(\sum_l \frac{\sigma_\eta^2 m_{lk}^2}{\sigma_y^2+\sigma_\eta^2 m_{lk}})(\sum_{i,j} b_{ijk})\right) \\
&= \frac{1}{\sigma_y^2}\sum_j \frac{\sigma_\eta^2}{\sigma_y^2+\sigma_\eta^2 m_{jk}}(\sum_i b_{ijk})^2 - \frac{1}{\sigma_y^2}\frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2 m_k}(\sum_{q,p} b_{qpk})^2 \\
&\quad + \frac{1}{\sigma_y^2}\frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2\sum_l(\sigma_y^2 m_{lk}/(\sigma_y^2+\sigma_\eta^2 m_{lk}))}(\sum_j \frac{\sigma_y^2}{\sigma_y^2+\sigma_\eta^2 m_{jk}}\sum_i b_{ijk})^2
\end{aligned}$$

Returning to the original integrand, the expression for Z becomes:

$$
\begin{aligned}
Z &= (\beta-\mu)^T\Sigma^{-1}(\beta-\mu) + \sum_k \left[\frac{\sigma_y^2+\sigma_\zeta^2 m_k}{\sigma_y^2\sigma_\zeta^2}\left(\zeta_k - \frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2 m_k}\sum_{i,j} a_{ijk}\right)^2 + (\eta_k-\hat{\mu}_{\eta,k})^T\hat{\Sigma}_{\eta,k}^{-1}(\eta_k-\hat{\mu}_{\eta,k})\right] \\
&\quad + \sum_k \left[\frac{1}{\sigma_y^2}\sum_{i,j} b_{ijk}^2 - \frac{1}{\sigma_y^2}\frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2 m_k}(\sum_{i,j} b_{ijk})^2 - \hat{\mu}_{\eta,k}^T\hat{\Sigma}_{\eta,k}^{-1}\hat{\mu}_{\eta,k}\right] \\
&= (\beta-\mu)^T\Sigma^{-1}(\beta-\mu) + \sum_k \left[\frac{\sigma_y^2+\sigma_\zeta^2 m_k}{\sigma_y^2\sigma_\zeta^2}\left(\zeta_k - \frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2 m_k}\sum_{i,j} a_{ijk}\right)^2 + (\eta_k-\hat{\mu}_{\eta,k})^T\hat{\Sigma}_{\eta,k}^{-1}(\eta_k-\hat{\mu}_{\eta,k})\right] \\
&\quad + \frac{1}{\sigma_y^2}\sum_k \Bigg[\sum_{i,j} b_{ijk}^2 - \sum_j \frac{\sigma_\eta^2}{\sigma_y^2+\sigma_\eta^2 m_{jk}}(\sum_i b_{ijk})^2 \\
&\qquad\qquad - \frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2\sum_l(\sigma_\eta^2 m_{lk}/(\sigma_y^2+\sigma_\eta^2 m_{lk}))}(\sum_j \frac{\sigma_y^2}{\sigma_y^2+\sigma_\eta^2 m_{jk}}\sum_i b_{ijk})^2\Bigg] \\
&= (\beta-\mu)^T\Sigma^{-1}(\beta-\mu) + \sum_k \left[\frac{\sigma_y^2+\sigma_\zeta^2 m_k}{\sigma_y^2\sigma_\zeta^2}\left(\zeta_k - \frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2 m_k}\sum_{i,j} a_{ijk}\right)^2 + (\eta_k-\hat{\mu}_{\eta,k})^T\hat{\Sigma}_{\eta,k}^{-1}(\eta_k-\hat{\mu}_{\eta,k})\right] \\
&\quad + \frac{1}{\sigma_y^2}\sum_k \Bigg[\sum_{i,j} (y_{ijk}-\beta^T x_{ijk})^2 - \sum_j \frac{\sigma_\eta^2}{\sigma_y^2+\sigma_\eta^2 m_{jk}}(\sum_i (y_{ijk}-\beta^T x_{ijk}))^2 \\
&\qquad\qquad - \frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2\sum_l(\sigma_\eta^2 m_{lk}/(\sigma_y^2+\sigma_\eta^2 m_{lk}))}(\sum_j \frac{\sigma_y^2}{\sigma_y^2+\sigma_\eta^2 m_{jk}}\sum_i (y_{ijk}-\beta^T x_{ijk}))^2\Bigg] \\
&= (\beta-\hat{\mu})^T\hat{\Sigma}^{-1}(\beta-\hat{\mu}) + \sum_k \left[\frac{\sigma_y^2+\sigma_\zeta^2 m_k}{\sigma_y^2\sigma_\zeta^2}\left(\zeta_k - \frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2 m_k}\sum_{i,j} a_{ijk}\right)^2 + (\eta_k-\hat{\mu}_{\eta,k})^T\hat{\Sigma}_{\eta,k}^{-1}(\eta_k-\hat{\mu}_{\eta,k})\right] \\
&\quad + \mu^T\Sigma^{-1}\mu - \hat{\mu}^T\hat{\Sigma}^{-1}\hat{\mu} + \frac{1}{\sigma_y^2}\sum_k \Bigg[\sum_{i,j} y_{ijk}^2 - \sum_j \frac{\sigma_\eta^2}{\sigma_y^2+\sigma_\eta^2 m_{jk}}(\sum_i y_{ijk})^2 \\
&\qquad\qquad - \frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2\sum_l(\sigma_\eta^2 m_{lk}/(\sigma_y^2+\sigma_\eta^2 m_{lk}))}(\sum_j \frac{\sigma_y^2}{\sigma_y^2+\sigma_\eta^2 m_{jk}}\sum_i y_{ijk})^2\Bigg]
\end{aligned}
$$

where the posterior mean and posterior covariance for β are:

$$
\begin{aligned}
\hat{\Sigma} = \Bigg(\Sigma^{-1} &+ \frac{1}{\sigma_y^2}\sum_{i,j,k} x_{ijk}x_{ijk}^T - \frac{1}{\sigma_y^2}\sum_{j,k}\left(\frac{\sigma_\eta^2}{\sigma_y^2+\sigma_\eta^2 m_{jk}}(\sum_i x_{ijk})(\sum_q x_{qjk}^T)\right) \\
&- \frac{1}{\sigma_y^2}\sum_k \frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2\sum_l(\sigma_\eta^2 m_{lk}/(\sigma_y^2+\sigma_\eta^2 m_{lk}))}\times \\
&\qquad (\sum_p \frac{\sigma_y^2}{\sigma_y^2+\sigma_\eta^2 m_{pk}}\sum_q x_{qpk})(\sum_j \frac{\sigma_y^2}{\sigma_y^2+\sigma_\eta^2 m_{jk}}\sum_i x_{ijk}^T)\Bigg)^{-1}
\end{aligned}
\tag{B.3}
$$

$$\hat{\mu} = \hat{\Sigma}\Bigg(\Sigma^{-1}\mu + \frac{1}{\sigma_y^2}\sum_{i,j,k} x_{ijk}y_{ijk} - \frac{1}{\sigma_y^2}\sum_{j,k}\Bigg(\frac{\sigma_\eta^2}{\sigma_y^2+\sigma_\eta^2 m_{jk}}(\sum_i x_{ijk})(\sum_q y_{qjk})\Bigg) \tag{B.4}$$
$$- \frac{1}{\sigma_y^2}\sum_k \frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2\sum_l(\sigma_\eta^2 m_{lk}/(\sigma_y^2+\sigma_\eta^2 m_{lk}))}\times$$
$$(\sum_p \frac{\sigma_y^2}{\sigma_y^2+\sigma_\eta^2 m_{lp}}\sum_q x_{qpk})(\sum_j \frac{\sigma_y^2}{\sigma_y^2+\sigma_\eta^2 m_{jk}}\sum_i y_{ijk})\Bigg)$$

The integrated likelihood is:

$$p(\mathcal{D}|\mathcal{M}_{31},\sigma_y^2,\sigma_\eta^2,\sigma_\zeta^2) = \frac{|\hat{\Sigma}|^{1/2}}{(2\pi\sigma_y^2)^{m/2}\sigma_\eta^n|\Sigma|^{1/2}}\prod_k\Bigg(\frac{1}{|\hat{\Sigma}_{\eta,k}^{-1}|^{1/2}}\sqrt{\frac{\sigma_y^2}{\sigma_y^2+\sigma_\zeta^2 m_k}}\Bigg)\times$$
$$\exp\Bigg(-\frac{1}{2}\Bigg(\mu^T\Sigma^{-1}\mu - \hat{\mu}^T\hat{\Sigma}^{-1}\hat{\mu}$$
$$+\frac{1}{\sigma_y^2}\sum_k\Bigg(\sum_{i,j} y_{ijk}^2 - \sum_j \frac{\sigma_\eta^2}{\sigma_y^2+\sigma_\eta^2 m_{jk}}(\sum_i y_{ijk})^2\Bigg)$$
$$-\frac{1}{\sigma_y^2}\sum_k \frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2\sum_l(\sigma_\eta^2 m_{lk}/(\sigma_y^2+\sigma_\eta^2 m_{lk}))}\times$$
$$(\sum_j \frac{\sigma_y^2}{\sigma_y^2+\sigma_\eta^2 m_{jk}}\sum_i y_{ijk})^2\Bigg)\Bigg)$$

This can be simplified, using the identity $|M \pm \lambda x x^T| = |M|(1 \pm \lambda x^T M^{-1} x)$, the determinant of $\hat{\Sigma}_{\eta,k}^{-1}$ is:

$$\begin{aligned}
|\hat{\Sigma}_{\eta,k}^{-1}| &= |A|\Bigg(1 - \frac{1}{\sigma_y^2}\frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2 m_k}m_k^T A^{-1} m_k\Bigg) \\
&= \Bigg(\prod_j \frac{\sigma_y^2+\sigma_\eta^2 m_{jk}}{\sigma_y^2\sigma_\eta^2}\Bigg)\Bigg(1 - \frac{1}{\sigma_y^2}\frac{\sigma_\zeta^2}{\sigma_y^2+\sigma_\zeta^2 m_k}\sum_{j,l} m_{jk}\frac{\sigma_y^2\sigma_\eta^2}{\sigma_y^2+\sigma_\eta^2 m_{jk}}\delta_{jl}m_{lk}\Bigg) \\
&= \frac{\sigma_y^2+\sigma_\zeta^2 m_k - \sigma_\zeta^2\sum_l(\sigma_\eta^2 m_{lk}^2/(\sigma_y^2+\sigma_\eta^2 m_{lk}))}{\sigma_y^2+\sigma_\zeta^2 m_k}\Bigg(\prod_j \frac{\sigma_y^2+\sigma_\eta^2 m_{jk}}{\sigma_y^2\sigma_\eta^2}\Bigg) \\
&= \frac{\sigma_y^2+\sigma_\zeta^2\sum_l(\sigma_\eta^2 m_{lk}/(\sigma_y^2+\sigma_\eta^2 m_{lk}))}{\sigma_y^2+\sigma_\zeta^2 m_k}\Bigg(\prod_j \frac{\sigma_y^2+\sigma_\eta^2 m_{jk}}{\sigma_y^2\sigma_\eta^2}\Bigg)
\end{aligned}$$

Then:

$$\frac{1}{\sigma_\eta^n}\prod_k\Bigg(\frac{1}{|\hat{\Sigma}_{\eta,k}^{-1}|^{1/2}}\sqrt{\frac{\sigma_y^2}{\sigma_y^2+\sigma_\zeta^2 m_k}}\Bigg) =$$
$$\prod_k\Bigg(\frac{\sigma_y^2}{\sigma_y^2+\sigma_\zeta^2\sum_l(\sigma_\eta^2 m_{lk}/(\sigma_y^2+\sigma_\eta^2 m_{lk}))}\Bigg)^{1/2}\times\prod_{j,k}\Bigg(\frac{\sigma_y^2}{\sigma_y^2+\sigma_\eta^2 m_{jk}}\Bigg)^{1/2}$$

Sampling from the posterior and the posterior predictive distribution. For the three-level model (Equation 3.5), the SMC trace (or another method) provides samples from the posterior distributions for the variance parameters, i.e. $\theta_{\sigma^2}^{(s)}$ for $s = 1, \ldots, S$, where $\theta_{\sigma^2} = (\sigma_y^2, \sigma_\eta^2, \sigma_\zeta^2)$. The posterior distributions are sampled upwards through the multilevel structure. This starts with $\beta^{(s)}|\theta_{\sigma^2}^{(s)} \sim N(\hat{\mu}, \hat{\Sigma})$, using Equations B.3 and B.4 (e.g. replacing variance terms with their samples from the posterior). Then, the second-level terms are sampled from posterior $\eta_k^{(s)}|\beta^{(s)}, \theta_{\sigma^2}^{(s)} \sim N(\hat{\mu}_{\eta,k}, \hat{\Sigma}_{\eta,k})$ using Equations B.1 and B.2. This is a multivariate Gaussian distribution for all second-level groups within a given third-level group, rather than separate univariate Gaussians for each second-level group independently of the higher-level structure. Next, the third-level terms are sampled from posterior $\zeta_k^{(s)}|\eta_k^{(s)}, \beta^{(s)}, \theta_{\sigma^2}^{(s)} \sim N(\hat{\mu}_{\zeta,k}, \hat{\sigma}_{\zeta,k}^2)$, where:

$$\hat{\mu}_{\zeta,k} = \frac{\sigma_\zeta^2}{\sigma_y^2 + \sigma_\zeta^2 m_k} \sum_{i,j} (y_{ijk} - \beta^T x_{ijk} - \eta_{jk})^2, \;\; \hat{\sigma}_{\zeta,k}^2 = \frac{\sigma_y^2 \sigma_\zeta^2}{\sigma_y^2 + \sigma_\zeta^2 m_k}$$

Finally, the replicated data from the posterior predictive distribution is:

$$y_{ijk}^{\text{rep}}|\zeta_k^{(s)}, \eta_k^{(s)}, \beta^{(s)}, \theta_{\sigma^2}^{(s)} \sim N(\beta^T x_{ijk} + \eta_{jk} + \zeta_k, \sigma_y^2)$$

This is repeated S times to give multiple instances of replicated data. In a posterior predictive check, the value of a test statistic $T(y)$ or test quantity $T(y, \theta)$ for the observed data is compared against the distribution of values of this test quantity for each of the different replicated data.

B.3 Comparing cohorts: priors and further results

In Section 3.3, I modelled inflammatory markers and information-theoretic variables separately as univariate functions of time, using both Bayesian and frequentist approaches. I did this firstly using two-level models (with patient groups) and then using three-level models (with patient and hospital groups). For each variable, I modelled three datasets: the Covid-19 cohort $\mathcal{D}_1$, the sepsis cohort $\mathcal{D}_2$ and the combined cohort $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2$. In each case, I converted the (normalised) timestamps t_{ijk} to covariates x_{ijk} using Equation 3.7, with $d_1 = 5$ fixed changepoints s_n (which were 0, 0.2, 0.4, 0.6 and 0.8). The Fourier terms had periodicity $P = 0.5$ and $d_2 = 4$. In the unnormalised timestamp data, this meant the longest Fourier period was 7.5 days and the shortest Fourier period was 1.875 days (or 45 hours). The full dimension of x_{ijk} was $d = 1 + d_1 + 2d_2 = 14$. The three-level Bayesian

model was of the form described in Equation 3.10, i.e.:

$$\begin{aligned}\mathcal{M}_{31}: \; & y_{ijk} = \beta^T x_{ijk} + \eta_{jk} + \zeta_k + \epsilon_{ijk}, \; \epsilon_{ijk} \sim N(0, \sigma_y^2), \; \eta_{jk} \sim N(0, \sigma_\eta^2), \; \zeta_k \sim N(0, \sigma_\zeta^2) \\ & \theta = (\beta^T, \sigma_y^2, \sigma_\eta^2, \sigma_\zeta^2)^T \sim \mathbb{P}_\theta\end{aligned}$$

The two-level Bayesian model was instead:

$$\begin{aligned}\mathcal{M}_{21}: \; & y_{ij} = \beta^T x_{ij} + \eta_j + \epsilon_{ij}, \; \epsilon_{ij} \sim N(0, \sigma_y^2), \; \eta_j \sim N(0, \sigma_\eta^2) \\ & \theta = (\beta^T, \sigma_y^2, \sigma_\eta^2)^T \sim \mathbb{P}_\theta\end{aligned}$$

In Section 3.3, I left details about the priors $\mathbb{P}_\theta$ to this Appendix. In each case, $\mathbb{P}_\theta$ was a composed of independent priors for each variable, with were either Gaussian (for regression coefficients β) or inverse-gamma (for each variance parameter). In all Bayesian models, the prior for β was $\beta \sim N(0, \Sigma)$, where Σ was a diagonal covariance matrix. The first $1 + d_1$ elements of the covariance diagonal, corresponding to the piece-wise linear component in Equation 3.7, had value 1. The remaining d_2 elements, corresponding to the Fourier term, had value 0.05. For the three-level models, the prior for σ_y^2 was $IG(3, 1)$ and the priors for σ_η^2 and σ_ζ^2 were $IG(3, 0.5)$. The two-level model had the same prior for σ_η^2, but an $IG(3, 1.5)$ prior for σ_y^2. Similarly to Section 3.2.3, this meant that any randomly sampled value y_{ijk} or y_{ik} from $\mathbb{P}_\theta$ should have similar expected value and variance. In Table B.1, the intercept-only three-level model was:

$$\begin{aligned}\mathcal{M}_{33}: \; & y_{ijk} = \beta + \eta_{jk} + \zeta_k + \epsilon_{ijk}, \; \epsilon_{ijk} \sim N(0, \sigma_y^2), \; \eta_{jk} \sim N(0, \sigma_\eta^2), \; \zeta_k \sim N(0, \sigma_\zeta^2) \\ & \beta \sim N(0, 1), \; \sigma_y^2 \sim IG(3, 1), \; \sigma_\eta^2 \sim IG(3, 0.5), \; \sigma_\zeta^2 \sim IG(3, 0.5),\end{aligned}$$

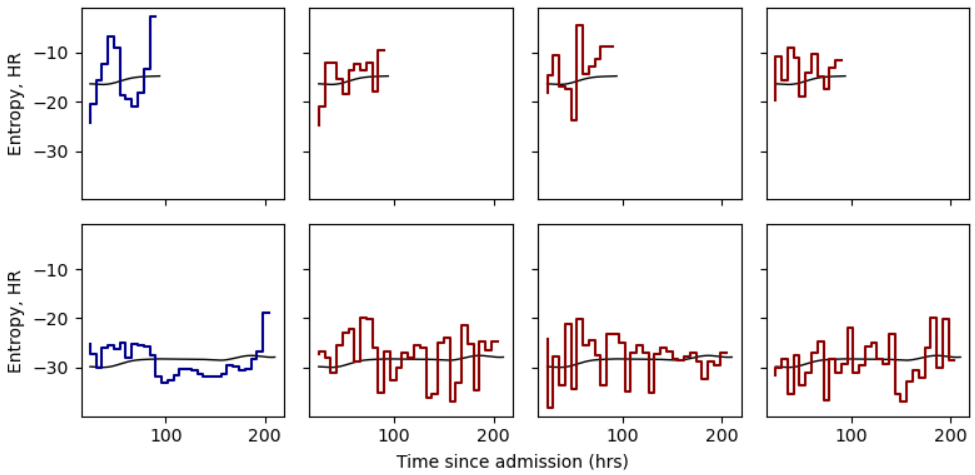

Figure B.1: Samples from the posterior predictive distribution. The left subplots (blue) show trajectories of HR entropy for two patients from the sepsis cohort (aligned at admission), with the Bayesian model fit. The remaining columns show replicated data sampled from the posterior predictive.

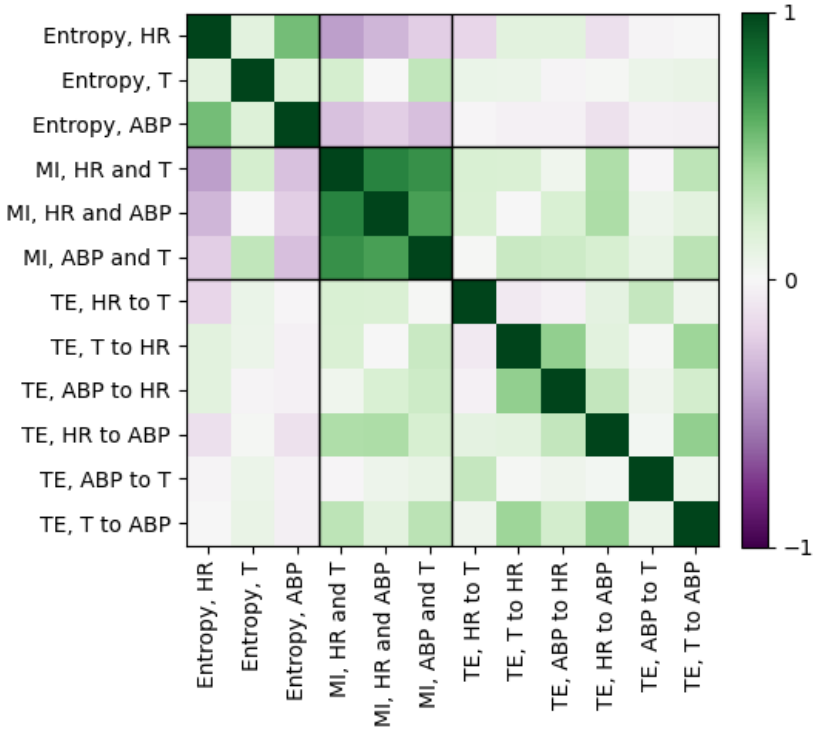

Figure B.2: Correlations between information-theoretic measures for Covid-19 and sepsis data. There was generally weak negative correlation between entropy values and the other information-theoretic measures, and generally weak positive correlation between mutual information values and transfer entropy values. The only strong correlations were between mutual information measures. The strongest negative correlation (-0.42) was between HR entropy values and T and HR mutual information values.

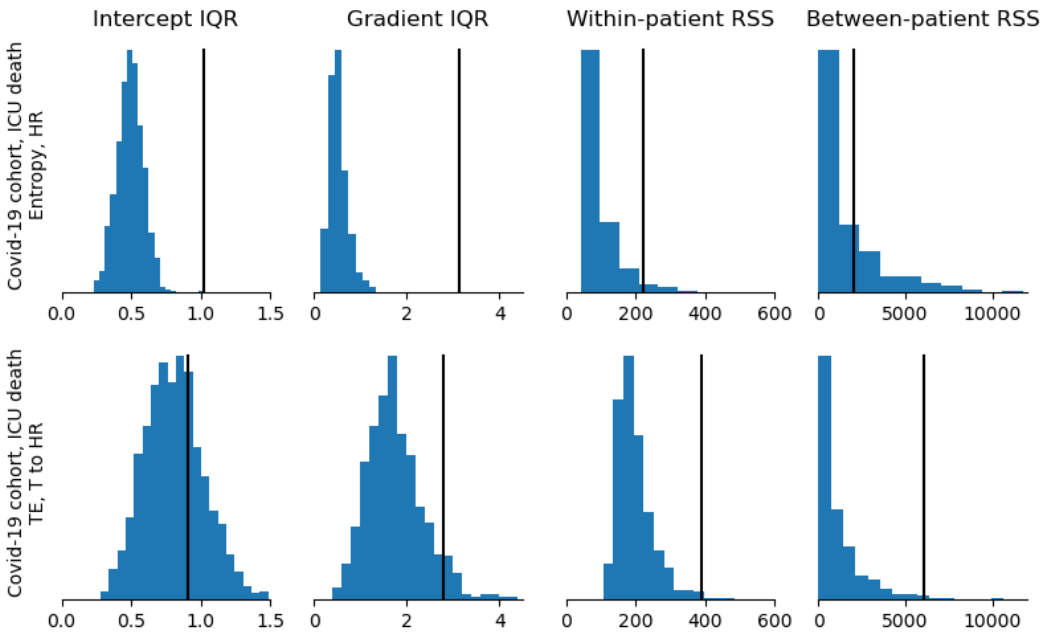

Figure B.3: A subset of posterior predictive checks. Four test statistics were defined in Section 3.3 for model checking: IQRs of patient-specific intercepts and gradients, within-patient RSS and between-patient RSS. The figure shows samples of the test statistic from the posterior predictive distribution, along with the observed value (vertical line).

Three-level models (with group-level patient and hospital terms)

	SMC with integrated likelihoods					
	Baseline model			Alternate model		
Variable	Full model $\log p(\mathcal{D}\|\mathcal{M}_b)$	Intercept-only $\log p(\mathcal{D}\|\mathcal{M}_{b^*})$	$\log_{10} \mathrm{BF}_{bb^*}$	Full model $\log p(\mathcal{D}\|\mathcal{M}_a)$	Intercept-only $\log p(\mathcal{D}\|\mathcal{M}_{a^*})$	$\log_{10} \mathrm{BF}_{aa^*}$
Entropy, T	-8761.78 (0.02)	-8854.32 (0.03)	40.19	-8715.44 (0.05)	-8835.01 (0.04)	51.93
Entropy, HR	-13325.86 (0.03)	-13480.03 (0.04)	66.96	-13221.09 (0.04)	-13429.24 (0.05)	90.40
Entropy, ABP	-14372.46 (0.04)	-15318.18 (0.02)	410.72	-14264.73 (0.04)	-15301.39 (0.04)	450.22
MI, HR and T	-9412.71 (0.04)	-9940.01 (0.04)	229.00	-9414.92 (0.05)	-9943.74 (0.03)	229.66
Transfer entropy, HR to T	-10104.68 (0.03)	-10107.32 (0.03)	1.15	-10102.14 (0.04)	-10107.06 (0.02)	2.14
Transfer entropy, T to HR	-10245.52 (0.02)	-10276.19 (0.03)	13.32	-10216.67 (0.04)	-10266.35 (0.03)	21.58
MI, ABP and T	9931.20 (0.03)	-10379.36 (0.03)	8820.76	-9928.21 (0.04)	-10380.52 (0.05)	196.44
Transfer entropy, ABP to T	-10025.14 (0.02)	-10022.93 (0.03)	-0.96	-10005.87 (0.03)	-10009.73 (0.03)	1.68
Transfer entropy, T to ABP	-10005.87 (0.03)	-10284.90 (0.02)	121.18	-10232.86 (0.03)	-10288.21 (0.05)	24.04
MI, ABP and HR	-13979.65 (0.03)	-14556.37 (0.04)	250.47	-13907.62 (0.06)	-14557.98 (0.03)	282.45
Transfer entropy, ABP to HR	-16323.07 (0.03)	-16429.87 (0.03)	46.38	-16317.1 (0.05)	-16421.03 (0.02)	45.14
Transfer entropy, HR to ABP	-16136.40 (0.04)	-16235.63 (0.03)	43.10	-16123.6 (0.06)	-16220.09 (0.06)	41.91
C-reactive protein (mg/l)	-3988.38 (0.04)	-4309.85 (0.03)	139.61	-3980.23 (0.03)	-4303.88 (0.03)	140.56
Leukocytes (10^9/l)	-4144.67 (0.03)	-4150.76 (0.02)	2.64	-3714.77 (0.04)	-3779.55 (0.03)	28.13

Table B.1: Bayesian model evidence comparisons of all of the information-theoretic and inflammatory marker variables between full model and intercept-only model. This involved three-level models, with second-level patient groups and third-level hospital groups. The Bayes factors was calculated using log model evidence estimates from sequential Monte Carlo (SMC) with integrated likelihoods (Equations 3.10, 3.11, 3.14, 3.16). These are colour-coded according to the interpretation table of Jeffrey [128], including substantial in favour of $\mathcal{M}_{b^*}$ ($\log_{10} \mathrm{BF}_{ab}$ between -1 and -0.5), strong in favour of $\mathcal{M}_b$ (between 1 and 1.5), very strong in favour of $\mathcal{M}_a$ (between 1.5 and 2) and decisive in favour of $\mathcal{M}_b$ or $\mathcal{M}_a$ (greater than 2).

APPENDIX C

DEEPCLEAN IMPLEMENTATION

This appendix contains supplementary material for the DeepClean artefact detection framework. Figure C.1 shows the ROC curves, as the automatic MSE threshold was varied. Table C.1 shows the DeepClean VAE network architecture.

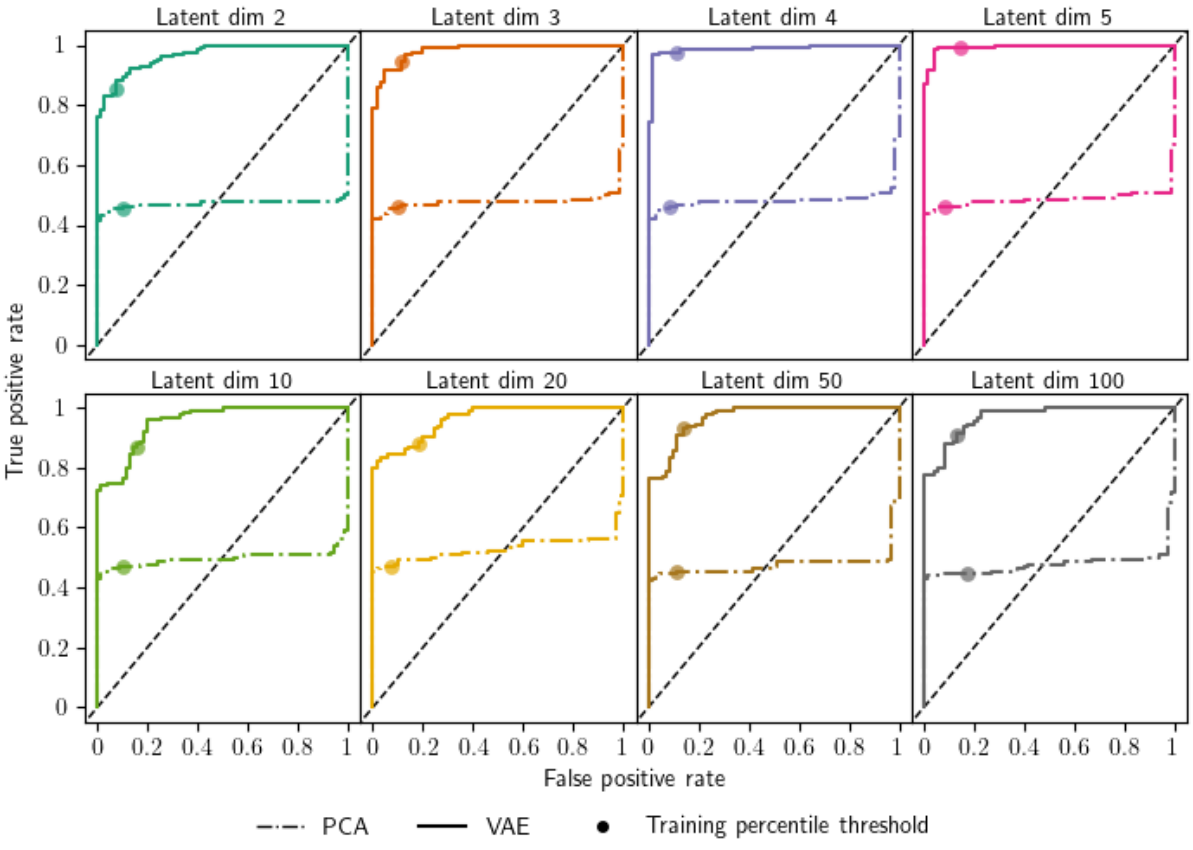

Figure C.1: Receiver operating characteristic (ROC) curves for artefact detection with DeepClean and PCA. True positive rate (TPR) and false positive rate (FPR) are the same as sensitivity and (1 - specificity) respectively. The points marked on each curve correspond to the automatic percentile threshold, for each latent dimension.

Encoder			Decoder		
Layer	Hyperparameters	Dimensions	Layer	Hyperparameters	Dimensions
Input x	None	1250×1	Input z	None	d_z
1-d convolutional	$n = 8,\ s_k = 15$	1250×8	Fully connected	$n = 16$	16
Max value pooling	$s_p = 5$	250×8	Fully connected	$n = 800$	800
1-d convolutional	$n = 16,\ s_k = 15$	250×16	Reshape	None	50×16
Max value pooling	$s_p = 5$	50×16	1-d convolutional	$n = 16,\ s_k = 15$	50×16
Dropout	$r = 0.1$	50×16	Upsampling	$s_u = 5$	250×16
1-d convolutional	$n = 16,\ s_k = 15$	50×16	Dropout	$r = 0.1$	250×16
Flatten	None	800	1-d convolutional	$n = 8,\ s_k = 15$	250×8
Dropout	$r = 0.1$	800	Upsampling	$s_u = 5$	1250×8
Fully connected	$n = 16$	16	Dropout	$r = 0.1$	1250×8
$2\times$ fully connected	$n = d_z$	$d_z,\ d_z$	1-d convolutional	$n = 1,\ s_k = 15$	1250×1
Outputs $\mu_z, \log(\sigma_z^2)$	None	$d_z,\ d_z$	Output μ_x	None	1250×1

Table C.1: DeepClean encoder and decoder network architecture, with tensor dimensions and hyperparameters. The inputs to each network are the first row. Each row after this describes a transformation applied to the one above. 1-dimensional convolutional layers have n convolutional filters with kernel size m. Pooling have pool size s_p and upsampling layers have repeat size s_u. Dropout layers have the dropout rate r, and dense fully connected (FC) layers have n units. The activation functions for every convolutional layer and FC layer were 'ReLu' functions. In all other cases, the activation was linear.

APPENDIX D

TIMEGAN ARCHITECTURE

The TimeGAN model [180] used in Chapter 5 consists of four neural networks, which act as the following: embedding, recovery, sequence generation and discriminator. In Chapter 5, I denoted the data as $x_i^r = (u_i, v_{i1}, \ldots, v_{iT})^T$, with time-independent (static) component $u_i \in \mathbb{R}_U$ and time-varying component $v_{i\tau} \in \mathbb{R}^V$ for $\tau = 1, \ldots, T$. In [180], the authors present the TimeGAN architecture with slightly different notation, using static features s and temporal features $x_{1:T} = (x_1, \ldots, x_T)^T$ in feature vector spaces $\mathcal{S}$ and $\mathcal{X}$ respectively (e.g. $\mathbb{R}^U$ and $\mathbb{R}^V$). These are associated with embedding vectors $h_\mathcal{S}$ and $h_{1:T}$, latent vectors $z_\mathcal{S}$ and $z_{1:T}$, and reconstructions $\tilde{s}$ and $\tilde{x}_{1:T}$. The corresponding vector spaces for the embedding vectors are $\mathcal{H}_\mathcal{S}$ and $\mathcal{H}_\mathcal{X}$ and for latent vectors are $\mathcal{Z}_\mathcal{S}$ and $\mathcal{Z}_\mathcal{X}$. Generally speaking, the four neural networks are functions acting on these vector spaces:

$$
\begin{aligned}
e &\colon \mathcal{S} \times \prod_t \mathcal{X} \to \mathcal{H}_\mathcal{S} \times \prod_t \mathcal{H}_\mathcal{X}, && s, x_{1:T} \mapsto h_\mathcal{S}, h_{1:T} = e(s, x_{1:T}) \\
r &\colon \mathcal{H}_\mathcal{S} \times \prod_t \mathcal{H}_\mathcal{X} \to \mathcal{S} \times \prod_t \mathcal{X}, && h_\mathcal{S}, h_{1:T} \mapsto \tilde{s}, \tilde{x}_{1:T} = r(h_\mathcal{S}, h_{1:T}) \\
g &\colon \mathcal{Z}_\mathcal{S} \times \prod_t \mathcal{Z}_\mathcal{X} \to \mathcal{H}_\mathcal{S} \times \prod_t \mathcal{H}_\mathcal{X}, && z_\mathcal{S}, z_{1:T} \mapsto \hat{h}_\mathcal{S}, \hat{h}_{1:T} = g(z_\mathcal{S}, z_{1:T}) \\
d &\colon \mathcal{H}_\mathcal{S} \times \prod_t \mathcal{H}_\mathcal{X} \to \prod_{t+1} [0, 1], && \tilde{h}_\mathcal{S}, \tilde{h}_{1:T} \mapsto \tilde{y}_\mathcal{S}, \tilde{y}_{1:T} = d(\tilde{h}_\mathcal{S}, \tilde{h}_{1:T})
\end{aligned}
$$

Here, the latent vector $z_\mathcal{S}, z_{1:T}$ is sampled from a known distribution (e.g. a Gaussian distribution for $z_\mathcal{S}$ and a Wiener process for $z_{1:T}$), the output of the sequence generating function g is a synthetic embedding $\hat{h}_\mathcal{S}, \hat{h}_{1:T}$ and the input to the discriminator function d is either the real embedding $h_\mathcal{S}, h_{1:T}$ or the synthetic embedding $\hat{h}_\mathcal{S}, \hat{h}_{1:T}$. Random sampling to determine the input to the discriminator is unseen by the model. The output of the discriminator function d is the usual GAN classification $\tilde{y}_\mathcal{S}, \tilde{y}_{1:T}$ (i.e. whether the model believes the current embedding vector $\tilde{h}_\mathcal{S}, \tilde{h}_{1:T}$ is real or synthetic). These functions can have different neural network architectures but must follow some autoregressive/causal

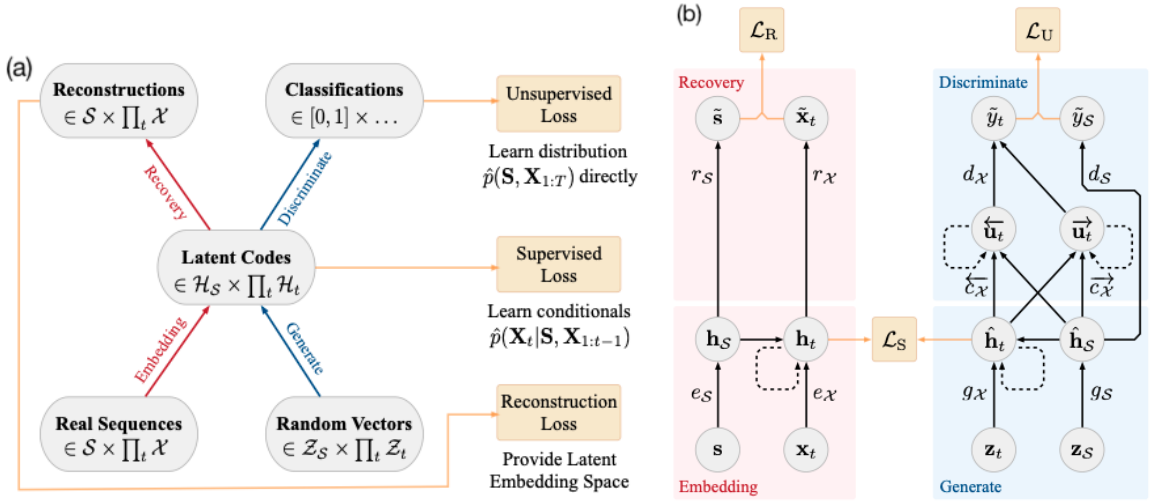

Figure D.1: TimeGAN architecture (from [180]). (a) Block diagram of networks and loss functions. (b) The default TimeGAN architecture using RNNs and feedforward networks.

ordering rules. The default TimeGAN architecture uses a combination of recurrent and feedforward networks (Figure D.1).

APPENDIX E

CODE AVAILABILITY

E.1 Causal influence indices for bivariate time-series

One goal of the reproducibility project in Section 2.2 was to make an open-access code resource for all the causal influence indices reviewed in this chapter. My code is openly available at the GitHub repository `https://github.com/tedinburgh/causality-review`, with permanent DOI at [90]. The data that supported the findings of this study are openly available at the same repository. I received a CODECHECK certificate (`https://codecheck.org.uk`) that confirmed all computations and figures in my paper [21] could be independently executed. This CODECHECK has permament DOI [222]. Existing open-access code for some causal influence indices include repositories for information theory and transfer entropy in IDTxl [91] v1.1 and PyIF [223], and for convergent cross mapping in pyEDM [92] v1.7.4. I checked results for transfer entropy and convergent cross mapping against the IDTxl and pyEDM repositories respectively. All code in our repository and in these others is Python.

E.2 Bayesian model selection for multilevel models

Open-source code and all datasets for Section 3.2 is available in the GitHub repository (1), with permanent DOI at [224]. The AmsterdamUMCdb dataset discussed in Section 3.3 is freely-accessible, with access upon request through (2). This is supported by a GitHub repository (3). The code for Section 3.3 is available on request, and will be made available open-access when this analysis is ready to submit as a publication.

(1) `https://github.com/tedinburgh/model-evidence-with-integrated-likelihood`
(2) `https://www.amsterdammedicaldatascience.nl`
(3) `https://github.com/AmsterdamUMC/AmsterdamUMCdb`

E.3 DeepClean artefact detection

The data used in this work were recorded as part of routine clinical care and not available for open access. Open-access code used in this chapter is available at `https://github.com/tedinburgh/deepclean`.

E.4 Synthetic medical time-series data

This chapter uses the TimeGAN generative model, which is supported at the GitHub repositories (1) and (2). The code for my analysis in this chapter is available on request, and will be made available open-access when this chapter is ready to submit as a publication. Section 5.4 is built on a project I lead to implement the Sepsis-3 criteria in real data from AmsterdamUMCdb, and then to describe the epidemiology. The first part of this project was published at [112], with supporting code available open-access at [225] and in the accompanying GitHub repository (3).

(1) `https://github.com/jsyoon0823/TimeGAN`
(2) `https://github.com/vanderschaarlab/synthcity`
(3) `https://github.com/tedinburgh/sepsis3-amsterdamumcdb`