On AI Frameworks for Healthcare Time-Series Analysis: Missingness, Dynamics and Representation

Linglong Qian

Supervisor: Dr. Zina Ibrahim

Prof. Richard Dobson

The Department of Biostatistics and Health Informatics

King's College London

This dissertation is submitted for the degree of

Doctor of Philosophy in Health Informatics

February 2025

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

I acknowledge the use of AI software to rewrite, rephrase and/or paraphrase parts of this thesis to ensure the quality and standard of the English used. I declare that my use of AI software is consistent with the research degree award criteria outlined in the Framework for Postgraduate Research Awards. This thesis is a genuine account of the research I have undertaken, and the content can still be considered my own words, with all references cited accordingly.

Linglong Qian

February 2025

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisors, Professor Richard Dobson and Dr Zina Ibrahim. From my MSc dissertation to my PhD, Zina has been an unwavering source of encouragement, guidance, and mentorship. Throughout these four years, I have faced numerous challenges—navigating the disruptions of COVID, struggling with limited computational resources, overcoming difficulties in data acquisition, and enduring countless late nights writing papers to meet conference deadlines. Even in moments of uncertainty, whether applying for scholarships or tackling research setbacks, she has always been there, offering her time, wisdom, and support. I am incredibly fortunate to have had such a dedicated and inspiring mentor.

Richard has fostered an environment that values collaboration, innovation, and real-world impact. His leadership has provided me with invaluable opportunities to work at the intersection of academia and healthcare, allowing me to contribute to meaningful research guided by domain experts. His insights and strategic thinking have profoundly shaped the direction of my work, and I am grateful for his continuous support.

I am also immensely grateful to Professor James Teo and Dr Hugh Logan Ellis, who played a crucial role in overcoming the challenges of hospital data access. His patience, persistence, and invaluable expertise have been instrumental in enabling the progress of my research.

A special thanks to my friends and frequent collaborators, Zeljko Kraljevic and Anthony Shek, for their generous support during times when research obstacles seemed

insurmountable. Their insights, advice, and encouragement have helped shape many of the ideas and solutions in this thesis.

I extend my appreciation to KCL DRIVE-Health CDT, Precision Health Informatics Data, and the Biostatistics and Health Informatics department. The vibrant, collaborative, and supportive environment within these groups has provided me with not only a stimulating academic experience but also a strong sense of belonging. I am grateful to the many colleagues who have contributed to discussions, shared their expertise, and offered their friendship along the way.

Finally, my heartfelt thanks go to my family for their unwavering support throughout this journey. Their belief in me, their patience, and their constant encouragement have been the foundation upon which I have built this work. This PhD would not have been possible without their love and understanding.

Abstract

Patient data are highly dimensional, multimodal, noisy, irregularly sampled and riddled with missingness. While neural network models show tremendous potential for uncovering complex patterns within patient treatment timelines, their utility for reliable analytics requires robust handling of missing data that preserves both accuracy and trustworthiness. This thesis advances the field through four complementary contributions aimed at better aligning neural architectures with electronic health record (EHR) characteristics.

First, through comprehensive benchmarking of neural imputation approaches, I identify critical gaps between theoretical model capabilities and practical performance, particularly in handling structured missingness patterns common in clinical data. This analysis reveals that implementation choices and evaluation practices significantly impact model performance, highlighting the need for more rigorous evaluation frameworks.

Building on these insights, I present two novel architectures: CSAI (Conditional Self-Attention Imputation) and DEARI (Deep Attention Recurrent Imputation). CSAI incorporates domain knowledge through an adaptive decay mechanism and transformer-enhanced initialisation, demonstrating superior performance in capturing the complex dependencies present in healthcare time-series. DEARI extends this work through a scalable deep architecture with integrated ability to quantify confidence in the produced output, particularly valuable for complex patient trajectories where understanding prediction reliability is crucial.

Finally, I introduce METHOD (Modular Efficient Transformer for Health Outcome Discovery), which rethinks the fundamental approach to handling irregular clinical data by

adapting modern transformer innovations specifically for healthcare applications. Through extensive evaluation, I demonstrate that these architectures achieve significant improvements over existing approaches, particularly in scenarios with high missingness rates and complex variable relationships.

This research provides both theoretical insights and practical tools for handling missing data in healthcare analytics, with the developed models being integrated into the open-source Python library `PyPOTS`, ensuring accessibility and reproducibility for the broader research community.

Table of contents

List of figures

Chapter 1

Introduction

1.1 Background and Motivation

The current data deluge in medicine due to the digitisation of patient care presents an enormous opportunity to uncover algorithmic insight and high-quality indicators of recommended and personalised treatments using data stored in Electronic Health Records (EHRs) [171]. This opportunity is supported by the ubiquity of models based on neural networks (NNs), like convolutional neural networks (CNN) [58], recurrent neural networks (RNN) [145], and multilayer perceptrons (MLP) [8]. These models operate by learning an appropriate mapping from complex feature vectors to a target output [35]. In the context of EHRs, the outputs could be class labels (e.g., mortality yes/no), regression scores (e.g., length of hospital stays), unsupervised clusters (e.g., disease subtypes), or latent embeddings (e.g., patient state representations) [111].

A multitude of sophisticated NN-based models have been designed to use EHR data to perform patient-centred predictive tasks [36, 179, 151, 69, 6]. Those have found numerous applications, ranging from the prediction of clinical events to disease progression modelling. The impact of these models has been established under *controlled settings*. Early warning systems processing vital signs and laboratory values can predict deterioration before clinical manifestation [43, 85]. Here, NN models have demonstrated superior accuracy compared to

traditional scoring systems [85]. In sepsis detection, where each hour of delayed treatment increases mortality by 7-8%, controlled experiments have demonstrated that deep NN models have been associated with a 5.0% increase in sepsis bundle compliance and a 1.9% decrease in hospital mortality [15]. In cardiology, algorithms analysing ECG measurements and vital signs can detect subtle patterns indicating impending arrhythmias [22], enabling preventive interventions. Despite the promising results, wide-scale validation with multimodal EHR data remains constrained by robustness issues due to data complexity, in addition to issues related to domain alignment [166].

The primary function of EHRs is to support clinical care. Data accumulates during routine healthcare activities, creating patient trajectories that inform clinical decision-making. These numerical trajectories manifest as complex time series, which are inherently noisy and multidimensional, encompassing numerous clinical variables collected according to specific workflows and recording practices [125, 4, 192]. The workflow-driven nature of data collection renders the time series irregularly sampled, with significant missingness rates and patterns which are intertwined with institutional and clinical protocols [101, 87].

The way EHR data is generated also creates sophisticated interdependencies between features, such as relationships between vital signs and their corresponding laboratory results. These interconnections create spatiotemporal correlations and latent relationships that pose significant algorithmic and computational challenges [83]. Consequently, carefully designed frameworks are needed to efficiently extract meaningful insights from these intricate data patterns.

In this thesis, I present a number of studies on the handling of the complexities of missing data patterns within ***structured EHR data*** through two complementary streams, with the aim of building robust predictive models:

1) **I ask if we are building models the right way**: This stream *constitutes the majority of the contributions I make in this thesis* and examines the nature of EHR time series to address gaps in how current neural approaches handle complex missingness in EHR time

series. I critically evaluate whether existing neural network-based imputation approaches are appropriately designed to align with the characteristics of the EHR. This investigation has yielded architectural contributions that address the alignment between models and EHR data along key dimensions: scaling deep imputation models to the distributional properties of EHRs, incorporation of domain insight, incorporating the ability to quantify confidence in the model's imputation, and studying the effect of preprocessing choices on model performance. The aim of this stream is to align both imputation and predictive methodologies based on neural networks with the unique characteristics of EHR time series, enabling more robust and reliable downstream predictive tasks.

2) I then **ask if we are building the right models**: Here, I challenge the traditional approach to handling data irregularity (i.e., completion of missing data) by exploring alternative paradigms. This investigation is motivated by recent advances in transformer architectures and their success in capturing long-range dependencies in sequential data. While these architectures have revolutionised natural language processing through large language models, their direct application to healthcare time-series presents unique challenges due to the complex temporal structures and patient-specific dependencies in clinical data. I therefore investigate how to adapt modern transformer innovations specifically for healthcare applications through patient-aware attention mechanisms and efficient modelling techniques tailored to the unique characteristics of clinical time-series data.

Through this dual investigation, I aim to contribute to both the immediate practical challenges of handling EHR data complexity and the longer-term theoretical question of optimal approaches to learning from incomplete clinical time series.

1.2 Research Questions, Aims and Objectives

Overall, my guiding research question asks how can advanced neural network architectures better address the complex missingness inherent to EHRs. This overarching question resolves into two complementary questions:

First, I examine ***whether current neural network approaches to handling multi-variate time-series missingness are appropriately designed for EHR characteristics,*** addressing gaps in model architectures, evaluation frameworks, and implementation practices.

Second, I ***explore whether imputation is fundamentally necessary by exploring novel generative approaches that can naturally handle temporal and numerical aspects of clinical data***.

By addressing these research questions, I aim to satisfy the main goal of this thesis: advancing state-of-the-art methodologies for developing neural network models in the context of EHRs. To achieve this, I will pursue the following objectives:

1. Design and evaluate novel neural network architectures that better align with the characteristics of EHR time series, with particular focus on handling complex missingness.

2. Assess the critical role of implementation choices and evaluation frameworks in determining model performance and reliability.

3. Contribute to the development of open-source benchmarking frameworks that enable systematic and transparent comparison of approaches to handling EHR complexity.

4. Advance alternative modelling paradigms through the adaptation of generative models for clinical time series data.

5. Validate all proposed approaches across diverse clinical prediction tasks and datasets.

1.3 Ethical Considerations

The data from the clinical domain is highly sensitive and often contains personally identifiable information (PII) requiring careful, controlled access. Examples of PII in EHR text could be patient names, addresses, phone numbers, email addresses, health and social care histories, personal family details and further 3rd party details relating to healthcare providers, family, friends, etc. that should be kept private.

All research within this thesis has leveraged three clinical databases, namely MIMIC-III [75], MIMIC-IV [76] and eICU [126] – large real-world, clinical databases that are openly-available after an ethics training course and ethical approval.

1.4 Chapter Summaries

A summary of each chapter follows:

Chapter 2: provides the necessary background for future chapters that present novel methodological contributions. The chapter frames missing data mechanisms in the context of the unique characteristics of EHR data, and discusses machine learning paradigms for time series imputation, and modern transformer architectures. The chapter also introduces `PyPOTS`, a Python toolkit I am heavily involved in, used for benchmarking neural models on partially-observed time series, and finally details the healthcare datasets used throughout the thesis.

Chapter 3: presents a comprehensive review along three axes: neural network imputers for healthcare time-series, the emerging role of foundation models in time series analysis, and the evolution of patient health trajectory modelling. The review employs the concept of inductive bias of neural networks to organise my understanding of different architectural approaches and their alignment with EHR characteristics.

Chapter 4: presents a comprehensive benchmarking study of state-of-the-art imputation models based on neural networks which have been designed to operate in the context

of healthcare time-series. The study presented in the chapter establishes best practices for model selection and implementation and informs the development of the models I present in subsequent chapters.

Chapter 5: presents my first contribution to the field of neural network imputation for EHR time series, CSAI, which employs a novel domain-informed temporal decay mechanism to better align neural imputation frameworks with EHR data characteristics.

Chapter 6: further investigates whether a well-designed deeper architecture can improve performance on complex EHR datasets. The chapter introduces the DEARI model, which incorporates uncertainty quantification through a Bayesian marginalisation strategy to enable establishing one's confidence in the resulting imputation, and demonstrates superior performance on datasets with high dimensionality and complex missing patterns.

Chapter 7: introduces METHOD, which is my current take on adapting modern transformer innovations specifically for healthcare applications. The model introduces patient-aware attention mechanisms and efficient long-sequence modelling techniques to better handle the unique patient-specific dependencies in clinical data.

Chapter 8: compiles the contributions and insights gained from my research throughout my PhD, discusses the broader impact of my contributions on healthcare analytics and the research community, and outlines promising directions for future work. The chapter emphasises both the theoretical advances and practical tools developed through this research, particularly highlighting the integration of models into the open-source Python library, `PyPOTS`.

Chapter 2

Background, Methods, Tools and Datasets

This chapter introduces the necessary background of concepts, methods, tools and datasets used throughout the thesis. The chapter begins by introducing fundamental concepts that guide our current understanding of missing data in time series and missing data properties in Section 2.1. The chapter then presents an overview of how EHR data collection leads to unique characteristics, relating those to missingness properties in Section 2.2. Having established the necessary background, the chapter then presents the notation of incomplete multivariate time-series used in all my studies in Section 2.3.

In the sections that follow, the chapter presents an overview of the paradigms of machine learning, with a particular focus on neural networks in Section 2.4. This is followed by an introduction to `PyPOTS` [47], which is currently the only openly-available Python toolkit providing a transparent and controlled environment for benchmarking neural network models operating on partially-observed time-series (i.e. time-series with data). The discussion on `PyPOTS` is of high importance to this thesis for two reasons. First, `PyPOTS` the library is used to perform a large number of the experiments presented throughout the document. Second, I am personally invested in `PyPOTS`, having been deeply involved in the development and maturation of different `PyPOTS` components throughout

my studies, I have successfully deployed some of the novel work presented here into the package, making those available for researchers as open-source and open-access models (the details will be discussed in the pertinent parts of the thesis).

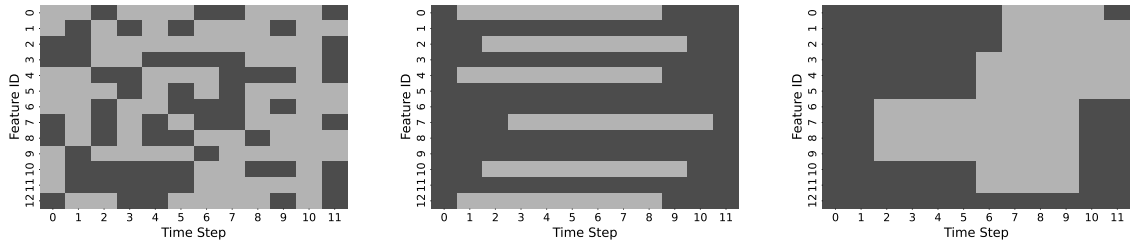
The chapter then continues with a comprehensive examination of modern Transformer architectures in Section 2.7, focusing on the key innovations that have enabled the development of today's large language models. This section details crucial advancements in architectural design, including improvements in **attention mechanisms** and **position encoding strategies**, as well as significant training innovations that have addressed computational efficiency and scalability challenges. Understanding these developments is particularly relevant as healthcare data analysis increasingly adopts and adapts transformer-based architectures for processing complex temporal sequences in electronic health records.

Finally, the chapter concludes by detailing the sources and properties of the datasets used in my work in Section 2.8.

2.1 Studying Missingness in Time Series

Missingness of data in time series can be understood through two fundamental perspectives: missingness mechanisms and missingness patterns. **Missing mechanisms** explain why data are missing, while missingness patterns define how these missing values manifest within a given dataset. Classically, missingness mechanisms are understood through Robin's classification of *Missing Completely at Random* (MCAR), *Missing at Random* (MAR) and *Missing not at Random* (MNAR) - those are discussed in detail in Appendix A. While Robin's classification explains the means by which a variable can be missing, it does not provide insight into the observed **patterns of missingness** in real datasets, which can take several forms and are related to the nature of data collections within a particular domain. In EHR time series, where a multitude of variables evolve over time, the patterns of missingness of the different variables can be generally described by the patterns shown in Figure 2.1. In *point-based missingness*, random scattered missing values

occur independently throughout the time series, appearing as isolated spots in an otherwise complete dataset. On the other hand, *subsequence missingness* manifest as continuous segments of missing values that occur in regular patterns across multiple features at consistent time intervals or within one variable across time, while *block missingness* forms large contiguous chunks of missing data that affect multiple features across extended time periods. The significance of these patterns lies in their implications for data analysis and imputation strategies, which will be discussed throughout the thesis.



(a) Point-based Missingness (b) Subsequence Missingness (c) Block Missingness

Fig. 2.1 Heatmap visualisation of the three different missing patterns. The observed values are presented in dark grey, while the missing values are in light grey.

2.2 Characteristics of EHR Data

The primary function of EHRs is to support clinical care. EHR data accumulates during routine healthcare activities, forming patient trajectories that clinicians rely on for decision-making. These trajectories cover various dimensions of a patient's health status, such as demographics, diagnoses, medications, procedures, vital signs, test results, images, and clinical text notes in a sequence of visits. Using the time series generated by this data resource to inform predictive tasks requires dealing with a unique set of characteristics; those are summarised in Table 2.1.

The multidimensional coverage of EHR time series renders them inherently **multimodal**, capturing patient health dynamics through *continuous measurements* (e.g., ECG signals), high- and low-frequency *discrete events* (e.g., heart rate, medication changes),

and *ordinal data* (e.g., cancer stages). These multimodal variables are recorded **asynchronously** at intervals determined by device capabilities, clinical events, and protocols, among others [71]. This irregularity in sampling EHR variables is a leading factor to the structured and high-rate missingness in this valuable resource [95, 174]. Device resolution dictates high-frequency measurements, such as automated recordings by blood pressure cuffs every five minutes in intensive care units, compared to capillary blood glucose measurements via a glucometer 4-6 times daily [114]. Event-based recordings are primarily driven by treatment requirements, exemplified by insulin administration before meals with subsequent adjustments based on glucose readings [137]. Drug recording patterns are particularly influenced by their effect cycles—the temporal sequence from administration through onset, peak effect, duration, and decline—creating predictable physiological timelines [113]. Laboratory tests, such as daily complete blood counts during acute leukaemia treatment or troponin tests if a patient experiences chest pain, are treatment-dependent and are performed based on anticipated clinical needs [124]. Additionally, institutional protocols impose structured recording intervals, such as nursing assessments clustering around 12-hour shift end intervals, physician notes predominantly recorded during morning rounds, and risk assessments performed at admission and discharge. Some measurements follow strict periodic schedules, such as quarterly cancer staging during chemotherapy, while others occur irregularly in response to acute clinical needs, as seen with emergency surgical procedures.

Finally, because many health parameters are inherently connected, clinical variables exhibit complex interdependencies, and their missingness patterns often carry meaningful information about patient state and care processes. Many variables are **cross-sectionally correlated**, which can introduce redundancy by overemphasising certain characteristics. For example, clinical practice often dictates ordering test panels, and laboratory tests are seldom requested in isolation; ordering electrolyte tests typically includes kidney function tests, and calcium tests require concurrent albumin measurements to adjust calcium levels [124]. EHR time-series correlations also extend across the temporal dimension,

Table 2.1 Characteristics and dependencies of clinical time series and the resulting missingness mechanisms. **MCAR:** missing completely at random; **MAR:** missing at random; **MNAR:** missing not at random.

Modality	Clinical Category	Clinical Examples	Recording Patterns	Temporal Dependencies	Cross-sectional Correlations	Missingness Patterns
Continuous	Waveform Signals	Invasive vital sign monitoring, ECG signals	Millisecond	Short-term: beat-to-beat variations	Correlation with derived vitals	MNAR: equipment disconnection, e.g. patient bathing
High-Frequency	Derived Vitals	Heart rate, Blood Pressure, Temperature	Every 1-5 minutes; setting-dependent	Short: acute physiological responses	Strong correlation within vital panels	MAR: during in-hospital patient transport
Discrete	Laboratory Tests	Full blood count, Kidney function	Treatment-dependent; ordered in panels	Short: acute organ dysfunction	Strong correlation within panels	MAR: depends on disease severity
	Ventilator Data	FiO2, PEEP, Tidal volume	Protocol-driven sampling	Short: ventilation adjustments	Strong correlation within parameters	MCAR/MNAR: technical/clinical
	Medications	Antibiotics, Vasopressors	Event-based, effect cycles	Short: treatment response	Correlation with lab results	Structured: protocol-based
	Procedures	Surgery, Endoscopy	Irregular, protocol-driven	Short: acute intervention effects	Strong correlation with vitals	Structured: protocol-based
	Clinical Notes	Assessments, Care plans	Daily/shift-based updates	Long: disease progression	Correlation with multiple measures	MAR: severity dependent
Ordinal	Clinical Scores	PHQ9	Daily/weekly or shift-dependent	Short: acute deterioration	Correlation with vital signs	MAR: status dependent
	Disease Staging	Cancer stage, Heart failure class	Monthly/quarterly updates	Long: disease progression	Correlation with multiple markers	Structured: protocol-based
	Risk Assessments	Mortality, Readmission risk	Admission/ discharge timing	Short/Long: risk evolution	Complex correlation with multiple vars	Structured: event-based

encompassing **short-term and long-term temporal dependencies** that hold significant meaning within a patient's trajectory. Immediate physiological responses reveal acute body reactions, while prolonged interventions or chronic conditions manifest long-term effects. For example, a heart attack typically results in immediate-term changes in vital signs (including blood pressure and heart rate), intermediate-term changes in biomarkers (such as troponin and renal function markers), and if it progresses to heart failure, long-term alterations in multiple physiological parameters [163]. EHR correlations introduce an additional level of complexity for clinical predictive models, as clinical insights often lie in the extreme values of the complex data distributions of clinical variables, such as abnormally high or low blood sugar. An imputation model, therefore, needs to accurately discern those informative outliers which mark physiological events, from noise.

2.3 Terminology: Incomplete Multivariate Time-series

For a temporal observation over T discrete time steps, we represent a multivariate time series as a matrix $X \in \mathbb{R}^{T\times D}$. $X = \{x_1, x_2, ..., x_T\}$ is composed of T observations, each denoted by a vector $x_t \in \mathbb{R}^{1\times D}$ of D features observed at timestamp s_t.

Information related to missing values is encapsulated within two derived matrices (see Fig.2.2). The mask matrix $M \in \mathbb{R}^{T\times D}$ indicates whether each element of X is observed or missing.

$$m_t^d = \begin{cases} 0, & \text{if } x_t^d \text{ is missing} \\ 1, & \text{otherwise} \end{cases} \tag{2.1}$$

Furthermore, given that the time elapsed between consecutive observations can vary, we denote the time gaps at each time step by an additional component δ_t^d. $\delta_t^d \in \mathbb{R}^{T\times D}$ encodes the time gap between two successive observed values for each feature d, providing an additional indicator of temporal context to the dataset.

$$\delta_t^d = \begin{cases} s_t - s_{t-1} + \delta_{t-1}^d & \text{if } t > 1,\ m_t^d = 0 \\ s_t - s_{t-1} & \text{if } t > 1,\ m_t^d = 1 \\ 0 & \text{if } t = 1 \end{cases} \tag{2.2}$$

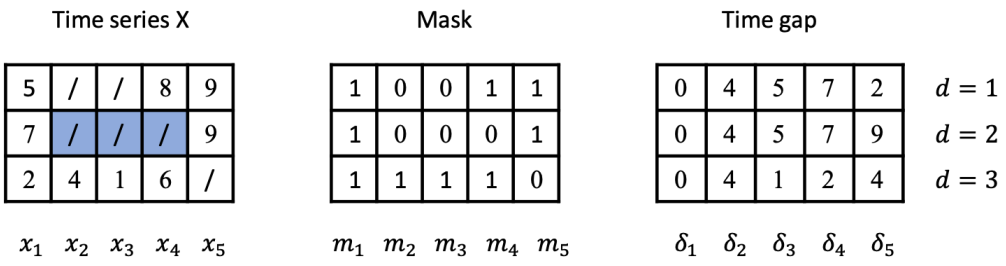


Fig. 2.2 Observations x_{1-5} in time-stamps $s_{1-5} = 0, 4, 5, 7, 9$. Feature d_2 was missing during s_{2-4}, the last observation took place at s_1. Hence, $\delta_5^2 = t_5 - t_1 = 9 - 0 = 9$.

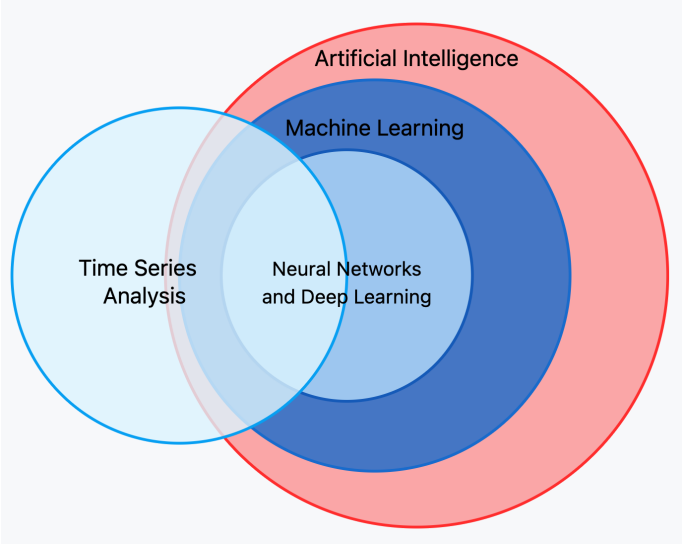


Fig. 2.3 Venn diagram showing the relationships between Artificial Intelligence, Machine Learning, Deep Learning, and Time Series Analysis methodologies.

2.4 Machine Learning for Time Series Imputation

I will now review machine learning (ML) methods relevant to the processing and imputation of time series data. Machine learning is concerned with the optimisation of a set of parameters θ with respect to observed data X, to perform a given task without explicit programming. Figure 2.3 shows how ML methods fit within wider methodological fields of time series analysis and artificial intelligence (AI). AI is a broad subfield of computer science focused on developing computational systems that can perform tasks that traditionally require human capabilities, such as pattern recognition, decision-making, and adaptation to new situations. Machine learning (and its subset of models based on artificial neural networks) is only one branch of AI, despite currently being the dominant paradigm, with the terms ML and AI sometimes used (inaccurately) interchangeably.

Time series analysis is broader than just the application of ML methods to temporal data. For example, traditional statistical methods such as ARIMA models, exponential smoothing, and state space models have been successfully applied to time series analysis without utilising ML techniques [16]. Similarly, rule-based imputation methods and domain-specific interpolation techniques have a long history of handling missing values

[96, 142] and could be considered time series analysis but not ML. ML methods exist outside of time series analysis, and the converse is also true.

Within the context of medical time series, machine learning methods can largely be split into two different paradigms.

Supervised Learning: Given a dataset $D = \{(X, y)\}$ where X is a time series and y is a vector of corresponding labels, supervised learning aims to find some function $f(X) = Y$. θ are the parameters of f, such that θ is the optimal parameters that bring us as close to Y given inputs X.

Unsupervised Learning: Given a dataset $D = X$ where X is a time series without labels, unsupervised learning aims to find some function $f(X)$ that captures underlying patterns in the data. θ are the parameters of f, such that θ is the optimal parameter that best represents the inherent structure of X.

Regardless of the objective function, machine learning algorithms differ in the way θ are optimised to find f. In neural networks, this is done through backpropagation. An introduction to artificial neural networks, first principle architectures (i.e. feed-forward networks), as well as specialised architectures (convolutional, recurrent, and transformer networks) used throughout the thesis, are provided in Appendix B

The dichotomy between these two traditional learning paradigms is not helpful within the context of medical time series. Supervised learning requires large amounts of labelled data, which is often scarce in healthcare due to the cost and time required for expert annotation [67]. Meanwhile, purely unsupervised approaches may fail to capture clinically relevant patterns without any form of guidance. These challenges have led the way to active efforts in using **self-supervised learning** in medical time-series, which creates supervisory signals from the data itself.

Self-Supervised Learning in Medical Time Series In the context of medical time series, self-supervised learning can leverage the inherent structure of physiological measurements and their temporal relationships to learn meaningful representations [182, 20], as the training signals are automatically derived from the data rather than manually provided. For example, a model can be trained to predict missing vital signs from surrounding measurements, or to identify temporal patterns in lab results, without requiring explicit labels. This paradigm has proven particularly effective for medical time series as it combines the advantages of supervised learning's directed optimisation with unsupervised learning's ability to work with unlabelled data.

Self-Supervised Learning: Given a dataset $D = X$ where X is a time-series, self-supervised learning creates labels y from the data itself and aims to find some function $f(X) = Y$. θ are the parameters of f, such that θ is the optimal parameter that brings us as close to the automatically generated Y given inputs X.

The majority of this thesis focuses on the development and application of self-supervised learning methods for time series imputation in medical data, to subsequently enable downstream tasks that maybe supervised, unsupervised, or semi-supervised. Recent work has shown that self-supervised approaches can effectively leverage the temporal structure of medical data to learn meaningful representations without requiring explicit labels [88, 177].

2.4.1 Machine Learning Experimental Methodology and Evaluation Metrics

In most ML modelling exercises, the primary aim is often to output a high-performing, generalisable model. During model development, a dataset X is often split into $X_{train}, X_{val}, X_{test}$ with ratios in the region of 80/10/10. This provides the majority (80%) of the original input data for model training and optimisation of actual model parameters θ. Then 10% for the

model, and hyperparameter improvement using the validation set, and the final 10% for final testing of the model. Practitioners should only report results of their models using the test set and never perform hyperparameter optimisation with the test set.

A model that has converged, i.e., the loss is no longer decreasing after a number of update iterations, but still outputs poor train, validation and even test set performance can be said to *underfit* the data. In our logistic regression example, this could mean that the TF-IDF matrix does not sufficiently capture the relationships between words or phrases for the task and/or the model itself lacks the parameters to capture those relationships.

Overfitting occurs if a model performs well on the training and validation sets and the training loss either is or approaches 0, but the test set performance is not good. This suggests the model has too closely fit to the specific idiosyncrasies of the train and validation data and therefore is not generalisable to the unseen test data.

Model performance is evaluated using several complementary metrics that capture different aspects of imputation accuracy. The primary metrics for unsupervised and semi-supervised methods, which we use to evaluate a model's imputation performance are mean absolute error (MAE) and mean squared error (MSE), with MSE providing additional sensitivity to larger errors. The mathematical definitions are given below:

$$\begin{aligned}
\mathrm{MAE}\,(\hat{y}, y, m) &= \frac{\sum_{d=1}^{D}\sum_{t=1}^{T}|(\hat{y}_t^d - y_t^d)\cdot m_t^d|}{\sum_{d=1}^{D}\sum_{t=1}^{T} m_t^d}, \\
\mathrm{MSE}\,(\hat{y}, y, m) &= \frac{\sum_{d=1}^{D}\sum_{t=1}^{T}((\hat{y}_t^d - y_t^d)\cdot m_t^d)^2}{\sum_{d=1}^{D}\sum_{t=1}^{T} m_t^d}, \\
\mathrm{MRE}\,(\hat{y}, y, m) &= \frac{\sum_{d=1}^{D}\sum_{t=1}^{T}|(\hat{y}_t^d - y_t^d)\cdot m_t^d|}{\sum_{d=1}^{D}\sum_{t=1}^{T}|y_t^d \cdot m_t^d|},
\end{aligned}$$

where:

- $\hat{y}$ is the estimated value
- y indicates the target value

- m represents the mask with time index t and dimension index d
- n is the number of data points

For downstream classification tasks that use the imputed values, the following metrics are used to evaluate supervised learning models operating on the data to perform a predictive task:

- Accuracy: $\frac{TP+TN}{TP+TN+FP+FN}$: a common metric, that is the proportion of both true positive (TP) and true negative (TN) predictions normalised by the sum of all possible prediction states.
- Specificity: $\frac{TN}{TN+FP}$: the proportion of correct negative predictions of total negative predictions (either true negative (TN) or false positive (FP)). E.g., if we have 10 negative samples and the model predicts 5 negatives correctly, we have 50% specificity.
- Sensitivity or Recall: $\frac{TP}{TP+FN}$: the proportion of positive predictions of total positive predictions (true positive (TP) and false negative (FN)). E.g., if we have 10 positive samples out of 20 in a dataset and the model only predicts 5 out of the 10 correct, we have 50% recall.
- Precision: $\frac{TP}{TP+FP}$: the proportion of positive predictions that are correct. E.g., if the model only makes 2 positive predictions and 1 is correct we have 50% precision.
- F1: $\frac{2\cdot(Precision\cdot Recall)}{Precision+Recall}$: The harmonic mean between precision and recall. F1 scores are often reported as they include both precision and recall, and therefore offer a good view of classifier performance irrespective of data-specific issues that could skew performance in favour of recall or precision. E.g., if a class only appears once or a few times, achieving high recall might be trivial for a classifier that has many false positives.

- ROC-AUC (Receiver Operating Characteristic Area Under Curve): A metric that plots the true positive rate (sensitivity/recall) against the false positive rate ($\frac{FP}{FP+TN}$) at various classification thresholds. The resulting curve's area represents the model's ability to distinguish between classes. A perfect classifier has ROC-AUC = 1, while random guessing yields 0.5. This metric is particularly useful when classes are relatively balanced.

- PR-AUC (Precision-Recall Area Under Curve): Similar to ROC-AUC, but plots precision against recall at various classification thresholds. The resulting curve's area indicates the model's ability to make accurate positive predictions while capturing all positive instances. PR-AUC is especially valuable for imbalanced datasets where negative instances greatly outnumber positive ones, as it focuses on the minority class performance without being influenced by a large number of true negatives.

2.5 Inductive Bias Across Neural Architectures & Frameworks

The general objective of machine learning models is to find the optimal parameters θ of the function f, as explained in Section 2.4. There exists an infinite number of solutions for θ, and each type of neural network architecture makes a set of preferences, priors, or assumptions that guide the learning process by reducing the space of possible solutions through prioritizing one solution over another, independent of the observed data. These preferences are termed the model's **inductive bias** [56]. The inductive bias of different neural network architectures and mathematical frameworks has a profound effect on their behaviour, subsequently affecting the types of dependencies a model is able to capture within a multivariate time-series. The literature review of neural network imputers I present in Chapter 3 is heavily grounded in the notion of inductive bias. I will therefore use this

section to provide an overview of the different inductive biases of neural network models that have been used in EHR neural network imputers.

Neural architectures differ in the way they *represent and process data*. **Recurrent Neural Networks (RNNs)** are intuitively suited to handle temporal sequences [107]. Their inherent inductive bias favours learning short-term temporal dependencies between variables over time by maintaining internal (hidden) states across time steps in their recurrent architectures. **Transformers** deviate from traditional sequence processing methods by capturing long-range dependencies through self-attention, with an inherent bias towards global contextuality [170]. **Convolutional Neural Networks (CNNs)** possess an inductive bias towards cross-sectional patterns, useful for detecting acute physiological changes that can mark important clinical events, e.g., detecting signs of leukaemia from white blood cell morphology. Finally, **Graph Neural Networks (GNNs)** [190] exhibit a relational graph bias, representing complex interdependencies through message-passing algorithms on node-edge topologies, enabling them to model the complex relational structures between interconnected health indicators.

Modern neural network imputers employ a variety of generative learning frameworks to improve their imputation; those are summarised in Figure 2.4. Generative frameworks diverge in their approach to data generation and uncertainty representation. **Variational Autoencoders** (VAEs) assume that data is generated from a latent space, and aim to learn a distribution, usually Gaussian [103], capturing the underlying structure of the data. As such, VAEs are inherently probabilistic models biased by the assumed distribution. **Mixture Density Networks (MDNs)** overcome the single-distribution bias by assuming that the data is generated from a mixture of probability distributions. They directly capture imputation uncertainty through a mixture of weights and variances in the assumed distributions.

Generative Adversarial Networks (GANs) [32] take distribution complexity further by being inherently biased towards generating realistically diverse data distributions. This is done through an adversarial setup where the generator aims to fool the discriminator. While this enables the effective generation of multivariate time-series, GANs often struggle

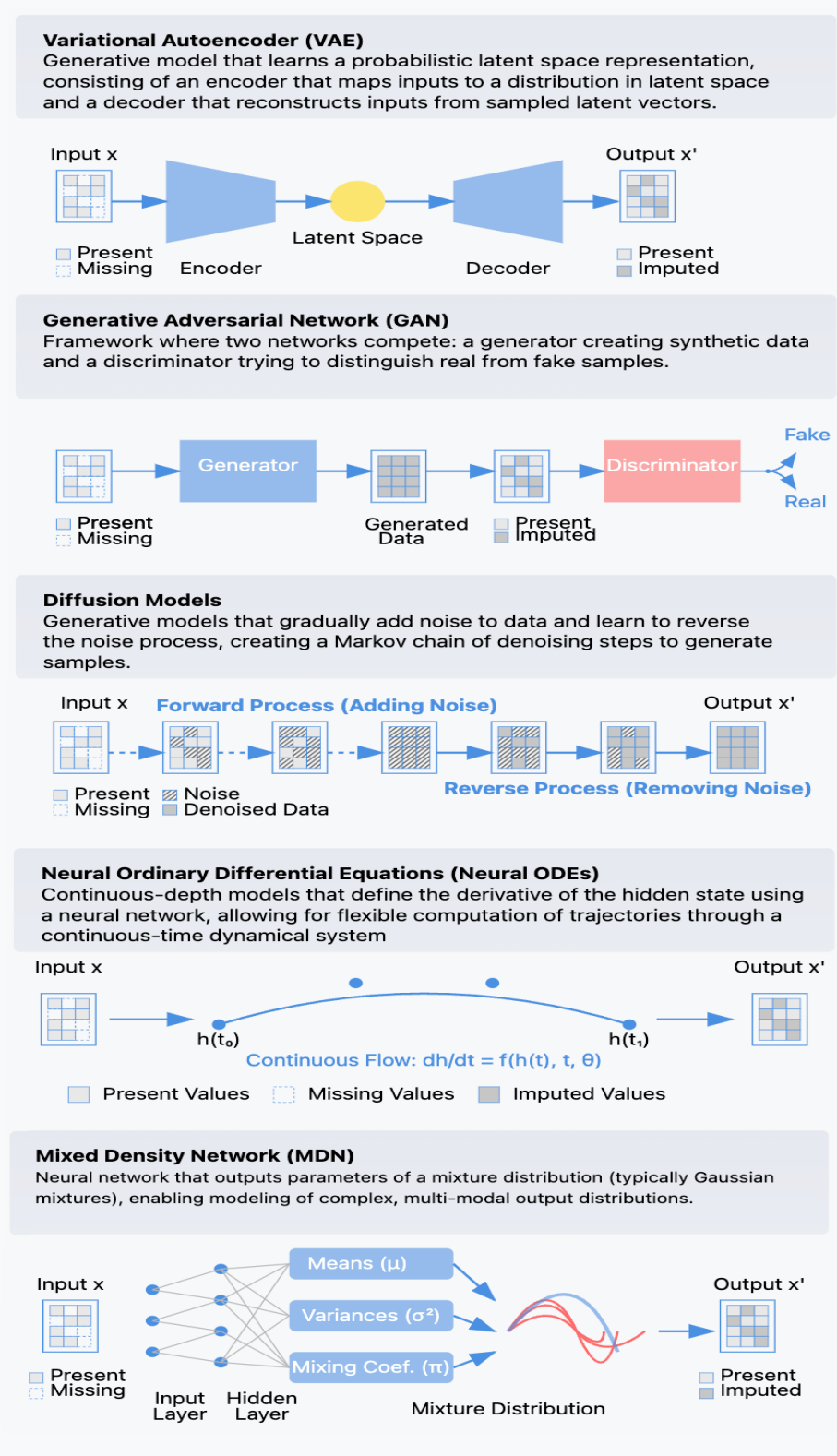


Fig. 2.4 A conceptual overview of generative frameworks used in medical time-series imputation.

with mode collapse, producing a limited variety of common patterns while missing rare but clinically significant events. Furthermore, GANs inherently lack direct mechanisms to quantify uncertainty within the imputations, and their application to establishing confidence in the generated data is still in its early stages [119]. **Neural Ordinary Differential Equations (Neural ODEs)** [25] describe the temporal evolution of the data using differential equations, using the learned function to simulate the dynamics of the data over time. ODE models favour continuous data transitions aligning with the functions learned by the model [144, 122]. While their continuity bias can help handle irregular sampling in EHR data [138], it also prevents them from capturing sudden physiological changes and discrete interventions common in healthcare settings (e.g., sudden onset of symptoms or treatment effects). Although uncertainty can be incorporated as a stochastic process, solving these equations is computationally intensive and highly sensitive to initial conditions. Finally, **Diffusion Models** [63, 153] employ a stochastic process to gradually generate synthetic data, progressing from random noise towards distributions that mimic the observed data. Diffusion models assume that data evolves over time according to a gradual process. However, similar to Neural ODEs, the gradual denoising bias may cause diffusion models to miss or smooth over clinically significant sudden changes and may struggle to capture complex dependencies between clinical variables that change at different scales. Although diffusion models do not provide an explicit mechanism to quantify uncertainty, they can estimate uncertainty by measuring the divergence between predicted and observed data distributions at each time step.

2.6 PyPOTS: A Python Toolbox for Data Mining on Partially-Observed Time Series

Real-world time series, exemplified by EHR time series, often suffer from missing values, creating significant challenges for data analysis and modelling. PyPOTS (Python Partially-

Observed Time Series) is a comprehensive Python toolbox specifically designed to address these challenges, providing a unified framework for handling partially-observed time series data [48]. PyPOTS provides easy, standardised access to a variety of algorithms for imputation, classification, clustering, and forecasting. It supports the entire imputation workflow, from data loading and processing to model building and data imputation, and is built via a modular design to provide the following capabilities (illustrated in Figure 2.5).

- A unified interface for various time series mining tasks including classification, clustering, forecasting, and imputation.
- Implementation of state-of-the-art deep learning models specifically designed for partially-observed time series.
- Comprehensive data processing utilities for handling missing values and irregular sampling.
- Standardised evaluation metrics and visualisation tools.

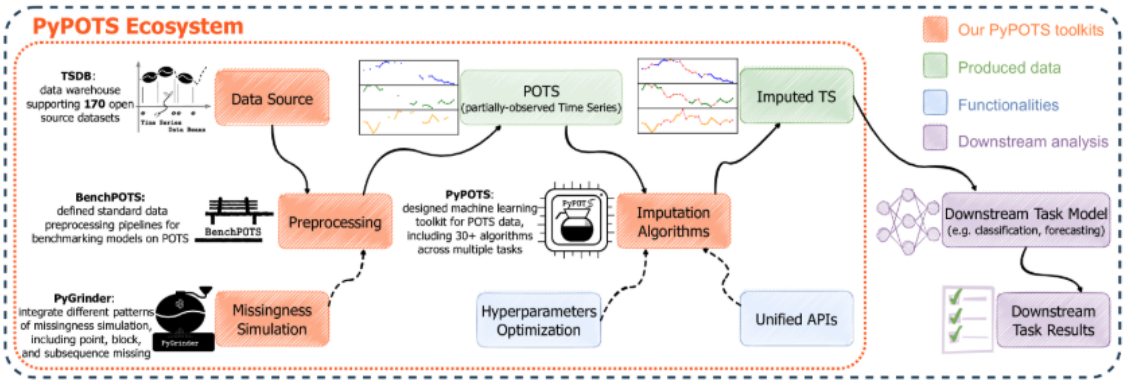


Fig. 2.5 The PyPOTS Ecosystem.

In this thesis, PyPOTS has been used to benchmark several imputation algorithms based on deep learning. In addition, part of the thesis' contribution is the incorporation of the novel architectures in the PyPOTS library. Both contexts will be detailed in the relevant chapters.

2.7 Evolution of Modern Transformer

The Transformer architecture in B.5, first introduced by Vaswani et al. [170], has revolutionised natural language processing (NLP) and become the foundation for modern large language models (LLMs). This section examines the key architectural innovations, training advances, and data processing improvements that have shaped the development of modern Transformers, leading to today's highly capable language models.

2.7.1 Architectural Innovations

The evolution of Transformer architecture has been driven by challenges such as computational efficiency, memory usage, and the ability to handle longer sequences. Early efforts primarily focused on improving the attention mechanism and position encoding strategies [162, 93].

Attention Mechanism Improvements

The original Transformer's self-attention mechanism scales quadratically with sequence length, creating significant computational and memory constraints. Several innovations have addressed this limitation:

- **Sparse Attention**: Introduced by Child et al. [27], sparse attention patterns reduce complexity by limiting each token's attention to a subset of other tokens.
- **Linear Attention**: Methods such as Performers [30] approximate attention using kernel tricks to achieve linear complexity, enabling scalable computations.
- **Local-Global Attention**: Models like Longformer [12] and BigBird [184] combine local attention patterns with global tokens to handle longer contexts efficiently.

Position Encoding Advances

Position encoding strategies have evolved significantly to improve the representation of token ordering:

- **Relative Position Encodings**: Proposed by Shaw, Uszkoreit, and Vaswani [148], these encodings better capture the relationships between tokens.
- **Rotary Position Embeddings (RoPE)**: Developed by Su et al. [156], RoPE improves generalisation to longer sequences by introducing rotational transformations.
- **ALiBi (Attention with Linear Biases)**: Proposed by Press, Smith, and Lewis [127], ALiBi enables models to extrapolate effectively to unseen sequence lengths.

2.7.2 Pre-training Strategies

The evolution of pre-training strategies has significantly influenced modern Transformer development:

- **Masked Language Modelling (MLM)**: Introduced by BERT [79], MLM enables bidirectional context understanding by predicting masked tokens.
- **Causal Language Modelling**: Employed by GPT models [17], this approach enables powerful generative capabilities.
- **Prefix Language Modelling**: Used in models like UniLM [41], combining benefits of both MLM and causal modelling.
- **Span-based Masking**: Implemented in models like SpanBERT [77], improving coherent phrase understanding.

2.7.3 Training Innovations

The scale and complexity of modern LLMs have necessitated numerous training innovations to improve efficiency, stability, and scalability.

Optimisation Techniques

Modern optimisation strategies have been developed to address the challenges of training large-scale models:

- **Mixed Precision Training**: Combining 16-bit and 32-bit floating-point computations reduces memory usage without sacrificing stability.
- **Gradient Accumulation**: Enables training with larger effective batch sizes on limited hardware by accumulating gradients over multiple iterations.
- **Activation Checkpointing**: Trades computation for memory by recomputing activations during the backward pass [57].

Data Processing Advances

Efficient data handling and preparation have become critical for scaling LLM training:

- **Advanced Tokenisation**: Methods such as SentencePiece [82] and BPE-dropout improve the tokenisation process for diverse languages and domains.
- **Efficient Data Loading**: Innovations such as memory-mapped files and streaming datasets ensure high-throughput data pipelines.
- **Data Mixing Strategies**: Combining datasets with varying characteristics improves model robustness and generalisation [132].

2.7.4 Modern Architectures

Recent architectural advancements have further enhanced the efficiency and capability of Transformer-based models:

- **Flash Attention**: Introduced by Dao et al. [34], this technique dramatically improves the efficiency of attention computation.

- **Grouped-Query Attention (GQA)**: Reduces computational costs while maintaining model quality [2].
- **Multi-Query Attention**: Offers better inference efficiency by sharing attention key-value pairs across queries.

2.7.5 Emerging Techniques

Several promising research directions are shaping the future of Transformers:

- **Mixture of Experts (MoE)**: Techniques such as GLaM [45] enable larger model capacity without proportional computational costs.
- **Retrieval-Augmented Generation (RAG)**: Combines parametric knowledge with external non-parametric memory for improved reasoning and factual accuracy [86].
- **Continuous Learning**: New methods aim to enable models to integrate updated knowledge dynamically without requiring full retraining.

These advances collectively have enabled the development of increasingly powerful language models while improving efficiency and reducing computational requirements. The field continues to evolve rapidly, with new innovations regularly emerging to address remaining challenges in scalability, efficiency, and model capabilities.

2.8 Datasets Used in this Thesis

The experiments performed in this thesis used four healthcare benchmarks obtained from the following widely used databases. The details of missing data distribution for each generated benchmark are found in Appendix C.

The eICU database. The eICU Collaborative Research Database[1] [126] is a public multi-centre database with anonymised health data connected to more than 200,000 US ICU hospitalizations. I followed the only benchmark extract available for eICU [149] to produce a dataset comprising 20 variables for 30,680 patients.

The MIMIC-III database. Medical Information Mart for Intensive Care III (MIMIC-III)[2] [75], an extensive, freely-available single-hospital database of over 40,000 critical care patients based in Boston, Massachusetts, between 2001 and 2012. I followed two well-cited benchmarks [59] and [128] to produce two datasets comprising 59 variables for 21,128 patients and 89 variables for 14,188 patients.

The MIMIC-IV database. Medical Information Mart for Intensive Care IV (MIMIC-IV)[3] [76], a publicly-available multi-modal database containing comprehensive clinical information of patients admitted to the emergency department or an intensive care unit at the Beth Israel Deaconess Medical Center in Boston, MA. MIMIC-IV contains data for over 65,000 patients admitted to an ICU and over 200,000 patients admitted to the emergency department.

The PhysioNet2012 dataset. The PhysioNet/Computing in Cardiology Challenge 2012 dataset[4] [152] includes 12,000 ICU patient records from the MIMIC II Clinical database, version 2.6 [140], focusing on patient-specific prediction of in-hospital mortality using data from the first 48 hours of ICU admission. The whole dataset has three subsets, and we only use the subset A in experiments.

For all data sets, I reproduced the available benchmarking, skipping the steps that remove all-**NAN** samples to retain the data with its original missingness. The missingness

[1]https://physionet.org/content/eicu-crd/2.0/
[2]https://physionet.org/content/mimiciii/1.4/
[3]https://physionet.org/content/mimiciv/3.1/
[4]https://physionet.org/content/challenge-2012/1.0.0/

Dataset	Size	Avg Baseline Missingness	Feature Correlation	(Static; Categorical) Variables
eICU	30,680 x 20	40.53%	0.14	(6; 3)
MIMIC_59	21,128 x 59	61%	0.17	(0;0)
MIMIC_89	14,188 x 89	78%	0.14	(10; 10)
Physionet	3,997 x 35	51%	0.12	(0; 0)

Table 2.2 Characteristics of the four datasets: **Size** is described as number of samples $\times$ number of variables. **Average baseline missingness** of all features. **Feature correlation** is the average feature-wise Pearson correlation coefficient. **(static; categorical)** the number of static (as opposed to dynamic) and categorical (as opposed to numerical) variables in the dataset.

ratios for each resulting benchmark is detailed in Appendix C. Each of the datasets used has a different data distribution and characteristics as shown in Table 2.2 below.

Chapter 3

Literature Review

This chapter reviews the literature along three domains. First, Section 3.1 provides an overview of neural network imputation models operating within the context of EHR time series. Here, I use the concept of inductive bias (covered in Section 2.5) to structure our understanding of the characteristics of each model, enabling the alignment of each with a specific subset of EHR data characteristics. Second, Section 3.2 explores the rapidly evolving landscape of foundation models in time series analysis, examining how recent advances in large language models and semantic tokenisation strategies can be adapted for temporal data. This final part of the chapter presents a focused discussion of patient health trajectory modelling (Section 3.3), investigating how transformer-based architectures and standardized frameworks like MEDS are reshaping our approach to longitudinal healthcare data analysis. Together, these sections provide a comprehensive view of both traditional imputation methods and emerging paradigms in medical time series analysis.

3.1 Taxonomy of Neural Network Imputers

The exploration of the literature presented here organises neural network imputers by examining their approaches to handling the complex characteristics of EHR data discussed in Section 2.2. Backed by the knowledge that inductive bias significantly influences a

model's behaviour and generalisation capabilities [56], I present a taxonomy of neural network imputers that has been formulated using the following observations (illustrated in Figure 3.1).

1) **Component Integration:** Many modern neural network imputers combine neural network **architectures** with probabilistic generative **frameworks**, each serving a complementary role with distinct inductive biases. Architectures encode structural assumptions about data patterns and determine how data is processed, while frameworks determine preferences about how missing values are modelled and how the underlying distribution of the data is discovered, guiding the imputation process towards plausible generalisations that capture and replicate complex data distributions. The resulting synergy of preferences shapes an imputer's attempt to generate data that reflects realistic and clinically relevant patterns.

2) **Hierarchical Organisation:** An imputer's inductive biases accumulate: design modifications influence how fundamental architectural and framework biases inform higher-level preferences about data dependencies and patterns. This hierarchy guides systematic model development by connecting design choices to underlying assumptions about the data's complexity and missingness patterns.

3) **Model-Bias Alignment**: We hypothesise that a neural network imputer's performance depends on how well its fundamental and higher-level biases align with dataset characteristics. Mismatches between architectural, framework, or high-level biases and data properties can limit model effectiveness, guiding systematic choices in model design.

My organisation of the literature is shown in Tables 3.1 and 3.2. Table 3.1 lists all neural network imputation methods surveyed along with their basic component (architecture, framework) inductive bias. The selection criteria for inclusion of the relevant models are as follows: neural network imputation methods designed for multivariate time series, evaluated on at least one EHR dataset. I have excluded statistical and traditional machine

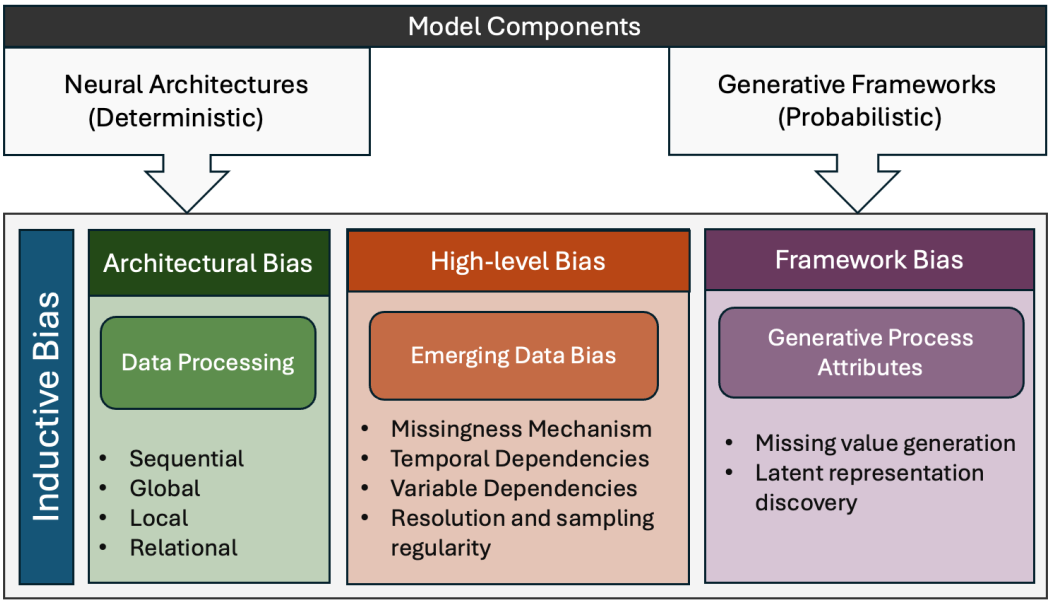


Fig. 3.1 The hierarchy of dimensions of inductive bias governing the behaviour of neural network imputation models.

learning methods and those without EHR benchmark validation. In Table 3.2, I delve into the higher-level biases that form due to design modifications applied to the components to further improve imputation performance. This table lists the types of missingness based on what has been reported in the original papers. Below is the discussion of the design principles of each of the models reviewed:

Models Based on Modified Architectures

Modified RNN-based imputers extend the basic RNN short-term temporal dependency bias by incorporating specialised mechanisms to address different EHR data characteristics. **GRUD** [22] aims to address the issue of irregular sampling by embedding an exponential decay assumption where past observations lose relevance at a fixed rate. The decay assumption also aligns the model with the recording patterns of high-frequency variables where both natural physiological decay and equipment-based MNAR patterns (e.g., disconnections) occur. However, GRUD does not explicitly model cross-sectional correlations, which are modelled in **BRITS** [20] via an additional fully-connected layer

in addition to the model's RNN backbone. BRITS further enhances the robustness of its temporal modelling by utilising bidirectional dynamics to capture temporal correlations in the forward and backward directions. Despite being one of the earliest EHR neural network imputers, BRITS' ability to robustly capture temporal and cross-sectional correlations has kept it a popular model today. **MRNN** [183] aims to address temporal complexity and irregular sampling by incorporating auxiliary variables to capture multi-resolution dependencies, acknowledging varying temporal granularities from minute-level vitals to daily assessments. While this approach improves temporal dependency modelling for a clinical category (labs, medications, procedures), it can oversimplify the cross-sectional correlations between modalities.

The rise of the Transformer architecture has led to the development of models aiming to take advantage of its global perspective to identify subtle, yet clinically significant patterns that might be overlooked by models with a narrower focus. To preserve the strict sequential integrity that defines clinical time-series [185, 23] and to capture equally useful short-term temporal associations [46], these models use different bespoke modifications. **GLIMA** [157] incorporates both intra-sequence and inter-sequence attention to capture local and global dependencies, **MTSIT** [181] introduces a modified attention mechanism incorporating temporal positional encoding to preserve sequential information. Finally, **SAITS** [49] employs a dual-view self-attention mechanism, where the first view focuses on temporal dependencies within each variable while the second view captures spatio-temporal cross-variable interactions.

TiledCNN [173] and **TimesNet** [175] transform time series data into two-dimensional tensors, enabling simultaneous extraction of spatial and temporal features through CNN architecture. This approach uncovers patterns that span multiple time points and variables within physiological signals, such as concurrent analysis of ECG readings, respiratory rates, and oxygen saturation levels [189]. However, these models introduce additional complexity in transforming and interpreting time series in 2D spaces, potentially obscuring the temporal sequence and causal relationships inherent in the data [160]. TSI-GNN

[55] extends GNNs to represent EHR missingness using a bipartite graph structure that captures spatiotemporal dependencies, where temporal edges connect the same variable across consecutive timestamps while feature edges link different variables within the same timestamp. Although the approach can theoretically capture EHR characteristics, it faces challenges in aligning the static nature of graphs with the dynamic nature of medical time series [73]. Additionally, constructing and interpreting these graph models requires significant computational resources and domain knowledge, limiting their scalability and practicality [176].

Models with a Generative Component

These models operate by generating a latent representation of the EHR time-series with the aim of achieving one that captures the different EHR data properties enabling robust imputation. **MIWAE** [104], **GP-VAE** [53], **HI-VAE** [116], **V-RIN** [115], **Shi-VAE** [11] and **Supnot-MIVAE** [81] adopt the VAE framework to map diverse data types and modalities. However, the success of a VAE model depends on its ability to create meaningful latent representations that align with their assumed distribution - a challenge with complex EHR data with numerous clinical subtleties. HI-VAE [116], MIWAE [104] and GP-VAE [53], attempt to use architectural bias to improve Gaussian-based predictions, but their performance has been reported to plummet with heterogeneity in observations and extended missingness typical of EHRs [187]. V-RIN [115] incorporates an uncertainty-aware Gated Recurrent Unit (GRU) to blend temporal dynamics with the Gaussian imputations. Supnot-MIVAE [81] extends this approach by introducing an additional classifier to refine the evidence lower bounds, enhancing imputation accuracy, while Shi-VAE [11] further expands these capabilities by including LSTMs for better temporal structure handling. While these hybrid VAE models bypass the distribution problem by incorporating temporal dynamics, they face significant challenges in producing clinically relevant outputs. Furthermore, the computational intensity for training VAEs,

Table 3.1 Chronological overview of neural network models specialised in medical time series imputation. **Component Bias:** the bias induced by the imputer's components—architecture and (optionally) generative framework.

Model	Year	Architecture	Component Bias	Uncertainty Quantification
Modified Single Architectures				
MRNN [183]	2017	RNN	Sequential	✘
GRUD [22]	2018	RNN	Sequential	✘
BRITS [20]	2018	RNN	Sequential	✘
GLIMA [157]	2020	Attention	Globality	✘
MTSIT [181]	2022	Attention	Globality	✘
SAITS [49]	2023	Attention	Globality	✘
Tiled CNN [173]	2015	CNN	Locality	✘
TimesNet [175]	2023	CNN	Locality	✘
TSI-GNN [55]	2021	GNN	Relational	✘
VAEs				
MIWAE [104]	2019	CNN	Locality, Stochasticity	✔
GP-VAE [53]	2020	CNN	Locality, Stochasticity	✔
HI-VAE [116]	2020	MLP	Stochasticity	✔
V-RIN [115]	2021	RNN	Sequential, Stochasticity	✔
Shi-VAE [11]	2022	RNN	Sequential, Stochasticity	✔
supnot-MIWAE [81]	2023	CNN, Attention	Locality, Globality, Stochasticity	✔
MDNs				
CDNet [98]	2022	RNN	Sequential, Mixture	✔
GAN				
VIGAN [147]	2017	CNN	Locality, Adversariality	✘
GRUI-GAN [100]	2018	RNN	Sequential, Adversariality	✘
E^2GAN [99]	2019	RNN	Sequential, Adversariality	✘
NAOMI [97]	2019	MLP	Adversariality	✘
US-GAN [109]	2021	RNN	Sequential, Adversariality	✘
Sim-GAN [123]	2022	CNN	Locality, Adversariality	✘
Diffusion				
CSDI [161]	2021	Attention	Globality, Gradualism	✔
SSSD [5]	2023	CNN	Locality, Gradualism	✔
CSBI [26]	2023	CNN, Attention	Locality, Globality, Gradualism	✔
DA-TASWDM [178]	2023	Attention	Globality, Gradualism	✘
Neural ODEs				
CRU [144]	2022	RNN	Sequential, Continuity	✔
CSDE [122]	2022	MLP	Continuity	✔

especially when integrating temporal dynamics, remains a barrier to their wide adoption for large medical datasets.

MDN imputers generate complex, non-Gaussian probabilistic distributions capturing multimodal clinical measurements. However, determining optimal mixture components remains challenging given EHR heterogeneity. **CDNET** [98] addresses this through a compound architecture integrating GRUs and regulated Attention Networks, enabling simultaneous modelling of temporal dependencies and feature distributions. This setup allows for robust handling of irregular sampling patterns whilst capturing the underlying multimodal distributions of clinical variables. While promising in theory, MDN models face implementation challenges due to computational complexity and difficulty in optimising parameters while maintaining clinical relevance [194].

GRUI-GAN [100], E^2**GAN** [99], **NAOMI** [97] and **US-GAN** [109] leverage GAN's adversarial process to generate realistic synthetic EHR time-series through adversarial training, potentially handling irregular sampling and temporal dependencies in EHR data. In order to achieve stable adversarial training dynamics, GRUI-GAN [100] incorporates modified GRUs in both the generator and discriminator, but faces difficulties in optimising the noise vector for generation. E^2GAN [99] tackles this limitation by incorporating a denoising autoencoder structure, while NAOMI [97] introduces a non-autoregressive approach to minimise error accumulation in extended sequences. While these hybrid GAN models demonstrate promising capabilities in generating synthetic EHR data, they face significant challenges in ensuring clinical reliability and avoiding mode collapse. US-GAN [109] addresses these issues by implementing a temporal reminder matrix and additional classification layers. The wide adoption of GAN-based models is hindered by their inability to quantify confidence, coupled with training instability.

Diffusion-based imputers operate by gradually denoising data through an iterative process, aiming to learn the irregular sampling patterns and temporal dependencies through reverse diffusion steps. The success of a diffusion model depends on achieving efficient computation while maintaining accurate temporal modelling. **CSDI** [161] attempts to

Table 3.2 Breakdown of the temporal and cross-sectional dependencies and reported missing data strategies of neural EHR imputers.

Model	Temporal Dependencies	Spatial Dependencies	Sparsity Handling	Missingness Mechanisms
Modified Architectures				
MRNN	Hierarchical	Weak cross-sectional	Multiple sampling frequencies	MAR
GRUD	Short-term with decay	None	Irregular sampling via decay	MCAR, MNAR
BRITS	Bidirectional	Cross-sectional via FC	No explicit handling	MAR
TimesNet	Local via 2D encoding	Local via convolution	Regular grid assumption	MAR
Tiled CNN	Multi-scale temporal	Via Gramian fields	Regular grid assumption	MAR
GLIMA	Global via attention	Global attention	Attention-based interpolation	MCAR, MAR
MTSIT	Global with locality	Global attention	Position-based attention	MCAR, MAR
SAITS	Global with local refinement	Global attention	Self-attention masking	MCAR
GNNs				
TSI-GNN	Graph-structured	Bipartite graph	Graph-based interpolation	MCAR
VAEs				
MIWAE	Latent encoding	Latent space	Probabilistic sampling	MCAR, MAR
GP-VAE	Gaussian process	Latent space	Gaussian process interpolation	MCAR, MAR, MNAR
V-RIN	GRU dynamics	Latent space	GRU-based handling	MCAR, MAR
HI-VAE	Latent encoding	Latent space	Probabilistic encoding	MCAR
Shi-VAE	LSTM dynamics	Latent space	LSTM-based handling	MAR
supnot-MIWAE	Local-global combined	Attention-based	Multi-scale handling	MNAR
MDNs				
CDNet	GRU-based	Attention-based	Mixture-based interpolation	-
GAN				
VIGAN	Local adversarial	CNN-based	No explicit handling	-
GRUI-GAN	Irregular modelling	GRU-based	GRU-based adaptation	-
E^2GAN	Denoising sequence	GRU-based	Denoising-based	-
NAOMI	Multiresolution	Multiresolution		
US-GAN	Reminder matrix	RNN-based	Temporal reminder	MCAR
Sim-GAN	Local patterns	CNN-based	No explicit handling	-
Diffusion				
CSDI	Gradual denoising	Conditional	Progressive refinement	MCAR, MAR, MNAR
SSSD	State-space evolution	Local structure	State-space modelling	MCAR, MAR, MNAR
CSBI	Bridge-based	Local-global	Bridge-based refinement	MAR
DA-TASWDM	Dynamic attention	Global attention	Attention-based handling	MAR
Neural ODEs				
CRU	Probabilistic states	None	ODE-based interpolation	MAR
CSDE	Markov dynamics	None	ODE-based interpolation	-

address this through transformer architectures but faces scalability issues due to quadratic complexity [150]. **SSSD** [5] tackles this limitation using structured state-space models, while **CSBI** [26] and **DA-TASWDM** [178] further enhance efficiency by integrating spatio-temporal dependencies and dynamic temporal relationships. While these models demonstrate promising capabilities in handling temporal dependencies, their high computational cost remains a barrier.

Finally, **CRU** [144] and **CSDE** [169] generate continuous data transitions through ODE-learned functions [122], addressing the challenge of irregular EHR sampling [138]. CRU [144] employs a linear stochastic differential equation (SDE) [169] within a latent space structure, integrating continuous-discrete Kalman filters with medical time series analysis. CSDE [122] introduces a probabilistic framework that enhances traditional dynamic models through Markov dynamic programming [64] and multi-conditional forward-backwards losses, enabling robust training and theoretical optimality. However, learning stable dynamics functions remains challenging with sparse and irregular EHR data, and like diffusion models, the computational complexity of neural ODEs continues to be a significant barrier.

3.1.1 The Current Benchmarking Paradigm of NN Imputers: Gaps & Tradeoffs

Having established the theoretical foundations of deep EHR imputers, I turn to examining how those models are benchmarked in the literature. The evaluation of deep imputation models presents unique challenges due to the inherent nature of missing data - we cannot directly measure accuracy on truly missing values. Instead, evaluation relies on simulating missing data conditions in controlled settings. The dominant approach to addressing this is through *masking*— a technique where certain data points are deliberately designated as missing during training and evaluation. Masking provides a controlled way to test how an

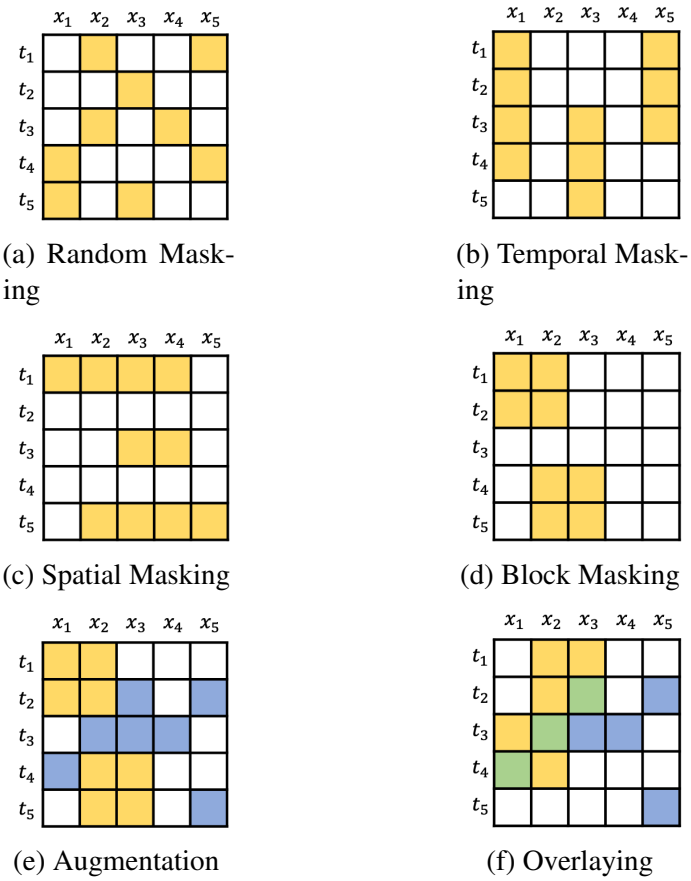
algorithm handles incomplete datasets and is thus essential for performance evaluation. The different patterns of data masking are illustrated in Figure 3.2.

Upon examining the current evaluation paradigm of neural imputers, the following limitations were identified. First, the simulation of missing data must align with the theoretical assumptions and capabilities of the models being tested. My review of the literature reveals significant discrepancies between how models are evaluated and the missingness mechanisms they are designed to handle. Second, there is considerable variation in how different models implement and report their evaluation procedures, making direct comparisons between models challenging. Finally, the preprocessing steps and experimental design choices that influence model performance are often underreported or inconsistently applied across studies. These limitations manifest most clearly in the use of masking — while the choice of masking strategy can significantly impact model performance, there is little standardisation in how masking is implemented or reported in the literature. Below, I provide a detailed discussion of the gaps identified:

Misalignment with Missingness Assumptions

As shown in Table 3.2, deep imputers have been designed to recognise different flavours of missingness (MCAR, MAR, MNAR). During experimental evaluation, however, all models shown in Table 3.2 use random masking (Figure 3.2 (a)) to generate missing datasets, predominantly producing MCAR scenarios and drastically oversimplifying the EHR dependencies as discussed in section 2.2.

Interestingly, the literature contains masking techniques that can capture spatio-temporal missingness patterns[38], including temporal masking (Figure 3.2 (b)), which captures missinngess patterns over time, spatial masking (Figure 3.2 (c)), which captures cross-sectional missingness and block masking (Figure 3.2 (d)), which combines the two to concurrently capture different flavours of temporal and cross-sectional correlations and



(a) Random Masking

(b) Temporal Masking

(c) Spatial Masking

(d) Block Masking

(e) Augmentation

(f) Overlaying

Fig. 3.2 Masking techniques and approaches demonstrated over a time-series of five features ($x_1 - x_5$) and five time points ($t_1 - t_5$). The yellow cells indicate those labelled as missing via masking: (a) **random masking:** points are randomly selected to be masked and used as ground truths; (b) **temporal masking:** consecutive time points within single features are masked, simulating real-world temporal recording patterns where feature recording frequencies vary; (c) **spatial masking:** entire features are masked across specific time periods, simulating scenarios where related measurements are missing together (e.g., multiple lab tests typically ordered as a panel); (d) **block masking:** contiguous blocks of data points are masked across both time and features, mimicking real-world situations where related measurements are missing together. In (e) augmentation and (f) overlaying, the blue cells indicate cells that are missing within the original data. In (e), the masked (yellow) cells have no overlap with the original missingness in the data. Green: masked data coming from both the original missingness and artificial missingness. In (f), overlaying masks cells from either the original missingness or simulates artificial missingness from non-missing data.

dependencies. Despite their direct applicability to biomedical domains, the only examples using spatio-temporal masking of time-series come from the traffic domain [91, 180].

The above problem is exasperated by the lack of information in published work. With the exception of BRITS [20] and CSDI [161], the use of random masking is not mentioned in the experimental design, and one must examine the accompanying code to discern it. While the use of random masking facilitates model evaluation, it contrasts with the complex and informative MNAR patterns observed in real-world EHRs [54] which many deep imputers have been designed to address. The discrepancy between the theoretical model and experimental evaluation technique therefore highly undermines a deep imputer's capacity, leaving it under-evaluated. Standardisation tools such as PyPOTS [48] offer unified masking functionalities, urging a shift towards more sophisticated and data-driven masking designs.

Under-reporting of Masking Pipelines

There are significant discrepancies and under-reporting of when masking is introduced during experimental evaluation. Data could be pre-masked before being ingested by the model, or masked dynamically during the training phase. Traditional pre-masking methods, while more straightforward, limit the model's training to incomplete datasets, reducing its ability to learn from the entire range of clinical features and associated dependencies. In contrast, adopting in-mini batch masking strategies promises a more dynamic approach by iteratively masking different subsets of the same dataset across training epochs. However, this approach risks overfitting, as the model may become too focused on the artificial missing patterns and fail to recognise the original data structures. Therefore, the decision of when masking is introduced can have a profound impact on the model's capacity to interpret the diverse missing patterns found in a given dataset [131]. Despite the potential impact on the results, this aspect of the experimental design is not reported in most deep

imputers discussed in this survey, except for BRITS and GRUD, which mask before training, and CSDI and STAITS, which use in-mini-batch masking during training.

Overlooked Design Decisions

There are other decisions that highly influence the resulting masked data but are not discussed in most of the deep imputation literature. An important issue is the methodology used to implement masking [130]. Generally, masking can be implemented using *overlaying* [48] or *augmenting* [29] as shown in Figures 3.2 e-f. Overlaying involves adding artificial missingness in addition to the original missingness the dataset contains, while augmentations only mask complete data, separating the artificial missingness generated from the original missingness. Although overlaying exposes the model to a broader array of missing data scenarios, leading to more robust training and effective imputation strategies, it requires complex evaluation processes and increases the risk of overfitting. On the other hand, augmenting simplifies the model's learning process by allowing it to learn from the artificially introduced missingness without interference from the original missing patterns, but may not fully equip the model to handle the real missingness patterns of the data. It is unclear how any of the deep imputers implement masking, creating a big gap in our understanding of the rigour of the evaluation techniques, especially in models designed to accommodate non-random EHR missingness.

3.2 Time Series Modelling in the Era of Foundation Models

The rapid advancements in large-scale models have reshaped the way we approach time series analysis, parallelling transformative progress in natural language processing and computer vision. The applicability of Large Language Models (LLMs) to the analysis of structured clinical data remains a subject of ongoing debate and active research [134, 188].

Several studies have raised critical questions regarding their effectiveness, particularly within the context of predictive tasks [120, 167]. A comparative analysis is found in [14] examined the performance of several state-of-the-art LLM-based time-series models, including TimeLLM [72], LLaMA [188], and GPT4TS [72]. A recent taxonomy [74] distinguishes between Large Language Models for Time Series (LLM4TS) and Pre-trained Foundation Models for Time Series (PFM4TS). This division captures a fundamental distinction: the adaptation of existing language models to temporal data versus the creation of dedicated foundation models tailored to time series tasks. However, this binary distinction does not account for the growing prevalence of hybrid approaches that blend these paradigms. On the other hand, a more granular categorisation found in [110] divides methodologies into four categories:

1. Direct utilisation of LLMs.

2. Construction of general-purpose time series foundation models.

3. Development of domain-specific temporal models.

4. Creation of multimodal frameworks integrating text and time series data.

Regardless of the categorisation adopted, the fundamental question remains: are LLMs Necessary for Time Series Analysis? I argue that the usefulness of LLMs in time series analysis depends on two key factors:

1. **Sufficiency of Clinical Information:** In a considerable number of clinical prediction tasks, structured clinical data alone may provide sufficient information for achieving high accuracy [196, 172], making the sophisticated language reasoning capabilities of LLMs redundant.

2. **Human-Readable Interpretability:** When the goal of the analysis is to generate insights readily interpretable by human clinicians, LLMs can offer a significant advantage by presenting insight in human language [52, 193].

These factors suggest that the decision to use LLMs should be task-specific rather than a default choice. Here, the debate between general-purpose and domain-specific models becomes particularly relevant. General models, such as GPT-based architectures [191], are designed to handle various tasks but may underperform in specialised domains due to their lack of domain-specific inductive biases [155, 89]. In contrast, domain-specific models can incorporate tailored architectures and training strategies to exploit the unique characteristics of time series data [121, 28]. This is particularly evident in EHR analysis, where models specifically designed for clinical time-series have demonstrated superior performance in tasks such as disease progression modelling [186] and patient trajectory prediction [84]. The key tradeoffs identified are:

- **Performance vs. Generality**: Domain-specific models often achieve superior performance on specialised tasks, while general models offer broader applicability.
- **Training Complexity**: Fine-tuning general models for domain-specific tasks requires careful optimisation to avoid losing their general capabilities, as highlighted in [40].
- **Maintainability**: Domain-specific models may require frequent updates to accommodate new data, while general models offer more flexibility but risk overgeneralisation.

Therefore, while LLMs hold promise for enhancing human interpretability and cross-modal reasoning, their role in time series analysis should be critically assessed. Domain-specific models and models specialised in time-series provide viable alternatives, particularly for tasks where temporal data alone suffices or where interpretability is not required. Generally, however, the adaptation of LLMs for the analysis of structured EHR data presents several unique challenges:

- **Representation of Clinical Values:** The conversion of diverse clinical data, encompassing numerical measurements, categorical variables, and complex temporal

relationships, into a format that is effectively processed by LLMs while preserving their clinical meaning is a challenge [108, 146]. This transformation must carefully balance the need for compatibility with LLM architectures and the preservation of the meaning inherent in clinical values. Loss of information during this conversion can significantly impact the accuracy and reliability of the subsequent analysis.

- **Clinical Context:** Maintaining the crucial medical context and the relationships between different clinical measurements is essential for accurate interpretation [19, 195]. LLMs, by their nature, process sequential data. Preserving the contextual information, which may not be explicitly encoded in the sequential order, is crucial for avoiding misinterpretations. The relationships between different variables might be implicit and require careful consideration during the data representation process.

- **Temporal Understanding:** The temporal dimension of patient data is also a critical aspect that must be faithfully preserved within the LLM's sequential processing framework [65, 50]. The order of events and the timing of clinical measurements are often vital for accurate diagnosis and prognosis. Failing to accurately represent the temporal information can lead to flawed conclusions and inaccurate predictions. This requires carefully designed methods to encode time-dependent relationships.

3.2.1 Tokenisation Strategies and Their Implications: Rethinking Semantic Transferability

This section has so far presented an overview of the applicability and challenges in repurposing LLMs for structured medical time-series. There, however, remains the fundamental issue in large-scale time series modelling: how do we capture and transfer semantic meaning in temporal patterns? This issue is directly related to the process of tokenisation, which involves breaking down raw data into meaningful units (tokens) suitable for processing by LLMs [50, 62]. The fundamental challenge in time series tokenisation lies in the lack

of semantic consistency across contexts. Unlike the relatively well-defined vocabulary of natural language, where words or subwords serve as readily identifiable tokens with relatively stable semantic meanings, clinical data often exist within a continuous, mixed, and highly context-dependent space [164, 1]. For instance, a rising sequence of values might indicate a worsening medical condition, or progress towards stability, depending on the context. This variability introduces what can be termed the "temporal semantic gap": the same numerical value can represent completely different clinical meanings depending on the specific context, the patient's medical history, and the temporal sequence of events. The above significantly complicates the development of universally applicable tokenisation strategies, and indicates that the effectiveness of LLMs in EHR analysis is fundamentally dependent on the choice of tokenisation strategy.

Additionally, time series data differ from language in their structural nature. While language tokens are inherently discrete, and structured from a finite set of symbols, time series data often exhibit continuous and multi-scale characteristics. This continuous nature, coupled with domain variability, complicates the mapping of raw temporal patterns to semantically meaningful tokens. To bridge this gap, a key technique involves vector quantisation (VQ), which maps continuous temporal information into a compressed, finite representation space, enabling discrete tokenisation. By discretising latent representations, VQ serves as a bridge between the continuous nature of raw data and the discrete symbolic structures required for downstream tasks.

The above challenges highlight the importance of tokenisation strategies that go beyond simply encoding local and global patterns. They must also align temporal representations with semantic meanings that generalise across scenarios. Unlike multimodal data like images and audio, where continuous tokenisation (representation learning) has dominated, time series analysis often benefits from the introduction of discrete tokenisation approaches that draw inspiration from the structural principles of language, balancing inductive biases and task-specific requirements. Current tokenisation approaches can be broadly categorised into three main types:

Point-Level Tokenisation Point-level tokenisation focuses on discretising individual time points. This approach directly maps each raw data point into a discrete representation, exemplified by Chronos [7]. For example:

$$\text{Token}_t = Q(x_t) \tag{3.1}$$

where Q represents a quantisation function. While preserving fine-grained temporal granularity, this method introduces several challenges:

- **Value-Semantic Disconnect**: Similar values in different contexts might represent entirely different semantic meanings.
- **Context Loss**: The focus on point-wise representation often misses broader temporal patterns.
- **Granularity Trade-Off**: The choice of quantisation levels directly impacts the balance between information preservation and model complexity.

Segment-Level Tokenisation Segment-level tokenisation groups subsequences of the data, or "patches," into tokens. This strategy is inspired by Vision Transformers [44] and applied to time series through methods like PatchTST [118]:

$$\text{Token}_i = f(\text{Sequence}[i : i+k]) \tag{3.2}$$

where k represents the batch size. Segment-level tokenisation captures local patterns within fixed-length intervals, but it also faces limitations:

- **Semantic Ambiguity**: Patches might represent different semantic concepts depending on the context.
- **Boundary Effects**: Fixed patch sizes can disrupt meaningful patterns across segment boundaries.

- **Dependence on Latent Representations**: Many patch-based methods require further processing, such as embedding or vector quantisation, to generate meaningful tokens.

Traditional works such as Elastic Product Quantization (EPQ) [136] propose a flexible partitioning and quantization strategy tailored for time series data. EPQ dynamically adjusts partition boundaries to better accommodate the inherent variability and multi-scale nature of time series, preserving essential semantic patterns while achieving compact representations. While the symbolic approach SAX [92] allows one to run certain data mining algorithms on the efficiently manipulated symbolic representation, while producing identical results to the algorithms that operate on the original data. However, their reliance on traditional algorithms may limit their ability to capture complex semantic nuances, particularly in dynamic and high-dimensional contexts.

Recent advancements, such as GPT4TS [191] and TEMPO [18], incorporate dynamic normalisation and domain-specific knowledge to enhance segment-level tokenisation. Integrating techniques like vector quantisation [158] into these frameworks can further address challenges related to semantic ambiguity and boundary effects.

Instance-Level Tokenisation Instance-level tokenisation generates tokens from an entire sequence or its latent representation, focusing on capturing global patterns. Methods such as TOTEM [158] use end-to-end learning to map sequences into a latent space and discretise them. This approach offers strong global representation capabilities but introduces unique challenges:

- **Dynamic Dependency**: The tokenisation process is tightly coupled with end-to-end task-specific learning, which may reduce generalisability.

- **Semantic Inconsistency**: Fixed latent token dictionaries may fail to capture the diversity of temporal contexts.

- **Interpretability Trade-Off**: While global patterns are well captured, these tokens often lack explicit interpretability.

Traditional neural network architectures, such as RNNs and CNNs, encode strong inductive biases that are well-suited for specific types of data. RNNs emphasise temporal locality, while CNNs excel at spatial locality. Transformers, by contrast, rely on weaker inductive biases, instead learning relationships through attention mechanisms in B.5.

This flexibility underscores the importance of effective tokenisation strategies. As [24] highlights, tokenisation directly impacts the model generalizability. Each tokenisation strategy presents distinct trade-offs between granularity, semantic consistency, and interpretability. Point-level approaches offer fine-grained control but struggle with contextual meaning; segment-level methods balance local patterns with boundary challenges; while instance-level strategies capture global patterns at the cost of interpretability. The evolution of these approaches, from traditional quantization methods to recent neural techniques, reflects the ongoing effort to bridge the gap between continuous temporal data and discrete symbolic representations. The key challenge remains developing tokenisation strategies that can effectively balance these competing demands while maintaining semantic consistency across different domains and temporal scales.

3.3 Patient Health Trajectory Modelling

The concept of patient health trajectories has emerged as a powerful framework for representing and analysing longitudinal healthcare data. By modelling a patient's medical history as a temporal sequence of events, this approach enables the application of advanced sequence modelling techniques to healthcare analytics whilst preserving the temporal relationships inherent in clinical data. This section examines the evolution of health trajectory modelling, with particular emphasis on recent developments in transformer-based architectures and their implications for long-term temporal understanding.

3.3.1 Early Developments in Health Trajectory Modelling

The adaptation of transformer architectures to electronic health records (EHRs) marked a significant advancement in health trajectory modelling. BEHRT [89] pioneered this approach by introducing a multi-embedding strategy that captures various aspects of patient history, including events, visits, age, and positional information. This work demonstrated the potential of transformer-based architectures in understanding complex medical histories, leading to improved prediction of future health events.

Building upon this foundation, Med-BERT [133] demonstrated the scalability of such approaches by successfully training on EHR data from 20 million patients. This achievement validated the feasibility of large-scale health trajectory modelling and established the potential for developing comprehensive medical prediction systems.

3.3.2 Advanced Temporal Understanding

A critical advancement in health trajectory modelling came with CEHR-BERT [121], which introduced sophisticated strategies for integrating temporal information. By incorporating artificial time markers alongside traditional embeddings, this model achieved superior performance across multiple risk prediction tasks. This work highlighted the importance of explicit temporal representation in health trajectory understanding.

The introduction of Clinical Language Model Based Representations (CLMBR) [155] represented a paradigm shift in approach. By treating medical codes as words within a patient timeline, CLMBR demonstrated the effectiveness of applying natural language processing techniques to health trajectory modelling. This analogy between clinical sequences and natural language has proven particularly fruitful, enabling the application of established NLP techniques to healthcare data.

3.3.3 The Challenge of Long-Term Dependencies

A fundamental challenge in health trajectory modelling lies in handling extended patient histories, particularly those spanning thousands of records. As highlighted by Hi-BEHRT [90], the traditional transformer architecture's limitation to sequences of 512 tokens presents a significant bottleneck for comprehensive patient history analysis. This constraint is particularly problematic in healthcare, where historical events may have long-term implications for patient outcomes. While Hi-BEHRT proposed a hierarchical architecture to address this limitation by extracting local temporal features before higher-level processing, more efficient solutions have emerged from recent developments in large language models.

Modern advances in handling long sequences have revolutionised the approach to this challenge, detailed in Section 2.7. Key innovations include Flash Attention [34], which dramatically improves attention computation efficiency through IO-aware block-sparse algorithm design, and advanced positional encodings such as Rotary Position Embeddings (RoPE) that enable better generalisation to longer sequences. These techniques have become foundational components in state-of-the-art language models, enabling effective processing of sequences far beyond traditional length limitations while maintaining computational efficiency.

3.3.4 Towards Universal Health Trajectory Representation

A critical advancement in standardising health trajectory analysis has been the development of the Medical Event Data Standard (MEDS) [10], a comprehensive framework for structuring and normalising healthcare event sequences. MEDS addresses a fundamental challenge in healthcare analytics: the lack of standardisation in how medical events are represented across different healthcare systems. By providing a unified event-based data standard, MEDS enables:

- Standardised representation of medical events across different healthcare systems

- Enhanced interoperability between different health information technology systems
- Streamlined integration of diverse healthcare datasets for machine learning applications
- Consistent preprocessing and modelling workflows across different research initiatives

The significance of MEDS extends beyond mere data standardisation; it establishes a foundation for reproducible research and facilitates collaboration across the healthcare analytics community. The MEDS framework represents a crucial step towards enabling systematic evaluation and comparison of different health trajectory modelling approaches, with the potential for integrating future architectural innovations into a standardised ecosystem.

Building upon this standardisation effort, the Event Stream GPT library [105] represents another significant step towards standardising the preprocessing and modelling of continuous-time event sequences. This development leverages MEDS-compatible data structures to provide unified representation frameworks for healthcare analytics.

The feasibility of large-scale trajectory modelling has been further demonstrated by life2vec [141], which successfully processed life event sequences for 6 million Danish citizens. This work not only validated the scalability of trajectory-based approaches but also demonstrated their utility in understanding complex life outcomes, including health-related predictions.

Drawing parallels with multilingual language models [66, 33], multilingual models successfully align semantically equivalent phrases across languages despite lexical and syntactic disparities. Similarly, time series-text alignment requires handling domain-specific variability and context-dependent semantics at multiple levels of abstraction. Recent developments in zero-shot health trajectory prediction [135] have further built upon these standardisation efforts, introducing universal tokenisation formats for patient timelines. This work suggests that standardised representations could enable "translation"

between different healthcare systems' data. This aligns with the goals of MEDS, suggesting that effective health trajectory models can operate at multiple levels of abstraction.

These developments suggest that future health trajectory models will increasingly draw upon innovations in language model architecture while addressing the unique challenges of clinical data. The convergence of efficient computation techniques, sophisticated temporal understanding, and standardised representation frameworks promises to enable more comprehensive and accurate patient trajectory analysis.

3.4 Chapter Summary

This chapter reviewed the literature along three complementary axes. The first one focused on neural network-based imputation models that have been designed to operate within the context of EHR time series. Our take on the literature indicates that the effectiveness of neural network approaches for EHR data imputation is fundamentally tied to how well their designs align with EHR characteristics discussed in Chapter 2. While architectural choices (RNNs, Transformers, CNNs, GNNs) and generative frameworks (VAEs, GANs, Diffusion Models) each offer distinct advantages in handling specific EHR properties, no single model perfectly addresses all challenges inherent to healthcare data. Most approaches require additional design modifications to better align with EHR complexities, whether in handling short and long-temporal dependencies, cross-sectional correlations, or irregular sampling patterns. This insight motivates the studies presented in Chapters 4, 5, and 6.

The second axis explored the emerging role of foundation models in time series analysis, particularly examining the challenges and opportunities in adapting language model architectures to temporal healthcare data. The review presented in this chapter highlighted critical considerations in tokenisation strategies and semantic transferability, demonstrating that while large language models offer promising capabilities for healthcare analytics, their application requires careful consideration of temporal logic and domain-specific requirements. The analysis of recent studies questioning the necessity of LLMs

for time series tasks underscores the importance of developing hybrid approaches that combine the interpretability benefits of language models with the precision of specialised temporal architectures.

The third axis examined the evolution of patient health trajectory modelling, tracking the progression from early transformer-based approaches to modern frameworks for handling long-term dependencies in healthcare data. The emergence of standardised representations through initiatives like MEDS, coupled with advances in efficient computation techniques, suggests a promising direction for more comprehensive and accurate patient trajectory analysis. This insight motivates the development of the novel approaches in Chapter 7.

This comprehensive review of both traditional imputation methods and emerging paradigms in medical time series analysis provides the foundation for the novel contributions presented in subsequent chapters.

Chapter 4

Benchmarking Neural EHR Imputers: Gaps & Tradeoffs

This chapter is the beginning of my exploration of neural EHR imputers, which forms the bulk of my contributions in this thesis. Having established the theoretical foundations of neural EHR imputers and surveyed the literature and current evaluation paradigm, this chapter aims to validate the findings through experimental evaluation. More specifically, the chapter aims to quantify the key challenges and methodological requirements to inform the design of the models I present in subsequent chapters. Through experiments, I investigate two aspects that directly impact the usability of deep imputers formulated as two key objectives:

1) To quantify the relationship between model complexity and imputation performance, examining whether increasingly sophisticated architectures translate to proportional improvements in accuracy.

2) To experimentally assess the effectiveness of current evaluation and benchmarking practices discussed in 3.1.1, particularly focusing on how well experimental frameworks align with real-world missingness patterns in clinical data and measure the sensitivity of different deep imputers to increased difficulty in experimental settings.

4.1 Experimental Design

I perform four experiments designed to satisfy our evaluation objectives stated at the beginning of this chapter:

1) Experiment 1: Performance-Complexity Tradeoffs: Examines the relationship between model complexity and imputation accuracy by analysing performance against model size (parameter count), runtime efficiency, and theoretical complexity to determine whether architectural sophistication translates to proportional gains in accuracy.

2) Experiment 2: Impact of Missingness Mechanisms: Assesses model robustness and performance variations across different missingness patterns of point-based masking, temporal masking, and block masking.

3) Experiment 3: Masking Implementation Design: Examines how different masking implementation choices affect model learning, including: the timing of masking application (pre-masking vs in-minibatch), the masking strategy (overlaying vs augmentation), and the timing of normalisation.

4) Experiment 4: Downstream Task Evaluation: Investigates how different imputation approaches affect the performance of downstream clinical prediction tasks using both modular and end-to-end training strategies.

4.1.1 Implementation and Experimental Details

To ensure standardised evaluation and full control over experimental conditions, all experiments are performed using the `PyPOTS` library discussed in Chapter 2 and the widely-known Physionet Cardiology Challenge 2012 benchmarking dataset (PhysioNet 2012) [152]. `PyPOTS` incorporates 41 neural network models spanning imputation and downstream tasks. Of these, 8 are Neural EHR imputers designed and evaluated on EHR benchmarks (the full list of neural EHR imputers is provided in Tables 3.1 and 3.2) and form the basis of the experimental evaluation.

Models Evaluated: The subset of neural imputers incorporated within PyPOTS spans sequential models (BRITS, GRUD, MRNN), convolutional models (TimesNet), attention-based (SAITS), diffusion-based (CSDI), variational (GP-VAE), and adversarial approaches (US-GAN). While RNN-based architectures, being the earliest deep imputers, are well-represented in PyPOTS, the inclusion of highly-performing models using more recent approaches such as Neural ODEs into the package is ongoing work pending code verification and compatibility testing. Nevertheless, this diverse set of architectures enables broad evaluation across different modelling paradigms, allowing us to assess how theoretical advantages translate to practical performance.

Experimental Conditions and Environment: The four experiments relied on PyPOTS' PyGrinder[1] library to simulate different masking conditions and BenchPOTS[2] library control aspects of benchmarking pipeline. In all experiments, I mask 10% of the data as ground truths. Because the Physionet already contains a high proportion of missing values ($\sim$ 80%), increasing the missingness rates in our experiments would only be possible if point-based (random) masking is used. Therefore, 10% masking is used throughout the four experiments.

All experiments were conducted on a machine equipped with an NVIDIA A100 GPU, 80GB of RAM. Each model was trained with its recommended hyperparameters using PyGrinder and tuned through 5-fold cross-validation. The preprocessing scripts, model configurations, and training code are publicly available on GitHub[3], promoting transparency and reproducibility for the research community.

[1] https://pypots.com/ecosystem/#PyGrinder
[2] https://github.com/WenjieDu/BenchPOTS
[3] https://github.com/LinglongQian/Mask_rethinking

4.2 Experiment 1 Results: Performance Efficiency Trade-offs

The performance reported by the 8 models using 10% point-based missingness is shown in Table 4.1, along with the theoretical training complexity of the models derived from the original papers. Figure 4.1 shows the results of comparing the models' MAE with the number of parameters and actual training time (measured in hours). Examining the relationship between model complexity and imputation performance reveals patterns that challenge assumptions about architectural scaling.

Table 4.1 Performance and Complexity Analysis of Deep Imputation Models on PhysioNet 2012: **MAE**: Mean Absolute Error; **MSE**: Mean Squared Error; **Training Time Complexity**: The computational complexity of one complete training iteration (forward pass, loss computation, backward pass, and parameter updates) as a function of input dimensions. **Training Space Complexity:** Memory needed during training (including gradients, intermediate activations, batches) **N:** sequence length; **T:** number of features/variables; **S:** number of diffusion steps (for CSDI)

Model	MAE	MSE	Time Complexity	Space Complexity
BRITS	0.297 (0.001)	0.338 (0.001)	$O(NT)$	$O(2NT)$
MRNN	0.708 (0.029)	0.921 (0.044)	$O(NT)$	$O(NT)$
GRUD	0.450 (0.004)	0.492 (0.006)	$O(NT)$	$O(NT)$
TimesNet	0.353 (0.003)	0.355 (0.004)	$O(NT)$	$O(NT)$
SAITS	0.257 (0.019)	0.276 (0.018)	$O(N^2T)$	$O(N^2T)$
CSDI	0.252 (0.004)	0.313 (0.039)	$O(SN^2T)$	$O(N^2T)$
GP-VAE	0.445 (0.006)	0.499 (0.011)	$O(N^3)$	$O(N^2)$
US-GAN	0.310 (0.003)	0.318 (0.005)	$O(NT)$	$O(NT)$

CNN-based TimesNet, despite having the largest parameter count (64.9 million parameters), achieves only moderate performance (MAE 0.353), underperforming simpler architectures (BRITS and US-GAN). In contrast, the transformer-based SAITS achieves excellent performance (MAE 0.257) with 44.3 million parameters and surprisingly fast training time (1.02 hours), despite its quadratic complexity. As expected, the single-distribution bias of GP-VAE shows relatively lower performance (MAE 0.445) despite its elegant theoretical foundation. Moreover, despite having 0.5 million parameters, its

training time is of 10.59 hours is comparable to that the simpler and poorly-performing MRNN model (MAE: 0.708, number of parameters: 1.6 millions, training time: 10.93 hours). In comparison, GRUD has a comparable MAE to GP-VAE, but with the advantage of a small number of parameters (0.1 million parameters) and faster training (4.78 hours).

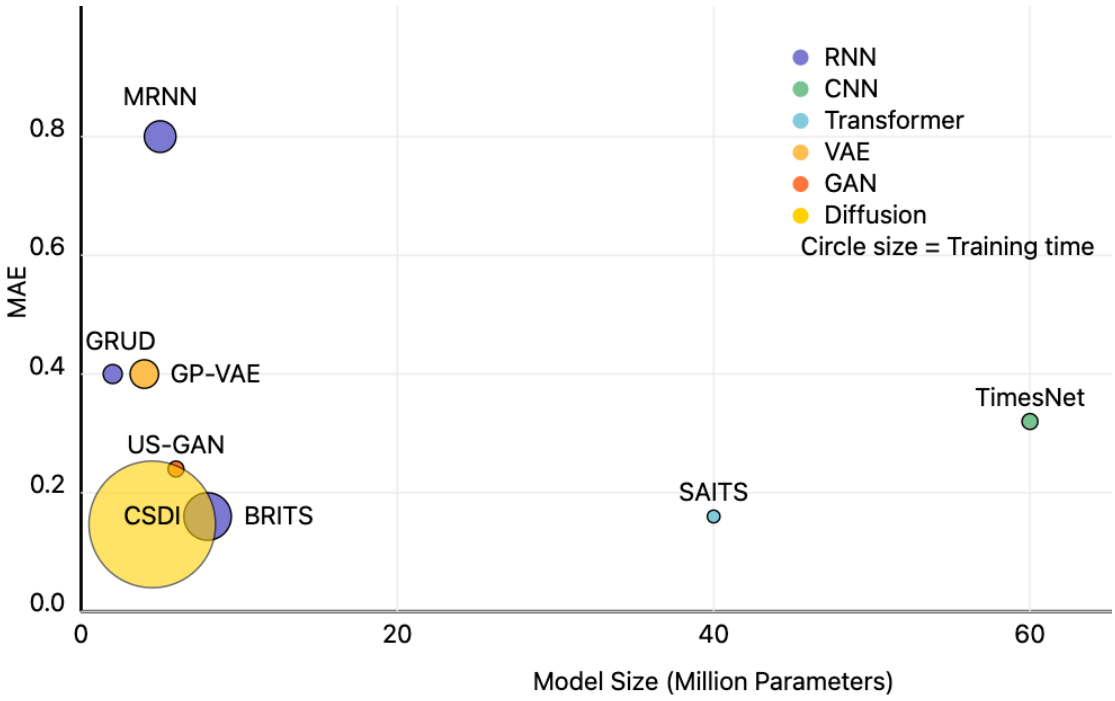


Fig. 4.1 Performance Efficiency of the eight models.

CSDI exemplifies how *architectural innovation can outweigh raw model size*, achieving the best accuracy (MAE 0.252) with only 1.5M parameters. However, this comes with a significant computational cost - 491 hours of training time compared to SAITS's 1.02 hours. This dramatic difference in computational requirements (nearly 500x) highlights the complex trade-offs in model selection and is in line with the fundamental nature of diffusion models, which require multiple forward passes and iterative denoising during training, which is reflected in their time complexity of $O(SN^2T)$, where S is the number of diffusion steps.

BRITS presents an interesting case in the complexity-performance trade-off: with 9.1M parameters, it achieves strong performance (MAE 0.297), only being outperformed by SAITS and CSDI (much larger models). However, BRITS' training time is significantly

large, especially given its relatively modest size (20.06 hours). This apparent inefficiency in RNN-based models like BRITS is a result of their sequential computations - RNNs process data points sequentially, making them inherently difficult to parallelise. While their theoretical complexity is linear ($O(NT)$), each time step must wait for the completion of previous steps. This suggests that theoretical complexity alone does not determine practical efficiency, and architectural innovations in newer models can overcome their higher computational complexity through better parallelisation and more efficient parameter utilisation.

4.3 Experiment 2 Results: The Effect of Missingness Mechanisms

This experiment assessed the effect of different masking techniques (point, subsequence or block-based masking) on model performance. The results shown in Figure 4.2 reveal that performance progressively degrades as masking patterns become more complex, with block masking consistently yielding the highest MAE scores. This general trend aligns with the increasing complexity of missingness patterns: point masking represents simple MCAR scenarios, subsequence masking captures time-dependent patterns, and block masking introduces both temporal and cross-sectional dependencies. Our results suggest that while no model is immune to the challenges of complex missingness patterns, architectures incorporating dedicated components to capture short and long-temporal dependencies and cross-sectional dynamics demonstrate better stability than those focused on local patterns or specific distributional assumptions. Specifically:

BRITS, CSDI and TimesNet show minimal degradation (MAE ranges 0.297-0.32, 0.25-0.28 and 0.33-0.36 respectively), validating their designs and demonstrating that their dedicated components effectively capture both temporal and cross-variable interactions even under complex missingness. SAITS exhibits a unique pattern, being the only model

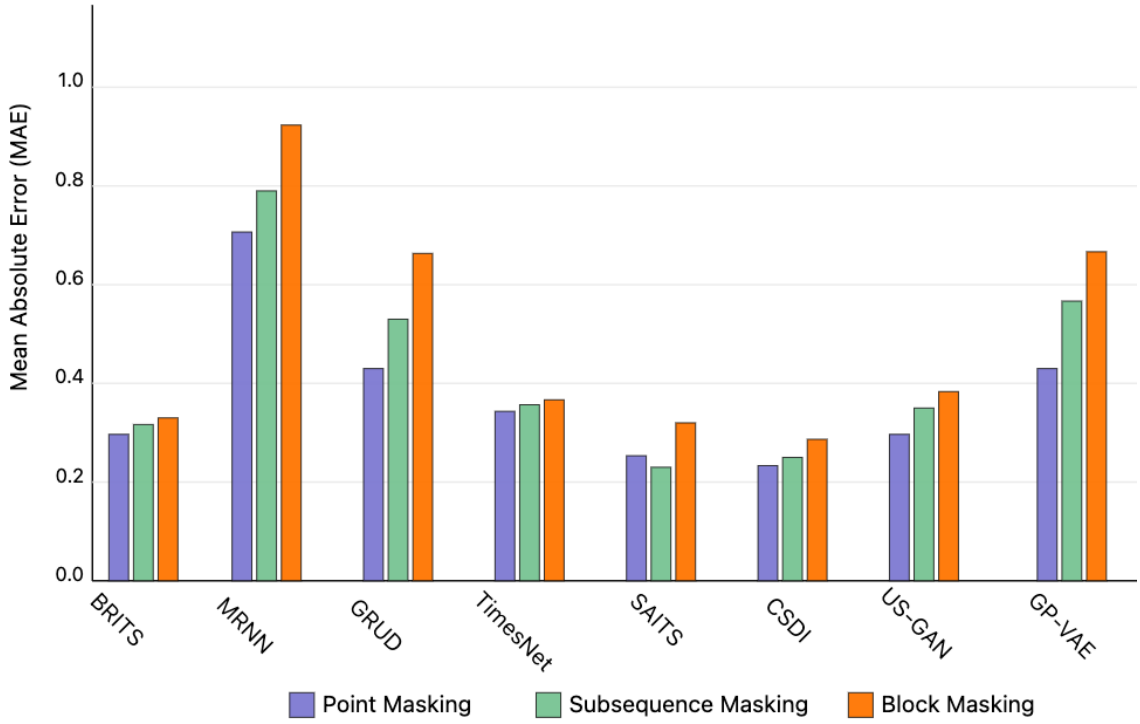


Fig. 4.2 The effect of different masking strategies on model performance measured in MAE.

with a lower MAE for sequence masking compared to point masking, demonstrating that its dual-view attention mechanism effectively captures short and long-term temporal dependencies. Both SAITS and US-GAN show moderate degradation to block missingness (MAE ranges 0.24-0.31 and 0.31 to 0.37 respectively), suggesting their dedicated components provide some robustness to complex missingness. GRUD, MRNN and GP-VAE show the highest sensitivity to masking complexity (MAE ranges: 0.45-0.65, 0.708-0.90 and 0.445-0.65 respectively), suggesting that their design assumptions are not representative of the complexities of EHR data missingnes and confirming known VAE limitations with heterogeneous data.

4.4 Experiment 3 Results: Masking Implementation Strategies

This experiment assessed three key aspects of masking implementation: **a)** the timing of masking application, comparing pre-masking the dataset against dynamic masking during training (mini-batch), **b)** the masking strategy, contrasting overlaying versus augmentation approaches, and **c)** the timing of normalisation with respect to masking (before or after).

The results shown in Table 4.2 reveal several patterns across these implementation choices. Mini-batch masking generally outperforms pre-masking approaches across most models. SAITS demonstrated superior performance with mini-batch masking, achieving the lowest MAE of 0.206 under overlay and 0.211 under augmentation strategies. CSDI showed similar effectiveness with mini-batch masking (MAE 0.226), though it exhibited some sensitivity to masking variability as evidenced by higher MAE (0.253) under overlay pre-masking. BRITS showed consistent performance across different strategies (MAE range 0.254-0.263), suggesting its bidirectional architecture provides robustness to masking implementation choices. TimesNet showed moderate performance with higher MAE values (around 0.290), while GRUD, GP-VAE and MRNN underperformed significantly with MAE exceeding 0.4, reflecting their limitations in capturing high-dimensional dependencies. Notably, the timing of normalisation showed minimal impact on performance across most models, suggesting this is a less critical design decision.

4.5 Experiment 4 Results: End-to-End Task Evaluation

Table 4.3 shows the classification results reported in PR_AUC and ROC_AUC for the 8 models using 10% point missingness. We used three types of classifiers for a broader comparison of the results. The first classifier uses the imputed data generated by the models, feeding those into an XGBoost classifier. The other two classifiers take an end-to-end approach by adding a classification layer to the models' architectures and differ by the

Table 4.2 Performances with different imputation methods on Physionet 2012 dataset. **NBM:** Normalisation Before Masking; **NAM:** Normalisation After Masking. Baselines included: **LOCF**: Last Observation Carried Forward, **Median** and **Mean**.

Model	Size	Augmentation Mini-Batch Mask NBM		Augmentation Pre-Mask NBM		Augmentation Pre-Mask NAM	
		MAE ↓	MSE ↓	MAE ↓	MSE ↓	MAE ↓	MSE ↓
SAITS	43.6M	**0.211±0.003**	**0.268±0.004**	0.267±0.002	0.287±0.001	0.267±0.007	0.290±0.001
GRUD	0.1M	0.422±0.001	0.474±0.003	0.483±0.002	0.401±0.005	0.483±0.002	0.403±0.003
TimesNet	44.3M	0.289±0.014	0.330±0.019	0.288±0.002	0.278±0.007	0.290±0.002	0.279±0.005
CSDI	0.3M	0.239±0.012	0.759±0.517	**0.237±0.006**	**0.302±0.057**	**0.241±0.017**	**0.430±0.151**
GPVAE	2.5M	0.425±0.011	0.511±0.017	0.399±0.002	0.402±0.004	0.396±0.001	0.401±0.004
US-GAN	0.9M	0.298±0.003	0.327±0.005	0.294±0.003	0.261±0.004	0.293±0.002	0.261±0.003
BRITS	1.3M	0.263±0.003	0.342±0.001	0.257±0.001	0.256±0.001	0.257±0.001	0.258±0.001
MRNN	0.07M	0.685±0.002	0.935±0.001	0.688±0.001	0.899±0.001	0.690±0.001	0.901±0.001
LOCF	/	0.411±5.551	0.613±0.0	0.404±0.0	0.506±0.0	0.404±0.0	0.507±0.0
Median	/	0.690±0.0	1.049±0.0	0.690±0.0	1.019±0.0	0.691±0.0	1.022±0.0
Mean	/	0.707±0.0	1.022±0.0	0.706±0.0	0.976±0.0	0.706±0.0	0.979±1.110
Model	**Size**	**Overlay Mini-Batch Mask NBM**		**Overlay Pre-Mask NBM**		**Overlay Pre-Mask NAM**	
		MAE ↓	**MSE ↓**	**MAE ↓**	**MSE ↓**	**MAE ↓**	**MSE ↓**
SAITS	43.6M	**0.206±0.002**	**0.227±0.005**	0.274±0.006	0.326±0.005	0.271±0.006	0.325±0.004
GRUD	0.1M	0.419±0.004	0.422±0.007	0.489±0.002	0.436±0.002	0.490±0.002	0.436±0.003
TimesNet	44.3M	0.273±0.011	0.242±0.018	0.293±0.003	0.290±0.011	0.291±0.003	0.288±0.007
CSDI	0.3M	0.226±0.010	0.279±0.051	**0.253±0.005**	**0.461±0.074**	**0.239±0.006**	**0.344±0.109**
GPVAE	2.5M	0.427±0.006	0.453±0.008	0.412±0.007	0.484±0.013	0.420±0.009	0.489±0.008
US-GAN	0.9M	0.295±0.002	0.261±0.007	0.297±0.002	0.284±0.005	0.299±0.003	0.287±0.004
BRITS	1.3M	0.254±0.001	0.265±0.001	0.262±0.000	0.288±0.003	0.263±0.001	0.294±0.003
MRNN	0.07M	0.682±0.000	0.905±0.001	0.685±0.001	0.926±0.002	0.684±0.001	0.923±0.002
LOCF	/	0.411±0.0	0.532±0.0	0.408±0.0	0.540±0.0	0.409±0.0	0.540±0.0
Median	/	0.687±0.0	1.019±0.0	0.686±0.0	1.030±0.0	0.686±0.0	1.030±0.0
Mean	/	0.705±0.0	0.990±1.110	0.702±0.0	1.001±0.0	0.702±0.0	1.000±0.0

type of network used. While the second classifier uses an RNN layer for classification, the third classifier uses a transformer layer.

Table 4.3 Classification performance on the PhysioNet 2012 dataset with 10% missingness rate.

	PR_AUC			ROC_AUC		
Model	**w XGB**	**w RNN**	**w Transformer**	**w XGB**	**w RNN**	**w Transformer**
SAITS	0.490 (0.000)	0.274 (0.054)	0.277 (0.039)	0.851 (0.000)	0.658 (0.076)	0.668 (0.037)
BRITS	0.455 (0.000)	0.315 (0.069)	0.270 (0.068)	0.841 (0.000)	0.667 (0.066)	0.646 (0.074)
MRNN	0.346 (0.000)	0.232 (0.013)	0.219 (0.012)	0.760 (0.000)	0.611 (0.008)	0.619 (0.015)
GRUD	0.423 (0.000)	0.373 (0.064)	0.392 (0.058)	0.840 (0.000)	0.721 (0.075)	0.745 (0.035)
TimesNet	0.432 (0.000)	0.332 (0.055)	0.344 (0.062)	0.823 (0.000)	0.687 (0.087)	0.710 (0.043)
CSDI	0.459 (0.000)	0.291 (0.056)	0.438 (0.016)	0.853 (0.000)	0.553 (0.108)	0.762 (0.019)
GP-VAE	0.456 (0.000)	0.396 (0.036)	0.334 (0.072)	0.816 (0.000)	0.745 (0.056)	0.701 (0.041)
US-GAN	0.492 (0.000)	0.333 (0.044)	0.223 (0.046)	0.839 (0.000)	0.708 (0.019)	0.610 (0.043)

The table provides some interesting results. First, all models generally underperform despite some showing relatively high ROC_AUC values. This is demonstrated by the PR_AUC results and is a direct consequence of the imbalanced nature of the Physionet

2012 dataset. In this dataset, mortality (positive class) is relatively rare, with the actual imbalance ratio being approximately 86% negative (survival) to 14% positive (mortality) cases. Second, XGBoost consistently outperforms the end-to-end approaches (RNN and Transformer) across all models. XGBoost's superior performance can be attributed to its strength in handling complex feature interactions and the effectiveness of separating imputation from classification, allowing each model to specialise in its task. While XGBoost ignores explicit temporal relationships, this limitation is mitigated by the temporal information already encoded in the imputed values by the imputation models, allowing the model to focus on mapping feature interactions for better predictive performance. These findings can indicate that a modular approach to imputation and downstream task implementation maybe favourable, allowing for specialised components to optimise a single task, especially given the notably high variance in the end-to-end approaches (especially for RNN), suggesting less stable performance. This observation is especially true for GRUD, which has no explicit design for capturing cross-sectional correlation. Its XGBoost performance indicates that the classifier complements the imputer's capabilities in this regard.

4.6 Key Findings

4.6.1 Design Choices Matter

Neural architectures exhibit inherent inductive preferences that cascade into sophisticated imputation strategies. However, design decisions can complement architectural bias to accommodate data characteristics. Our taxonomy and experimental results show that model sophistication is not directly linked to parameter count, but to the sophistication of bias integration to accommodate the complex features of EHR data. For example, BRITS' thoughtful design transcends RNNs' basic sequential modelling limitations by explicitly incorporating cross-sectional relationships through an additional fully connected layer,

demonstrating how carefully engineered architectural modifications can outperform some of the more elaborate models. Similarly, SAITS integrates global attention with local refinement mechanisms, generating higher-order representations that overcome architectural limitations.

4.6.2 Alignment with EHR Characteristics Matters

The most compelling models emerge when architectural and framework biases are deliberately aligned with clinical data characteristics, and clear deviation leads to suboptimal performance. This is exemplified by MRNN, where its multi-resolution temporal dynamic oversimplifies the cross-sectional relationships between different clinical measurements, and GRUD's exponential decay assumption, though innovative for handling irregular sampling, does not adequately account for the non-linear interdependencies between variables that emerge in complex medical trajectories. Similarly, VAEs are challenged by the misalignment between their distributional assumption and the heterogeneous nature of medical time series. VAEs' limitations may be overcome by MDNs' mixed distributions, but this remains untested by us as `PyPOTS` does not yet include an MDN model.

4.6.3 Benchmarking Design Matters

Our experimental evaluation demonstrates that rigorous benchmarking requires careful consideration of both masking strategies and implementation choices. The significant performance variations observed under different masking patterns (point, temporal, and block) reveal that simplistic random masking may lead to over-optimistic performance estimates that don't reflect the real missingness patterns within the data. Furthermore, implementation decisions such as the timing of masking application (pre-mask vs. mini-batch) and masking strategy (overlay vs. augmentation) significantly impact model evaluation, with performance variations of up to 20% in MAE. These findings emphasise that standardised,

comprehensive benchmarking frameworks such as `PyPOTS` are essential for meaningful model comparison and assessment of practical utility.

4.6.4 Choice of Downstream Model Matters

Our results surprisingly show that a classifier not specialised in time-series (XGBoost) outperformed end-to-end models. The results indicate that separating imputation from prediction tasks leads to better performance, with XGBoost outperforming end-to-end approaches when applied to imputed data. This aligns with observations in the literature that models combining imputation with downstream tasks (e.g., GRUU [78], V-RIN [115], and BRITS [20]) may suffer from reduced performance. This performance gap likely stems from the differing nature of these tasks: imputation requires capturing fine-grained temporal and feature relationships to reconstruct missing values accurately, whilst classification often relies on identifying broader, task-specific patterns in the data. By separating these tasks, each model can specialise in its respective objective, leading to better overall performance.

4.7 Chapter Summary

This chapter presented an experimental evaluation of representative neural time-series imputers, comparing models across model complexity, missingness mechanisms, and implementation choices. The results reveal important key insights about NN imputation approaches for healthcare time-series: (1) architectural sophistication does not necessarily translate to proportional performance gains, as evidenced by TimesNet's moderate performance despite its large parameter count; (2) models incorporating both temporal and cross-sectional components (SAITS, CSDI) generally demonstrate better robustness to complex missingness patterns; (3) implementation choices significantly impact performance, with mini-batch masking generally outperforming pre-masking approaches; and (4)

separating imputation from downstream tasks not only leads to more modular frameworks, but can lead to better overall performance compared to end-to-end approaches.

The results also show that while newer architectures like SAITS and CSDI achieve marginally better performance in some scenarios, BRITS emerges as a particularly compelling foundation for future development. Despite its relatively modest size (9.1M parameters), BRITS demonstrates remarkable consistency and robustness across different experimental conditions. It maintains strong performance (MAE 0.297) in standard settings, shows minimal degradation under complex missingness patterns (MAE range 0.297-0.32), and exhibits stable performance across different masking implementations (MAE range 0.254-0.263). While its training time is longer than some newer models, its architectural simplicity, consistent performance, and ability to handle both temporal and feature relationships make it an ideal backbone for further innovation. Notably, its bidirectional structure provides natural handling of temporal dependencies while maintaining interpretability, a crucial factor for healthcare applications. The efficiency-robustness balance struck by BRITS, combined with its architectural flexibility for modification, provides a solid foundation for developing more sophisticated imputation approaches in subsequent chapters.

The findings of this chapter inform the two models I present in Chapters 5 and 6. In both models, I extend the BRITS framework to further increase the model's capability to deal with EHR characteristics. Chapter 5 presents CSAI, which incorporates domain knowledge to enhance the model's capability to deal with the irregular sampling pervasive in the datasets and incorporates attention to initialise the model's parameters, enabling the capturing of global (and long-term temporal) information in addition to the dynamics already captured by BRITS. In Chapter 6, I take another approach. Here, I present DEARI, which investigates whether a well-designed deeper model can improve performance, particularly in datasets with higher complexity.

Chapter 5

Knowledge Enhanced Conditional Imputation for Healthcare Time-series

Building upon the key findings of Chapter 3.1.1, which revealed both the strengths and limitations of current architectures, this chapter presents the first of two models designed to further align neural imputers with EHR data characteristics. CSAI (Conditional Self-Attention Imputation) builds on the BRITS backbone through three key contributions specifically adapted to address EHR data challenges:

1) **Transformer-Enhanced Initial State Learning:** CSAI enhances the RNN-based BRITS with Transfomer features to capture both long- and short-range temporal dependencies prevalent in EHRs. This is done through an attention-based hidden state initialisation technique to capture

2) **Adaptive Clinical Decay Mechanism:** CSAI incorporates a domain-informed temporal decay mechanism to adjust the imputation process to clinical data recording patterns.

3) **Distribution-Aware Masking Framework:** CSAI is trained through a novel non-uniform masking strategy that models non-random missingness by calibrating weights according to both temporal and cross-sectional data characteristics.

Before delving into the design of CSAI, I first present a detailed discussion on the design of my chosen backbone model, BRITS, in Section 5.1.

5.1 Overview of the BRITS Model

The BRITS model, which serves as the foundation for our work, processes incomplete time series through a bidirectional recurrent architecture that exploits both temporal and feature-wise correlations. As illustrated in Figure 5.1, BRITS processes an incomplete time series by combining a fully connected regression module for cross-sectional correlations with a recurrent module for temporal dependencies.

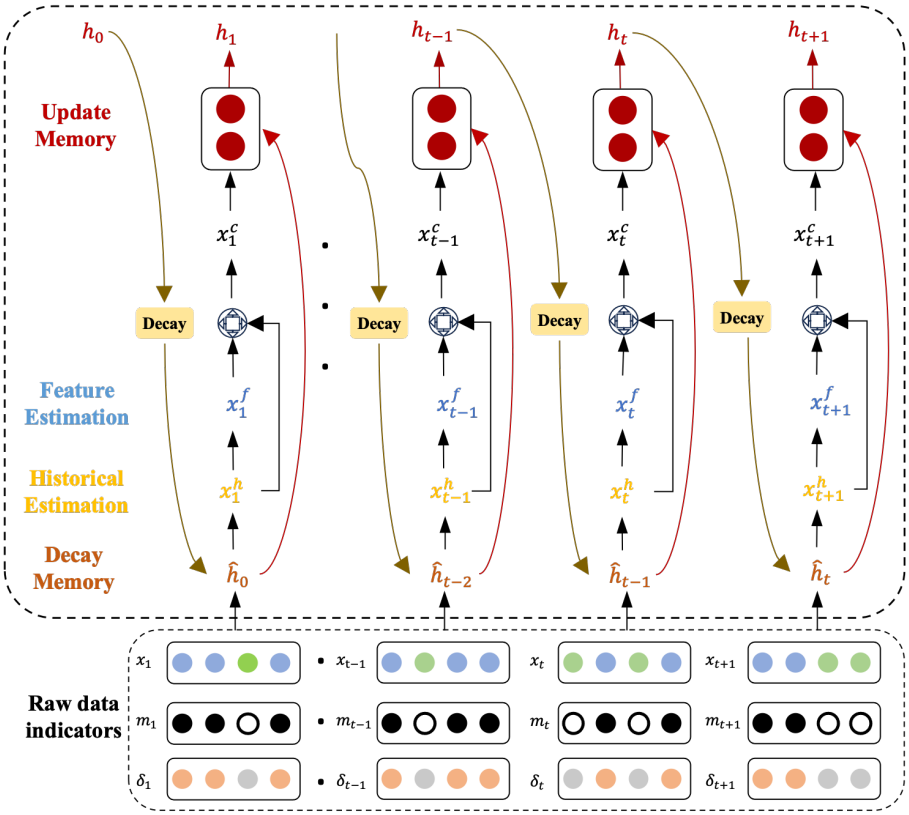


Fig. 5.1 The BRITS backbone process

The workflow of BRITS consists of four main steps:

1. **Temporal Decay**: For each timestamp, BRITS computes a decay factor based on the time elapsed since the last observation, using this to adjust the influence of historical information.

2. **Historical Estimation**: Using the decayed hidden state, the model generates an initial estimate for missing values based on temporal patterns.

3. **Feature-wise Refinement**: These estimates are refined through a fully connected layer that captures correlations between different features.

4. **Bidirectional Integration**: The process runs in both forward and backward directions, with the final imputation combining predictions from both passes.

The complete mathematical formulation of BRITS can be expressed as follows:

$$\text{Initialization} \quad h_0 = 0 \tag{5.1}$$

$$\text{Temporal decay} \quad \gamma_{th} = \exp\left(-\max(0, W_\gamma \delta_t + b_\gamma)\right) \tag{5.2}$$

$$\text{Decayed state} \quad \hat{h}_{t-1} = h_{t-1} \odot \gamma_t \tag{5.3}$$

$$\text{Historical estimation} \quad \hat{x}_t = W_x \hat{h}_{t-1} + b_x \tag{5.4}$$

$$\text{Historical complement} \quad x_t^h = m_t \odot x_t + (1 - m_t) \odot \hat{x}_t \tag{5.5}$$

$$\text{Feature-wise estimation} \quad x_t^f = W_z x_t^h + b_z \tag{5.6}$$

$$\text{Feature decay} \quad \gamma_{tf} = \exp\left(-\max(0, W_{\gamma f} \delta_t + b_{\gamma f})\right) \tag{5.7}$$

$$\text{Combination weight} \quad \hat{\beta}_t = \sigma(W_\beta [\gamma_{tf} \circ m_t] + b_\beta) \tag{5.8}$$

$$\text{Combined estimation} \quad \hat{x}_t^c = \beta_t \odot x_t^f + (1 - \beta_t) \odot x_t^h \tag{5.9}$$

$$\text{Final imputation} \quad x_t^c = m_t \odot x_t + (1 - m_t) \odot \hat{x}_t^c \tag{5.10}$$

$$\text{Hidden state update} \quad h_t = \sigma(W_t \hat{h}_{t-1} + U_h [x_t^c \circ m_t] + b_h) \tag{5.11}$$

where $\odot$ denotes element-wise multiplication and $\circ$ denotes concatenation. The entire process runs in both forward and backward directions, with the final imputation combining predictions from both passes.

Temporal Modelling

BRITS introduces the concept of temporal decay to model how the influence of past observations diminishes over time. For each time step, a decay factor $\gamma_t \in (0,1]$ is computed as:

$$\gamma_{th} = \exp\left(-\max(0, W_\gamma \delta_t + b_\gamma)\right)$$

where W_γ and b_γ are learnable parameters. This decay factor is used to transform the previous hidden state h_{t-1} into a decayed state $\hat{h}_{t-1}$:

$$\hat{h}_{t-1} = h_{t-1} \odot \gamma_t$$

Feature-wise Dependencies

The model captures correlations between features through a fully connected layer. For an observation x_t with missing values, BRITS first generates a historical estimation $\hat{x}_t$ using the decayed hidden state:

$$\hat{x}_t = W_x \hat{h}_{t-1} + b_x$$

This estimation is combined with the original observation using the mask vector to create a complement vector x_t^h:

$$x_t^h = m_t \odot x_t + (1 - m_t) \odot \hat{x}_t$$

Bidirectional Architecture

BRITS processes the time series in both forward and backward directions to leverage future information for imputation. The final imputed values are obtained by combining the

predictions from both directions:

$$x_t^c = m_t \odot x_t + (1 - m_t) \odot \hat{x}_t^c$$

where $\hat{x}_t^c$ is computed using a learnable combination of feature-wise and temporal estimations.

5.2 The CSAI Model

This section describes the modifications proposed to the BRITS architecture to incorporate a) a domain-informed temporal decay functionality, b) the ability to capture long-range correlations via a transformer-based conditional hidden state initialisation, and c) a novel non-uniform masking strategy to explicitly model non-random EHR missingness. The section concludes with the resulting learning framework of CSAI.

5.2.1 EHR-Tailored BRITS Adaptations

A. Domain-informed Temporal Decay: The BRITS decay function Eq.(5.2) is strictly dependent on temporal proximity, dynamically adjusting the contribution of a past observation to a missing value based on the length of the time gap between the two. Although this decay mechanism effectively captures the intuition that more recent observations carry greater diagnostic value, it overlooks domain-specific discrepancies, where different features follow distinct recording frequencies due to clinical practices. This can be illustrated in the example below.

Example 1 *Let features f_1 and f_2 represent heart rate (HR) and systolic blood pressure (SBP), respectively. As a vital sign indicative of a patient's state, HR is typically monitored at more frequent intervals compared to SBP. BRITS is presented with an observation x_t where both HR and SBP values are missing. The time gap vector δ_t shows $\delta_t^{f_1} = 2$*

(indicating that the last HR recording occurred 2 time units ago) and $\delta_t^{f_2} = 7$ *(indicating that the last SBP recording occurred 7 time units ago).*

According to the principle of temporal decay, BRITS would apply a stronger influence from the more recent HR observation compared to the older SBP observation, despite the fact that SBP is typically recorded less frequently and still carries significant diagnostic importance. Consequently, BRITS would incorrectly assign less weight to the last SBP observation compared to HR, overlooking the domain-specific recording patterns, where the last recording of SBP, despite a longer time gap, remains a critical input for imputation.

The proposed new decay mechanism prioritises recent observations while accounting for the natural variability in healthcare data collection. In addition to using the observed time gap δ_t^d, I modify the decay function to incorporate *the expected time gap* τ between two recordings of a given feature. The expected time gap τ_d for a feature is computed as the median of the time intervals between successive recordings of the feature d in the dataset. This adjustment allows the decay function to adapt to the recording patterns of different clinical features, ensuring that features like SBP, which are recorded less frequently, still carry appropriate weight during imputation.

The new decay factor γ_t^d for a feature d at time t is computed as:

$$\gamma_t^d = \exp(-\max(0, W_\gamma(\delta_t^d - \tau_d) + b_\gamma)) \tag{5.12}$$

Where:

- δ_t^d is the time gap since the last observation of feature d,
- τ_d is the median time gap for feature d,
- W_γ and b_γ are learnable parameters.

This formulation ensures that the decay factor peaks when the time gap δ_t^d closely matches the expected time gap τ_d, and declines as the difference between δ_t^d and τ_d

increases. Thus, observations that fall within their expected time gap contribute more strongly to the imputation process, while those that lie far outside the expected gap are down-weighted as illustrated in the example below.

Example 2 *For the same features f_1 and f_2, examining the dataset reveals that the median time gaps for the two features are $\tau_1 = 2$ for HR and $\tau_2 = 10$ for SBP. These medians reflect the fact that HR is routinely monitored more frequently than SBP in clinical practice. By leveraging these median time gaps, the model can account for the different recording frequencies. Here, since the last observed values for both HR and SBP fall within their respective median time gaps, the model assigns comparable importance to both past recordings when imputing the missing values. This approach ensures that the less frequently measured SBP, with its median gap of 10, is not unfairly penalised, preserving the clinical relevance of both features.*

B. Attention-based Hidden State Initialisation: The hidden state of the recurrent component of BRITS are not generated through the raw input. Instead, they receive incomplete data with missingness indicators for imputation, i.e. $\hat{h}$ replaces h in Eq.(5.3). However in BRITS, the initial hidden state is initialised to zero as in Eq. (5.1), which causes the model to rely solely on its internal parameters to estimate initial missing values in Eq. (5.4), ignoring crucial information from prior observations. This can be particularly problematic in medical data, where early data points are often crucial to understanding patient trends, and failure to incorporate them can lead to inaccurate imputations and the loss of important clinical insights as illustrated by the example below.

Example 3 *Consider a patient whose heart rate (HR) and systolic blood pressure (SBP) are being monitored. Over the last few recorded measurements, the patient's HR has been steadily increasing, indicating a deteriorating condition. However, there is a monitoring gap leading to missing initial values. With the hidden state initialised to zero, BRITS would not only fail to capture the upward trend in HR by disregarding the critical information*

from later measurements, but could also propagate erroneous assumptions about stability throughout the time series. As the hidden state learns from the imputed values, it would continuously stack these errors, potentially amplifying the misrepresentation of the patient's condition as stable when in fact urgent intervention is required.

To solve this issue, CSAI makes use of the last observed data point and a decay attention mechanism to generate a conditional distribution for the initial hidden state $q(h_{\text{init}}|x_{\text{last\_obs}}, \gamma_t^d))$ within the model distribution $p_\theta(x_t)$. This strategy is designed to provide a more contextually rich starting point for the model, thus enhancing the effectiveness of subsequent imputations. Instead of applying the decay factor directly to the previous hidden state as done in BRITS (Eq. (5.3) and Eq.(5.8)), I use the decay factor to modulate an attention mechanism to better capture long-range and feature-specific temporal dynamics, discussed below.

First, at each time step s_t, the last observation $x_{\text{last\_obs}} \in \mathbb{R}^{T \times D_{\text{feature}}}$ and the decay factor γ_t^d are projected and encoded to capture their temporal positioning within the sequence:

$$x'_{\text{last\_obs}} = \text{PosEncoder}(\text{InputProj}(x_{\text{last\_obs}})) \tag{5.13}$$

$$\gamma'_t = \text{PosEncoder}(\text{InputProj}(\gamma_t^d)) \tag{5.14}$$

Then the transformed input representations are concatenated and fed into a Transformer encoder, which is capable of capturing both long-range dependencies and feature-specific interactions:

$$C_{\text{in}} = \text{Concat}(x'_{\text{last\_obs}}, \gamma'_t) \tag{5.15}$$

$$C_{\text{out}} = \text{LN}(\text{FFN}(\text{LN}(\text{MSA}(C_{\text{in}})))) \tag{5.16}$$

The output of the Transformer is then passed through a series of 1D convolutional layers to adjust the dimensions and initialise the hidden state:

$$H_1 = \text{Conv1D}_1(C_{\text{out}} W_1 + b_1) \tag{5.17}$$

$$h_{\text{init}} = \text{Conv1D}_2(H_1 W_2 + b_2) \tag{5.18}$$

where Eq.(5.17) transforms C_{out} from $\mathbb{R}^{2L \times d_{\text{model}}}$ to $\mathbb{R}^{2L \times d_{\text{hidden}}}$ and produces H_1, Eq.(5.18) further scales H_1 to generate the initialised Hidden State h_{init}. This Transformer-based hidden-state initialisation allows the model to better capture the temporal dynamics and variability in feature-recording patterns, providing a more robust foundation for subsequent imputation steps. The CSAI architecture is shown in Figure 5.2. CSAI begins with an input embedding layer, followed by positional embedding to capture time dependencies. These embeddings are processed through multi-head attention, normalisation, and feed-forward layers. The output is used to initialise hidden states for subsequent recurrent layers, accounting for both temporal dynamics and domain-specific variability in recording patterns.

5.2.2 Non-Uniform Masking Strategy

The non-uniform masking strategy presented here fundamentally diverges from traditional approaches by leveraging the missingness distribution within the dataset. The strategy is predicated on the principle outlined in the introduction, whereby the probability of missingness in healthcare data is not uniformly distributed across features. Instead, it varies according to healthcare practices and patient conditions. By incorporating this variability into the masking process, I aim to create a more realistic and representative model of the missingness patterns within the dataset.

The masking algorithm generates masking probabilities using two factors:

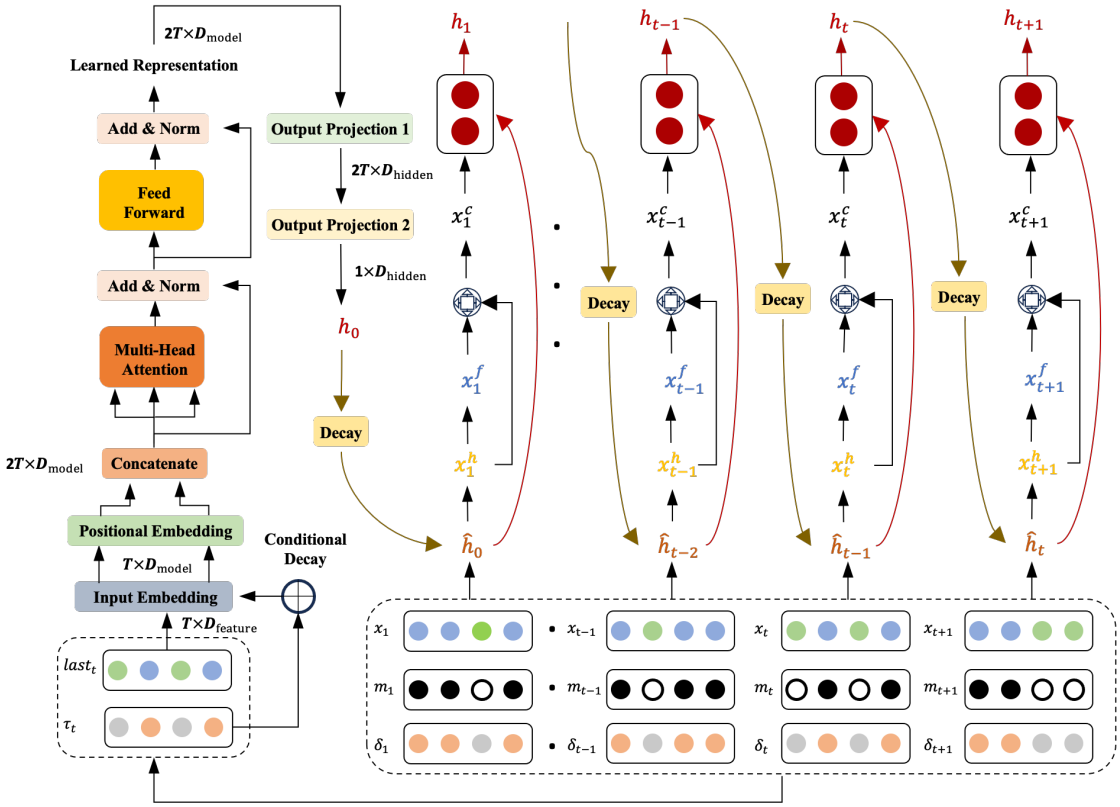


Fig. 5.2 The CSAI architecture consists of two main components: (1) A Transformer-based conditional initialisation module (left) that processes the last observed values (last_t) and expected time gaps (τ_t) through input embedding, positional encoding, and multi-head attention layers to generate contextually-rich initial hidden states, and (2) A bidirectional RNN component (right) that incorporates the domain-informed decay mechanism between time steps. The input at each time step consists of feature values (x_t), missingness indicators (m_t), and time gaps (δ_t). The decay mechanism (yellow boxes) modulates the influence of previous hidden states (h_{t-1}) and observations (x^f_{t-1}) based on both temporal proximity and expected recording frequencies of different features.

Missingness Distribution $P_{\text{dist}}(d)$: This reflects the likelihood of masking a feature based on its missingness patterns across similar or neighbouring observations.

Adjustment Factor $R_{\text{factor}}(d)$: A dynamic value that adjusts the masking probability based on how often the feature d is observed. Frequently observed features are masked less to avoid overfitting, while sparsely observed features are masked more to ensure sufficient model learning from limited data points.

For a given feature d, the non-uniform masking probability $P_{\text{nu}}(d)$ is determined as follows:

$$R_{\text{factor}}(d|U,I) = F(d,U,I) \tag{5.19}$$

$$P_{\text{nu}}(d) = R_{\text{factor}}(d|U,I) \times P_{\text{dist}}(d) \tag{5.20}$$

Where:

- U and I are the pre-defined parameters. U is the masking rate, i.e. the percentage of the ground truths masked during training. I is a weighting parameter.
- $R_{\text{factor}}(d|U,I)$ is the adjustment factor for feature d, conditioned by U and I.
- $P_{\text{dist}}(d)$ is d's missingness probability distribution.

The overall masking proportion is then adjusted to ensure consistency with the uniform masking rate U, while retaining the non-uniform characteristics of the individual features. The pseudocode for this algorithm is given below, producing a masked matrix X_M.

Algorithm 1 ***non-uniform-mask***

Input: Dataset X with D features, masking rate U, weighting parameter I
Output: Masked Dataset X_M
for each feature d in D **do**
  $P_{\text{dist}}(d) \leftarrow \text{compute}(x^d)$
  $R_{\text{factor}}(d) \leftarrow f(U,I)$
  $P_{\text{nu}}(d) \leftarrow R_{\text{factor}}(d) \times P_{\text{dist}}(d)$
end for
$U \leftarrow f(P_{\text{nu}},X)$
$X_M \leftarrow P_{\text{nu}} \times X$

5.2.3 Learning

CSAI is trained in an unsupervised manner by stochastically masking non-missing values using the non-uniform masking strategy and learning to impute them. At evaluation time, the trained RNN can impute both real and simulated missing values. CSAI's training

iterates over mini-batches of the input data, where each mini-batch comprises a set of samples with T time steps. The devised imputation loss function $\mathscr{L}_{imp}$ maintains the objective of minimising the reconstruction error ℓ_{obs} between the imputed and ground truth vectors x and $\hat{x}$, while maintaining consistency loss ℓ_{con} between the forward and backward imputed estimates $\overrightarrow{\hat{x}}$ and $\overleftarrow{\hat{x}}$ of CSAI's bi-directional RNN as follows:

$$\mathscr{L}_{imp} = \frac{1}{|\mathscr{B}|} \sum_{i \in \mathscr{B}} \sum_{t \in T_i} \lambda \Big[M_{it} \odot \ell_{\text{obs}}(x_{it}, \hat{x}_{it}) + (1 - M_{it}) \odot \ell_{\text{con}}(\overrightarrow{\hat{x}_{it}}, \overleftarrow{\hat{x}_{it}}) \Big] \tag{5.21}$$

Where $\mathscr{B}$ is the mini-batch size, T is the set of time steps over which the loss function is applied, M is the masking matrix, and λ is a hyperparameter that balances the two loss terms. Both ℓ_{obs} and ℓ_{cons} use the mean absolute error (MAE) to measure the difference between the components of each loss function.

Since CSAI *can* be used as an end-to-end pipeline to perform imputation and prediction, I further define prediction loss using binary cross-entropy as follows:

$$\mathscr{L}_{\text{pred}} = -\sum_{n=1}^{N} y^{(n)} \log(\hat{y}^{(n)}) \tag{5.22}$$

The resulting combined loss is computed as:

$$\mathscr{L}_{\text{combined}} = \alpha \mathscr{L}_{\text{imp}} + \beta \mathscr{L}_{\text{pred}} \tag{5.23}$$

Where β With α and β are preset parameters representing the weight of the respective loss component on $\mathscr{L}_{\text{combined}}$. Using CSAI solely as imputer is done by setting β to zero. With this loss function, CSAI uses backpropagation in an end-to-end manner to optimise model parameters during training.

5.3 Experimental Evaluation

I conducted several experiments to evaluate the proposed methodologies and their impact on CSAI's performance on widely-used healthcare benchmark datasets described in Section 2.8.

5.3.1 Experimental Design

I conducted two sets of experiments to test the methodologies proposed in this work. The first comprises **benchmark comparison experiments** comparing CSAI's imputation and subsequent classification performance against the current state of the art on two scales:

Experiment I: compares CSAI with the best-performing RNN models using the three benchmark datasets and three masking rations (5%, 10% and 20%). Here, CSAI is compared with the backbone bidirectional RNN model BRITS [20], the Gated Recurrent Unit with Decay (GRU-D) [22], the Variational Recurrent Imputation Network (V-RIN) [115], and the Multi-directional Recurrent Neural Network (M-RNN) model [183]. In this experiment, two tasks were undertaken to ensure a fair comparison: 1) Because the original BRITS implementation uses LSTM cells while all other baseline models use GRU cells, I implemented a GRU-based BRITS model to ensure a balanced evaluation across similar architectures, including its output along with the original BRITS (BRITS-GRU in the results), 2) I employed the new non-uniform masking algorithm as a pre-processing step for all models to avoid any bias arising from the masking step.

Experiment II: is a large-scale experimental evaluation with different flavours of neural imputation models using the Physionet dataset with a 10% masking ratio. This experiment was conducted using the PyPOTS library[1] [47], an open-source Python toolkit for benchmarking and analysing incomplete multivariate time series. PyPOTS provides access to 29 imputation algorithms, including CSAI and a number of benchmark datasets, including Physionet. In this experiment, CSAI is compared against a broad range of

[1] https://pypots.com/

neural network imputation models, such as Transformers, CNNs, GNNs, and diffusion models, as well as RNNs and standard statistical imputation methods. As non-uniform masking is not yet integrated into PyPOTS, I rely on the available masking setup, aiming to evaluate the contribution of CSAI's temporal decay and attention mechanism on the resulting performance.

The reader should note that although a comparison with generative models through E^2GAN [99] would have provided valuable insight, particularly given its lack of its quantitative comparison with BRITS, the publicly available implementation of E^2GAN is incompatible with my current setup and is also not part of PyPOTS.

The second set of experiments is **ablation studies** performed using the public Physionet dataset. First, I establish the effect of non-uniform masking across different data partitions, namely the training, validation, and test sets. The aim is to establish whether this approach actually improves a model's ability to generalise and handle missing data in a more realistic and robust manner. Finally, I evaluate the effect of the weighting parameter, which controls the degree of emphasis placed on different features in the non-uniform masking strategy. This parameter allows for fine-tuning of the feature representation during model training, directly influencing how the model handles missing data, and I evaluate the effect on CSAI's imputation and classification performance under different weighting parameter values.

5.3.2 Experimental Setup

Environment: I trained the models on an HPC node equipped with an NVIDIA A100 40GB GPU, running Ubuntu 20.04.6 LTS (Focal Fossa). The experiments were carried out using Python 3.8.16. For the first set of experiments comparing CSAI with other RNN models, I directly used the source code for BRITS, GRU-D, and MRNN, obtained from their respective repositories, along with CSAI. For the broader set of experiments comparing CSAI with all neural imputation models, I utilised the PyPOTS implementation

of all models, running all experiments through the PyPOTS package installed on the college's HPC node. Full package details can be found in the CSAI repository. To help reproduce the Python environment easily, I freeze the development environment with Anaconda and save it into file for reference, which is available in the project's `GitHub` repository. To ensure reproducibility of the results, I release all data preprocessing scripts, model implementations, and hyperparameter search configurations.

Dataset Pre-processing and Missingness Patterns Simulation: Each dataset was normalised to zero mean and unit variance. For each of the four datasets, I created three versions by further masking 5%, 10% and 20% cells in addition to the missingness already present in the original datasets. These masked cells have known ground truths and will form the basis of the comparison of imputation performance. To more effectively simulate EHR missingness scenarios, the first set of experiments use the proposed non-uniform masking as a pre-processing step to enable capturing patterns representative of common missingness data mechanisms in EHRs to the masked cells.

Hyper-parameter Optimisation: The performance of deep-learning models highly depends on the settings of hyper-parameters. To make fair comparisons across imputation methods and draw impartial conclusions, hyper-parameter optimisation is applied to all imputation algorithms using `PyPOTS`.

Training: Except the PhysioNet dataset, which already contains separated time series samples, all other datasets are split into training, validation, and test sets. Randomly, I selected 10% of each dataset for validation and another 10% for testing, training the models on the remaining data. I used the Adam optimiser and set the number of RNN hidden units to 108 for all models. The batch size is 64 for PhysioNet and 128 for the other datasets. A 5-fold cross-validation method was implemented to evaluate the models.

Downstream Task Design: To evaluate the impact of imputation on downstream analysis, I further perform a classification task to predict in-hospital mortality. In the Physionet dataset, each sample has a label indicating whether the patient is deceased in the ICU. For all other datasets, I extracted the results for each patient according to the code provided

in the benchmark. To demonstrate the flexibility of the model and its ability to perform using different classification architectures, I used different methodologies to implement the classifiers in the experiments. In **Experiment I**, classification was performed in an end-to-end manner by adding a classification layer to each architecture, utilising the hidden states from the imputation network to feed into the classification layer. In **Experiment II**, the imputation models were used to generate complete datasets, which were subsequently fed into three different classifiers for comparison: XGBoost, RNN and Transformer.

Table 5.1 Imputation performance using 5%, 10%, and 20% masking ratios on three datasets. The model with the lowest MAE in each setup is highlighted in bold.

MIMIC_59 (MAE)			
Model	**5%**	**10%**	**20%**
V-RIN	0.15457 ± 0.007	0.13818 ± 0.017	0.33697 ± 0.010
BRITS	0.15195 ± 0.018	0.14023 ± 0.009	0.34039 ± 0.019
BRITS_GRU	0.14793 ± 0.016	0.14193 ± 0.015	0.34169 ± 0.017
GRUD	0.30447 ± 0.012	0.28704 ± 0.014	0.48670 ± 0.017
MRNN	0.30573 ± 0.013	0.28342 ± 0.012	0.47198 ± 0.015
CSAI	**0.13119 ± 0.009**	**0.11291 ± 0.008**	**0.30976 ± 0.014**
eICU (MAE)			
V-RIN	0.24161 ± 0.015	0.24254 ± 0.013	0.25214 ± 0.019
BRITS	0.16699 ± 0.014	0.17053 ± 0.020	0.17681 ± 0.009
BRITS_GRU	0.17232 ± 0.010	0.17124 ± 0.019	0.17691 ± 0.013
GRUD	0.22274 ± 0.018	0.22560 ± 0.010	0.23098 ± 0.020
MRNN	0.47036 ± 0.015	0.47998 ± 0.017	0.50065 ± 0.020
CSAI	**0.15967 ± 0.017**	**0.16149 ± 0.011**	**0.16637 ± 0.015**
PhysioNet (MAE)			
V-RIN	0.26163 ± 0.015	0.27372 ± 0.010	0.29997 ± 0.018
BRITS	0.25634 ± 0.013	0.26762 ± 0.017	0.28722 ± 0.014
BRITS_GRU	0.25129 ± 0.012	0.26216 ± 0.011	0.28288 ± 0.018
GRUD	0.49406 ± 0.015	0.49779 ± 0.020	0.50952 ± 0.018
MRNN	0.54671 ± 0.013	0.55647 ± 0.014	0.57230 ± 0.017
CSAI	**0.24602 ± 0.014**	**0.25747 ± 0.017**	**0.27476 ± 0.019**

5.3.3 Experimental Results

Experiment I: Comparison with State-of-the-Art RNN Models on Healthcare Benchmarks: The evaluation of CSAI against four state-of-the-art RNN imputation models on

all three healthcare benchmarks across three masking ratios is shown in Tables 5.1 and 5.2. Table 5.1 shows the imputation performance, where CSAI consistently outperforms other models in all data sets and masking ratios (5%, 10%, and 20%). The best performance for all models is observed on the eICU and MIMIC_59 datasets, which, despite high missingness rates, offers a large number of samples for training. The Physionet dataset, while having the lowest baseline missingness rate, provides significantly fewer samples compared to the other datasets, limiting model performance across the board. The table also shows that the performance gap between CSAI and other models widens as the masking ratio increases, leading to conditions of high data loss. This is particularly pronounced in the highly-dimensional dataset MIMIC_59 at the 20% missingness ratio. For V-RIN, BRITS, and BRITS_GRU, the MAE is relatively stable across masking ratios but remains consistently higher than CSAI. GRUD and MRNN show notably higher MAE values, especially at higher masking ratios.

Table 5.2 further demonstrates CSAI's strong classification performance, achieving the highest AUC scores across various masking ratios in the eICU, PhysioNet, and MIMIC_59 datasets. The difference between CSAI's AUCs and those of other models is highest under the Physionet dataset, which has the smallest number of samples. Across all three datasets, CSAI's AUC only slightly decreases as the masking ratio increases to 20%. Although a similar stability is observed in BRITS and V-RIN, the AUCs are consistently lower than CSAI's.

Experiment II: Large-scale Comparison Using PyPOTs Table 5.3 shows classification results for 24 neural imputation models and four statistical imputers. This experiment uses the imputed data generated by each model as input to a classifier. I show the results for three classifiers, and XGBoost, RNN and a Transformer classifier. The reader should not that non-uniform masking was not implemented into CSAI here, since it is yet to be integrated into a callable capability in the PyPOTs library and it is therefore not possible to use it with any other imputation models apart from CSAI.

Table 5.2 Classification performance using 5%, 10%, and 20% masking ratios on three datasets. The model with the highest AUC in each setup is highlighted in bold.

MIMIC_59 (AUC)			
Model	**5%**	**10%**	**20%**
V-RIN	0.83280 ± 0.010	0.83312 ± 0.012	0.82733 ± 0.014
BRITS	0.82821 ± 0.010	0.82779 ± 0.012	0.82414 ± 0.013
BRITS_GRU	0.83188 ± 0.010	0.83074 ± 0.012	0.82688 ± 0.013
GRUD	0.82768 ± 0.012	0.82640 ± 0.014	0.82304 ± 0.012
MRNN	0.82106 ± 0.012	0.81708 ± 0.011	0.81282 ± 0.013
CSAI	$\mathbf{0.83524 \pm 0.014}$	$\mathbf{0.83367 \pm 0.013}$	$\mathbf{0.83109 \pm 0.012}$
eICU (AUC)			
V-RIN	0.88766 ± 0.012	0.88422 ± 0.013	0.88461 ± 0.015
BRITS	0.88669 ± 0.011	0.88521 ± 0.013	0.88569 ± 0.012
BRITS_GRU	0.88944 ± 0.012	0.88864 ± 0.015	0.88605 ± 0.011
GRUD	0.86495 ± 0.013	0.86456 ± 0.014	0.85801 ± 0.011
MRNN	0.87786 ± 0.011	0.87626 ± 0.014	0.87342 ± 0.015
CSAI	$\mathbf{0.88952 \pm 0.012}$	$\mathbf{0.88977 \pm 0.011}$	$\mathbf{0.88795 \pm 0.015}$
PhysioNet (AUC)			
V-RIN	0.83433 ± 0.011	0.82916 ± 0.010	0.82548 ± 0.015
BRITS	0.82214 ± 0.014	0.81175 ± 0.012	0.82181 ± 0.015
BRITS_GRU	0.80677 ± 0.014	0.81926 ± 0.013	0.80651 ± 0.015
GRUD	0.78997 ± 0.011	0.77652 ± 0.013	0.76989 ± 0.015
MRNN	0.80121 ± 0.014	0.79945 ± 0.013	0.79400 ± 0.015
CSAI	$\mathbf{0.86465 \pm 0.014}$	$\mathbf{0.85919 \pm 0.013}$	$\mathbf{0.83722 \pm 0.015}$

Overall, the XGBoost classifier achieved the best performance, with a mean AUC of 0.823 compared to 0.660 for the RNN classifier and 0.667 for the Transformer. With XGBoost, using CSAI as an imputer produced the best classification performance. Since XGBoost is a non-temporal model, it has clearly benefitted from CSAI's ability to encode sequential dependencies (across features and time) as informative features that complement XGBoost's strengths in feature-based learning. For both the RNN and Transformer classifiers, diffusion models produced the highest AUCs, largely due to their ability to offer RNN and Transformer imputers the advantage of smoother interpolations that likely enhance feature consistency across time steps, which is crucial for sequential models. Despite this, CSAI ranked highly under these classifiers, producing competitive AUC performance as a lightweight, efficient model that avoids the computational burden of diffusion models.

Ablation Study Figure 5.3 demonstrates a clear progression in CSAI's performance as its components are incrementally added. Starting from the baseline, the model's MAE improves with each addition, highlighting the impact of the domain-informed temporal decay, attention-based initialisation mechanism, and non-uniform masking. The final configuration, with all components included, achieves the lowest MAE across all three datasets. These results indicate that each component plays a distinct role in enhancing imputation accuracy, with the full model yielding the best performance.

I further evaluate the effect of non-uniform masking on algorithm performance by using the masking strategy on different combinations of the training, validation, and test sets and examining the number of training epochs required to achieve the reported performance for each. For objective evaluation of the non-uniform masking strategy, I conducted all experiments using the BRITS and BRITS-GRU baseline models, with the results shown in Table 5.4. These results demonstrate that consistently applying non-uniform masking across all data partitions (training, validation, and testing) yields the best performance, suggesting that the masking strategy effectively adjusts the representation of different features to optimally leverage the data distribution, enhancing the model's ability to handle the inherent heterogeneity of the dataset.

Table 5.3 Classification performance on the Physionet dataset using 10% masking ratio. The imputation was performed using the PyPOTS implementation of the major Transformer, RNN, CNN, GNN, diffusion, and statistical models. For comparison, imputation using XGBoost's internal imputer resulted in ROC_AUC 0.771 ± 0.000

Model Type	Model	XGB Classifier	RNN Classifier	Transformer Classifier
Transformers	**iTransformer**	0.852 ± 0.000	0.692 ± 0.073	0.685 ± 0.032
	SAITS	0.851 ± 0.000	0.658 ± 0.076	0.668 ± 0.037
	ETSformer	0.820 ± 0.000	0.734 ± 0.030	0.678 ± 0.013
	PatchTST	0.848 ± 0.000	0.703 ± 0.064	0.698 ± 0.061
	Crossformer	0.836 ± 0.000	0.683 ± 0.051	0.644 ± 0.026
	Informer	0.836 ± 0.000	0.701 ± 0.021	0.653 ± 0.029
	Autoformer	0.768 ± 0.000	0.597 ± 0.009	0.625 ± 0.008
	Pyraformer	0.829 ± 0.000	0.739 ± 0.026	0.654 ± 0.047
	Transformer	0.833 ± 0.000	0.723 ± 0.036	0.626 ± 0.036
RNN	**BRITS**	0.841 ± 0.000	0.667 ± 0.066	0.646 ± 0.074
	MRNN	0.760 ± 0.000	0.611 ± 0.008	0.619 ± 0.015
	GRUD	0.840 ± 0.000	0.701 ± 0.075	0.704 ± 0.035
	CSAI	**0.860** ± 0.000	0.702 ± 0.044	0.704 ± 0.023
CNN	**TimesNet**	0.823 ± 0.000	0.687 ± 0.087	0.710 ± 0.043
	MICN	0.824 ± 0.000	0.700 ± 0.071	0.686 ± 0.050
	SCINet	0.818 ± 0.000	0.663 ± 0.055	0.720 ± 0.034
GNNs	**StemGNN**	0.809 ± 0.000	0.534 ± 0.124	0.701 ± 0.025
	FreTS	0.840 ± 0.000	0.693 ± 0.035	0.674 ± 0.018
	Koopa	0.835 ± 0.000	0.509 ± 0.113	0.628 ± 0.036
	DLinear	0.851 ± 0.000	0.689 ± 0.038	0.611 ± 0.053
	FiLM	0.825 ± 0.000	0.565 ± 0.095	0.706 ± 0.046
Diffus.	**CSDI**	0.853 ± 0.000	0.553 ± 0.108	**0.762** ± 0.019
	US-GAN	0.839 ± 0.000	0.708 ± 0.019	0.610 ± 0.043
	GP-VAE	0.816 ± 0.000	**0.745** ± 0.056	0.701 ± 0.041
Stats	**Mean**	0.763 ± 0.000	0.620 ± 0.017	0.598 ± 0.025
	Median	0.772 ± 0.010	0.619 ± 0.013	0.605 ± 0.025
	LOCF	0.795 ± 0.033	0.634 ± 0.074	0.644 ± 0.061
	Linear	0.807 ± 0.036	0.640 ± 0.076	0.669 ± 0.069

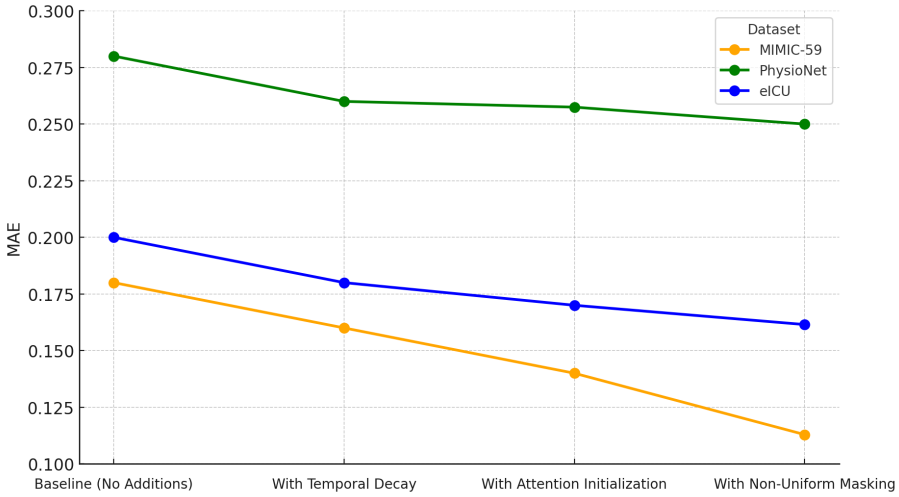


Fig. 5.3 Visualisation of the incremental increase in imputation performance (decrease in MAE) as the different components of CSAI are incrementally added. The figure shows CSAI's MAE using the three datasets and 10% masking ratio and the following variations: a) baseline model (i.e., a BRITS architecture), b) adding the temporal decay function, c) adding the attention-base initialisation mechanism, and finally d) using non-uniform masking.

Table 5.4 Performance comparison of applying the proposed non-uniform masking strategy on different combinations of the dataset: training, validation and testing. Under the 'All' configuration, non-uniform masking is applied during training, validation, and testing consistently reports the performance (bold font).

Model	**Masking**	**Imputation**		**Classification**		
		Epoch	**MAE**	**Epoch**	**MAE**	**AUC**
BRITS	All	182.2	**0.234929**	28.6	0.262997	**0.819142***
	Val_Test	187	0.235739	57	0.262268	0.816184
	Test_only	218.6	0.236001	61.6	0.265496	0.813821
	Train_only	215.8	0.266307	25.4	0.310624	0.815686
	None	218.6	0.26762	61.6	0.313257	0.81175
	Val_only	187	0.268386	57	0.309332	0.818605
BRITS-GRU	All	294.8	**0.231594**	21.8	0.26615	**0.821869***
	Val_Test	287.4	0.233243	21.6	0.267872	0.813855
	Test_only	286	0.23363	15.4	0.270678	0.811355
	Train_only	288.4	0.258989	17.6	0.313681	0.815697
	Val_only	287.4	0.262149	21.6	0.314871	0.81071
	None	286	0.262155	15.4	0.318493	0.819258

The non-uniform masking probability used by CSAI for each feature is determined by the parameters U and I, in addition to the feature's prior probability as shown in Eq.(5.19)-(5.20). I studied the effect of the weighting parameter I on the resulting imputation and classification performance. The findings are shown in Table 5.5 and are further illustrated in Fig.5.4. An optimal weighting parameter, shown to be around 5, results in the lowest imputation error, suggesting that a balanced representation of features is crucial for accuracy. However, for classification, increasing the weighting parameter leads to higher errors and a marginal decrease in the AUC, highlighting that excessive weighting may not uniformly improve performance across different machine learning tasks. These findings reveal the interplay between feature representation adjustments and task-specific model efficacy, underscoring the importance of calibration in the non-uniform masking approach for complex time series data.

Table 5.5 Impact of weighting parameter I on imputation and classification performance metrics. The table showcases the number of epochs required, imputation MAE and classification AUC across varying weighting factors.

I	**Imputation**		**Classification**		
	Epoch	**MAE**	**Epoch**	**MAE**	**AUC**
0	286	0.26215514	15.4	0.31849296	0.8192579
5	291.2	0.2310647	33.8	0.26586726	0.80975966
10	294.8	0.23159396	21.8	0.26615049	0.8118691
50	293	0.24170042	16.4	0.28670924	0.81583396
100	285.8	0.26885496	14.2	0.32437366	0.81381879
150	283.8	0.29943347	16.6	0.35346112	0.8144707
200	289.2	0.33235399	15.6	0.39360851	0.81173828

5.4 Chapter Summary and Significance

This chapter introduced CSAI, a novel approach for imputing missing data in multivariate medical time series. CSAI's novelty lies in its components specifically tailored to medical time-series, where data collection frequency and timing are highly variant, and long- and

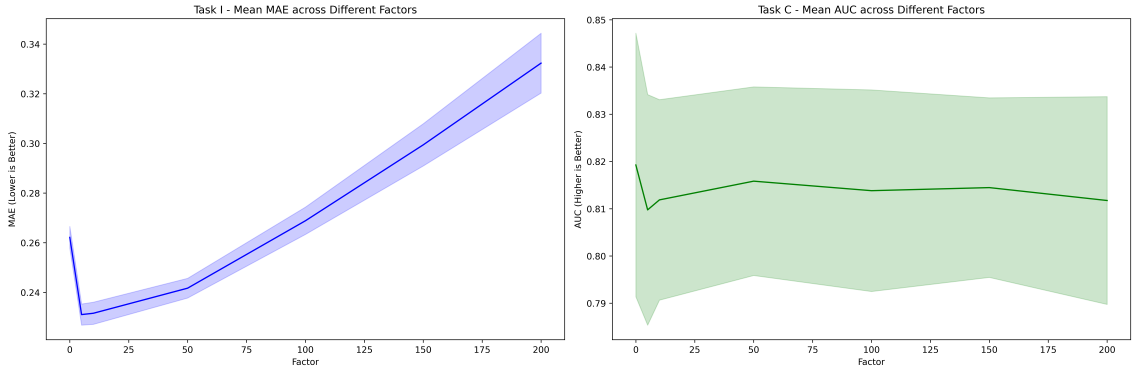


Fig. 5.4 Impact of Adjustment Factor on Model Performance in the Physionet Dataset

short-term correlations are pervasive. Through conditional knowledge embedding and attention mechanisms, CSAI demonstrates significant potential to improve the accuracy and reliability of medical data imputation and analysis.

The comprehensive evaluation across multiple healthcare datasets demonstrates CSAI's effectiveness in both data restoration and downstream predictive tasks, while maintaining alignment with clinical constraints. CSAI consistently outperformed established EHR imputers across multiple real-world healthcare datasets, particularly in scenarios with lower masking ratios.

However, CSAI's significance extends beyond its superior performance metrics. Following rigorous code and compatibility testing, CSAI has been integrated into the `PyPOTS` library (introduced in Section 2.6). This integration represents formal recognition of CSAI as one of the leading neural network-based EHR imputers[2]. The incorporation of CSAI into PyPOTS—an open-source Python toolbox designed for machine learning tasks on partially observed time-series enables fair comparison with other state-of-the-art methods and provides a robust implementation for the research community.

[2]Please find CSAI under 'Available Algorithms': https://github.com/WenjieDu/PyPOTS

Chapter 6

Uncertainty-Aware Deep Attention Recurrent Neural Network for Heterogeneous Time Series Imputation

The previous chapter demonstrated the value of incorporating domain knowledge and short- and long-term dependencies through CSAI. Like BRITS, CSAI remains a single-layer model, which uses an adaptive decay mechanism and transformer-enhanced initialisation to achieve its performance. An architecture adaptable to depth maybe needed when dealing with more complex patient trajectories and much larger datasets, which simpler models might fail to handle - particularly when trying to identify subtle deterioration signals that emerge from the interplay of hundreds of measurements over time. Under such complex conditions, quantifying uncertainty becomes crucial because imputation errors within interconnected physiological measurements could propagate through the deep architecture and significantly impact any insight derived from the data.

This is the motivation behind the work presented in this chapter, DEARI (DEep Attention Recurrent Imputation), a deep framework that introduces uncertainty quantification in deep imputation models. Like CSAI, DEARI is built on the BRITS backbone, and provides two key innovations:

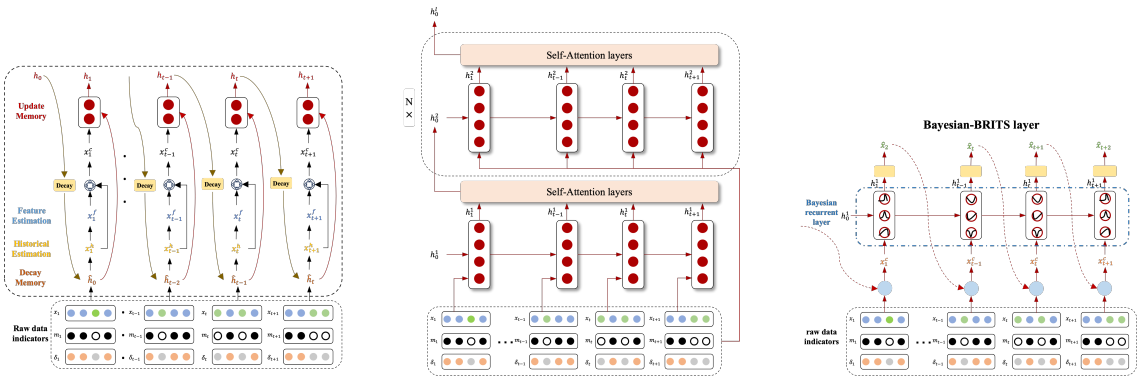
• DEARI enables deeper architectures by propagating a self-attention mechanism and a residual component through BRITS layers. DEARI comes with a reformulation of self-attention which overcomes information loss resulting from simply stacking BRITS layers. The standard stacking approach ignores estimated missing values in deeper layers, leading to no performance improvement. The residual component further complements the model by reducing loss with increasing depth, making the DEARI architecture highly scalable to depth and enabling more flexible and generalisable architectures for large and complex datasets.

• Where CSAI focused on accurate imputation through domain knowledge, DEARI extends this by adding uncertainty quantification through a Bayesian marginalisation strategy that provides trustworthy confidence bounds. This is achieved by incorporating the framework into a Bayesian neural network. DEARI overcomes the difficulties in training Bayesian NNs (discussed in Chapter 3) via a simple but practical training strategy to trade off exploration with exploitation, achieving high performance in large datasets.

The remainder of this chapter details DEARI's methodology, provides comprehensive experimental evaluation across multiple healthcare datasets, and discusses the implications of this work for clinical applications where understanding prediction reliability is crucial.

6.1 The DEARI Model

Two components enable scalable and flexible imputation via DEARI Fig. 6.1b 6.1c, namely: 1) **the deep-attention recurrent neural network**, which is a modified multilayer BRITS that has been augmented by a modified self-attention mechanism; and 2) **the Bayesian stochastic marginalisation strategy**, which quantifies uncertainty using Monte Carlo simulations.



(a) BRITS (b) DEARI (c) Bayesian

Fig. 6.1 Architectural evolution from BRITS to DEARI. (a) The baseline BRITS model showing its bidirectional recurrent structure. (b) The proposed DEARI architecture featuring multiple stacked layers enhanced with self-attention mechanisms and residual connections, which reduces to BRITS when using a single layer. (c) The Bayesian adaptation of DEARI incorporating uncertainty quantification through a Bayesian neural network. The progression demonstrates how DEARI builds upon BRITS by adding depth, attention mechanisms, and uncertainty awareness while maintaining the core bidirectional imputation strategy.

6.1.1 Deep Attention Recurrent neural network

An RNN cell is a non-linear transformation that maps the input signal x_t at time t and the hidden state of the previous time step h_{t-1} to the current hidden state h_t. Therefore, a standard recurrent network can be represented as:

$$h_t = \sigma(W_x x_t + W_h h_{t-1} + b_h) \tag{6.1}$$

When multiple RNN cells are stacked on top of each other, the hidden state of a layer $l-1$ is passed as input to the next-level layer l. This corresponds to the following recurrence relation:

$$h_t^l = \sigma(W_x' h_t^{l-1} + W_h' h_{t-1}^l + b_h') \tag{6.2}$$

Therefore, when stacking RNN layers, the first layer accepts raw input data - Eq 6.1, while subsequent layers receive the hidden state output of their previous layer - Eq 6.2. Common techniques for initialising the first hidden states of each layer l, h_0^l include zero initialisation, random initialisation, or initialisation according to a distribution. In practice, however, initial hidden states are set to $h_0^l = 0$ [3], which is the approach followed by BRITS.

However, the hidden states of the recurrent component of BRITS are not generated through the raw input. Instead, they receive incomplete data with missingness indicators for imputation - , i.e., $\hat{h}$ replaces h in Eq 6.2 in every layer of a stacked BRITS. Therefore, since information decay starts at the initial hidden state of each layer, initialising h_0^l to zeros in every layer will cause h_{t-1}^l to ignore previously estimated missing value and to rely on observations and corresponding indicators while losing useful information accumulating from previous layers. This also poses the risk of amplification of the errors caused by missingness due to continuous representation (hidden state) learning by simple stacking without a specific design.

One way to solve this issue is to simply replace h_0^l with the last hidden state of the previous layer, i.e. h_t^{l-1}. However, successful work in computer vision, specifically the Feature Pyramid Network [94], showed that concatenating all hidden state columns of the previous layer creates a temporal pyramid map carrying all information 'learnt' in the previous layer into the initial hidden state h_0^l. The intuition is provided in Eq 6.3.

$$h_0^l = f([h_0^{l-1}, h_1^{l-1}, ..., h_t^{l-1}]) \tag{6.3}$$

However, such initialisation will incur more computational complexity (a recurrent layer requires $\mathcal{O}(n)$ sequential operations). To enable capturing long-term dependencies of the fine-grained imputation task while managing the total computational complexity per layer, I adopt the self-attention mechanism [170] to better abstract an embedding of the temporal pyramid map.

Initialisation through Self-Attention To incorporate self-attention to our hidden state initialisation, I follow an approach similar to BERT [39]. I first prepend a learnable embedding ([CLS] token) to the hidden states of the previous layer - Eq 6.4. Each transformer encoder layer consists of two sublayers, a) a multiheaded self-attention (MSA) - Eq 6.5 and b) a feed-forward network (FFN) - Eq 6.6. A residual connection [60] is employed around each layer ($\hat{h}^{l-1}$ in MSA and $(\hat{h}^{l-1})'$ in FFN) followed by a layer norm (LN). Finally, CLS, which is used to set $\hat{h}^l_0$ of a layer l, is chosen as the first sequence of the transformer output $(\hat{h}^{l-1})^*$. Please note that in the first layer, $\hat{h}^1_0$ is initialised by all zeros as in BRITS.

$$\hat{h}^{l-1} = [\text{CLS}, \hat{h}^{l-1}_0, \hat{h}^{l-1}_1, ..., \hat{h}^{l-1}_t] \tag{6.4}$$

$$(\hat{h}^{l-1})' = \text{LN}(\text{MSA}(\hat{h}^{l-1}) + \hat{h}^{l-1}) \tag{6.5}$$

$$(\hat{h}^{l-1})^* = \text{LN}(FFN(\hat{h}^{l-1})') + (\hat{h}^{l-1})') \tag{6.6}$$

$$\hat{h}^l_0 = \text{CLS}^* \in (\hat{h}^{l-1})^* \tag{6.7}$$

Learning with Residuals Training deep RNNs (ones with more than 5 layers) remains a challenge despite the use of self-attention, even when LSTM gating mechanisms are used [165] because attention is known to lose rank with network depth [42]. For our imputation model, this issue is compounded by the fact that the decayed input is used in lieu of the raw data. To this end, I employ another special design in our model to overcome information loss; each deep layer can access the raw data x as a residual component. I do this by replacing the decayed hidden state $\hat{h}^{l-1}_t$ by the raw input x_t in the recurrent component of DEARI -Eq 6.8. Since for a layer l, the initial state $\hat{h}^l_0$ is initialised via the CLS token as in Eq 6.7, $\hat{h}^l_{t-1}$ in the recurrence equation will encode information from the previous layer $l-1$ via its self-attention mechanism. The residual term will instead enable the recurrent component of the final deterministic DEARI - Eq 6.8 to overcome information loss and

guarantee that the model achieves good performance and reliable convergence, even when scaled to 10 layers, as our experiments will show.

$$h_t^l = \sigma(W_x' x_t + W_h' \hat{h}_{t-1}^l + b_h') \tag{6.8}$$

The overall loss is calculated by mean pooling of all layer losses over all time steps using the combined vector x_t^c in the forward (F) and backward (B) directions, adjusted by the consistency between the forward and backward imputations $c_{t_F}^l$ and $c_{t_F}^l$ - Eq 6.9. In the equation, α and β are used to adjust the imputation loss (between the combined vector and the ground truth) and consistency loss, respectively. In this work, I use the *mean absolute error* for $\mathscr{L}_e$ as in BRITS. In BRITS, the authors reported that using the combined vector at each layer to calculate loss has led to slow convergence and has therefore accumulated estimation errors using the individual layer components of x_t^c, i.e. $\hat{x}_t$, x_t^c and x_t^{fc}. In our case, I found that using the combined vector works well. I therefore, use it instead. My choice is further driven by an observation: in BRITS as in DEARI, x_t^{fc} is derived from $\hat{x}_t$, making the two components highly co-dependent and their individual contributions to the final imputation is *learned* at each layer. When the two components are examined individually, one implicitly assumes that their contribution to the final imputation loss is comparable, which may skew the overall loss calculation.

$$\begin{aligned} \mathscr{L} = \frac{1}{L}\sum_{l=1}^{L}\sum_{t=1}^{T}\Bigg\{ \alpha \Big[\mathscr{L}_e\left(\mathbf{x}_t, (x_{t_F}^c)^l\right) \\ + \mathscr{L}_e\left(\mathbf{x}_t, (x_{t_B}^c)^l\right)\Big] + \beta \mathscr{L}_e\left(c_{t_F}^l, c_{t_B}^l\right) \Bigg\} \end{aligned} \tag{6.9}$$

6.1.2 Bayesian Marginalisation Strategy

It is desirable to have precise predictions, but it would be much better to provide corresponding trustworthy bounds. I use a Bayesian Marginalisation Strategy [102] to represent uncertainty due to errors, lack of knowledge about the true parameters of the black box model or the noise inherent in the data itself [117]. A Bayesian Neural Network (BNN) focuses on uncovering the distributions behind the network weights (marginalisation) [37], allowing one to explicitly represent uncertainty by computing the variability of weights. Having a distribution instead of a single value also makes it possible to evaluate the robustness of the model. In other words, estimating with certainty is a separate task from developing confidence estimates.

I begin by transforming DEARI into a BNN. This reformulation is inspired by the BNN implementation of [13], which introduces uncertainty in the weights of the network. In DEARI, missingness patterns are mainly observed by the hidden states and updated iteratively. It is therefore sufficient to transform DEARI's recurrent component into a Bayesian recurrent neural network by representing each of the RNN weights, i.e. W'_x, W'_h and b'_h of Eq 6.8 by random sampling from Gaussian distributions over possible values- Eqs 6.10 - 6.12, yielding the Bayesian DEARI recurrence - Eq 6.13.

$$\widetilde{W_t^l} = N(0,1) \times \log(1 + e^{\rho_{W_t^l}}) + \mu_{W_t^l} \tag{6.10}$$

$$\widetilde{W_h^l} = N(0,1) \times \log(1 + e^{\rho_{W_h^l}}) + \mu_{W_h^l} \tag{6.11}$$

$$\widetilde{b_h^l} = N(0,1) \times \log(1 + e^{\rho_{b_h^l}}) + \mu_{b_h^l} \tag{6.12}$$

$$h_t^l = \sigma(\widetilde{W_x^l} x_t + \widetilde{W_h^l} \hat{h}_{t-1}^l + \widetilde{b_h^l}) \tag{6.13}$$

To train the BNN, I use [13]'s *Bayes by Backprop* (BBB) algorithm, which transforms the intractable posterior estimation problem into an optimisation problem and defines the loss - Eq 6.14 as the divergence (measured by expectation $\mathbb{E}$) between the approximate

distribution of model parameters Q and the truly Bayesian posterior P. W^i is the i^{th} Monte Carlo sample drawn the weight distribution.

$$\begin{aligned}\mathscr{L}(\theta) &= -\mathbb{E}_{Q(\theta)}[\log P(X \mid \theta)] + \mathrm{KL}[Q(\theta) \parallel P(\theta)] \\ &= \sum_{i=1}^{n} \{\log Q(W^i \mid \theta) - \log p(W^i) - \log P(X \mid W^i)\}\end{aligned} \tag{6.14}$$

Using this approach, however, is still problematic. Our model architecture is much more sophisticated than the feed-forward networks examined by [13]. Each weight in the Bayesian layer is determined by the parameters that control its distribution. In a Gaussian distribution (used in DEARI's recurrent layer), the mean and standard deviation double the overall parameters, leading to difficult conversion and long training. This is compounded by the effect of multiple sampling on increasing the number of model operations and consequently, computational cost.

In response, I propose a simple but efficient training strategy to trade-off exploration (Bayesian marginalisation) and exploitation (deterministic optimisation)- Algor 2. At predetermined training steps, DEARI's recurrent component expands from a deterministic RNN layer (line 3) to a Bayesian layer (line 5) to investigate the uncertainty accumulated thus far. We wrap the training procedure with the expansion-contraction behaviour, thus minimising the overhead incurred by the BNN. In the current implementation, we define the Bayesian expansion behaviour to take place every 100 training step, where each step is calculated using the batch-size, the number of samples in the training set as well as the number of epochs.

6.2 Experimental Evaluation

In this section, we evaluate and carefully analyse DEARI's performance against the state of the art using five real-world healthcare, environment, and traffic datasets. We compare

Algorithm 2 Bayesian Marginalisation Strategy

1: $expand = f(batch\_size, n\_samples, epochs)$
2: **if** $\neg expand$ **then**
3: $\quad h_t^l = \sigma(W_x^l x_t + W_h^l \hat{h}_{t-1}^l + b_h^l)$
4: **else**
5: $\quad h_t^l = \sigma(\widetilde{W_x^l} x_t + \widetilde{W_h^l} \hat{h}_{t-1}^l + \widetilde{b_h^l})$
6: **end if**

DEARI against BRITS, GRUD, V-RIN(full) and MRNN. In all experiments, only the best model from each study is used for comparison.

6.2.1 Implementation Details

I used the Adam optimiser and set the number of RNN hidden units to 108 for all models. The batch size is 64 for PhysioNet data and 128 for the rest. All datasets are normalised with zero mean and unit variance for stable training. Because these experiments evaluate the merit of adding depth, I used random masking. I randomly mask $\{5\%, 10\%, 20\%\}$ observations in each dataset as the ground truth (validation data). 5-fold cross-validation is used to evaluate the models.Imputation performance is evaluated using mean absolute error (MAE) and mean relative error (MRE).

6.2.2 Experimental Results

Table 6.1 compares different combinations of DEARI components (a total of 7) and five baseline methods. The original BRITS uses an LSTM cell, but I also implemented a GRU-based BRITS, which performs better in 6 of the experiments performed. The training process was made uniform across all experiments. To enable reproducible comparisons, I implement the 3-layer DEARI, using mean pooling and the ['CLS'] token for DML on BRITS and DEARI respectively. Bayesian DEARI was implemented by unfreezing Bayesian layers every 100 steps and generating 10 simulations for all Bayesian components. The results show that DEARI outperforms BRITS in all experiments. The best DEARI model for the task, however, varies, and optimising the model for the task is part of

	MIMIC-III (59)			MIMIC-III (89)		
	5%	10%	20%	5%	10%	20%
Bayesian BRITS	0.09393 (0.23036)	0.11176 (0.27545)	0.13007 (0.32035)	0.28065 (0.43634)	0.29956 (0.46418)	0.31491 (0.49188)
Bayesian BRITS DML	0.09321 (0.22856)	0.11230 (0.27677)	0.13108 (0.32284)	0.28103 (0.43693)	0.30031 (0.46535)	0.31540 (0.49264)
Bayesian DEARI	0.09499 (0.23295)	0.10904 (0.26876)	**0.12457 (0.30679)***	0.26684 (0.41487)	0.28214 (0.43720)	0.29590 (0.46218)
Bayesian DEARI DML	0.09403 (0.23059)	0.10981 (0.27065)	0.12489 (0.30755)	0.26629 (0.41401)	0.28153 (0.43625)	0.29571 (0.46188)
BRITS DML	0.09532 (0.23380)	0.11622 (0.28639)	0.13151 (0.32389)	0.28135 (0.43746)	0.30088 (0.46625)	0.31741 (0.49577)
DEARI	**0.09165 (0.22476)***	**0.10789 (0.26591)***	0.12616 (0.31073)	**0.26496 (0.41196)***	**0.28088 (0.43524)***	0.29535 (0.46132)
DEARI DML	0.09168 (0.22485)	0.10845 (0.26730)	0.12602 (0.31036)	0.26533 (0.41254)	0.28091 (0.43530)	**0.29534 (0.46131)***
BRITS GRU	0.09581 (0.23501)	0.11496 (0.28339)	0.13109 (0.32289)	0.28114 (0.43713)	0.30092 (0.46631)	0.31744 (0.49583)
BRITS LSTM	0.09892 (0.24259)	0.11453 (0.28226)	0.13007 (0.32034)	0.28406 (0.44164)	0.30507 (0.47273)	0.32253 (0.50379)
V-RIN-full	0.10208 (0.25020)	0.11713 (0.28865)	0.13305 (0.32763)	0.28530 (0.44357)	0.30327 (0.46996)	0.32234 (0.50348)
GRUD	0.39554 (0.97070)	0.40830 (1.00721)	0.44031 (1.08499)	2.60073 (4.04391)	2.65272 (4.11172)	2.72067 (4.25052)
MRNN	0.25115 (0.61589)	0.25336 (0.62415)	0.25942 (0.63851)	0.50953 (0.79225)	0.51912 (0.80448)	0.52620 (0.82191)
	eICU			**PhysioNet**		
	5%	10%	20%	5%	10%	20%
Bayesian BRITS	0.1690 (0.2191)	0.1711 (0.2223)	0.1755 (0.2279)	0.2447 (0.3453)	0.2521 (0.3556)	0.2716 (0.3822)
Bayesian BRITS DML	0.1693 (0.2194)	0.1709 (0.2220)	0.1750 (0.2273)	0.2434 (0.3436)	0.2518 (0.3551)	0.2722 (0.3830)
Bayesian DEARI	**0.1582 (0.2051)***	**0.1602 (0.2081)***	**0.1645 (0.2136)***	0.2333 (0.3293)	0.2415 (0.3406)	**0.2592 (0.3648)***
Bayesian DEARI DML	0.1584 (0.2052)	0.1603 (0.2082)	0.1646 (0.2138)	**0.2329 (0.3288)***	0.2417 (0.3409)	0.2597 (0.3655)
BRITS DML	0.1733 (0.2246)	0.1704 (0.2213)	0.1755 (0.2279)	0.2484 (0.3506)	0.2561 (0.3613)	0.2766 (0.3892)
DEARI	0.1608 (0.2085)	0.1611 (0.2093)	0.1654 (0.2148)	0.2343 (0.3307)	0.2412 (0.3402)	0.2601 (0.3660)
DEARI DML	0.1610 (0.2086)	0.1615 (0.2098)	0.1657 (0.2152)	0.2337 (0.3298)	**0.2408 (0.3396)***	0.2599 (0.3658)
BRITS GRU	0.1734 (0.2247)	0.1703 (0.2212)	0.1753 (0.2277)	0.2477 (0.3496)	0.2564 (0.3617)	0.2772 (0.3901)
BRITS LSTM	0.1676 (0.2172)	0.1702 (0.2211)	0.1757 (0.2282)	0.2519 (0.3555)	0.2629 (0.3709)	0.2828 (0.3979)
V-RIN-full	0.2308 (0.2992)	0.2361 (0.3067)	0.2481 (0.3222)	0.2620 (0.3698)	0.2734 (0.3857)	0.2987 (0.4204)
GRUD	0.2213 (0.2868)	0.2241 (0.2911)	0.2293 (0.2977)	0.4858 (0.6857)	0.4960 (0.6998)	0.5015 (0.7057)
MRNN	0.4602 (0.5964)	0.4691 (0.6093)	0.4863 (0.6316)	0.5480 (0.7734)	0.5550 (0.7830)	0.5675 (0.7987)

Table 6.1 The mean absolute error (MAE) and mean relative error (MRE) for all datasets. The best results are marked by *.

ongoing research. GRU BRITS still holds SOTA over baseline models, but almost all the components and their combinations outperform GRU BRITS in every dataset. Bayesian DEARI works better in large datasets.

6.2.3 Comparison with CSAI

Comparing DEARI's performance the results obtained in Chapter 5 for CSAI and BRITS are shown in Figure 6.2. Both DEARI and CSAI demonstrate significant improvements over BRITS across most evaluated healthcare datasets. While CSAI achieves strong performance with MAEs of 0.25747, 0.11291, and 0.16149 on PhysioNet, MIMIC_59, and eICU respectively (at 10% masking ratio), DEARI shows comparable or better results with MAEs of 0.2412, 0.10789, and 0.1611 on the same datasets. The Bayesian variant of DEARI performs similarly to its non-Bayesian counterpart, with marginal improvements on some datasets like eICU (MAE 0.1602). The performance differences between the models can be understood through the datasets' characteristics (please see details in Section

2.8: DEARI shows the most significant improvements on MIMIC datasets, which have the highest baseline missingness (78%) and feature complexity (89 variables including 20 static/categorical variables). This suggests that DEARI's deep architecture is particularly valuable for complex, highly missing datasets where its attention mechanism can better capture intricate dependencies. On PhysioNet, which has fewer samples (3,997) but moderate missingness (51%), DEARI's uncertainty quantification proves beneficial, improving MAE by about 5% over CSAI. For eICU, which has more samples (30,680) but lower missingness (40.53%) and simpler feature relationships (only 9 static/categorical variables), the performance difference with BRITS is minimal, indicating that simpler architectures may suffice for less complex scenarios. These patterns demonstrate how DEARI's architecture - particularly its deep attention mechanism and uncertainty quantification - provide increasing benefits as dataset complexity and missingness increase.

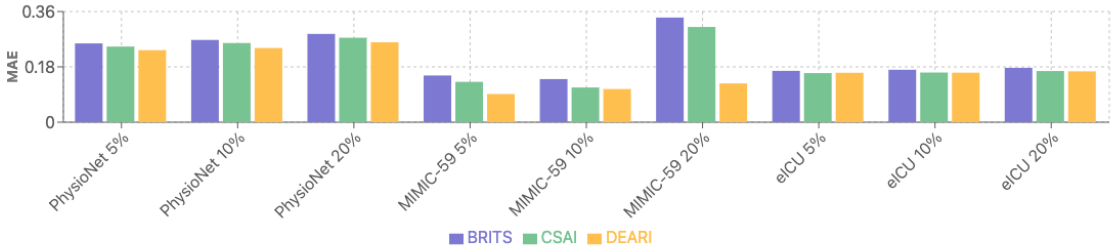


Fig. 6.2 Comparison of BRITS, CSAI, and DEARI performance across different datasets and masking ratios. Results show MAE for each model on PhysioNet, MIMIC-59, and eICU datasets with 5%, 10%, and 20% masking ratios. Lower MAE values indicate better performance, with DEARI generally achieving the lowest errors across most configurations.

6.2.4 Ablation Study

I now turn to evaluating the contributions of the different DEARI components using the PhysioNet Challenge 2012 dataset and a 10% masking ratio.

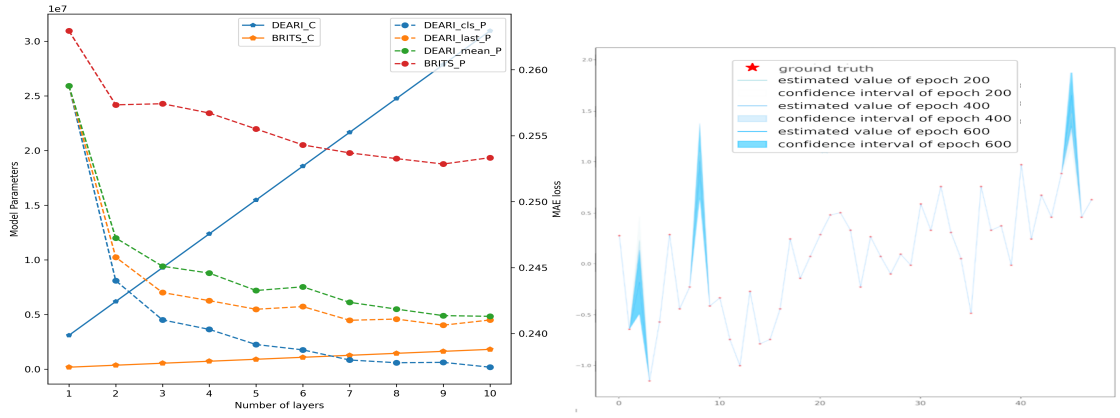
Model Complexity The solid lines of Fig 6.3a evaluate the effect of model depth on model complexity in both BRITS (BRITS_C) and DEARI (DEARI_C) measured by the

number of model parameters. The figure shows that model capability is directly related to the number of parameters. For DEARI, each additional 2-encoder layer increases the parameters by 2,149,330 (2×90,577 - parameter BRITS backbones, one in each direction, and 4×492,044 - parameter encoders, 2 in each direction). Fig. 6.3a shows the relationship between model size and performance; the introduction of the transformer encoder (self-attention) dominates both the parameter increase and the performance benefits. In other words, the proposed approach benefits mainly from the application of attention mechanisms. However, it is clear that model complexity and computational requirements grow linearly with the number of layers, while the performance improvement is eventually limited in this specific dataset. Nevertheless, compared to BRITS, DEARI's flexibility provides new possibilities for larger and more complex datasets. Reflecting on successful big transformer models, the BRITS backbone maybe a limiting factor constraining DEARI's generalisation, which motivates us to consider pure attention models in future work.

Trustworthy Boundary Fig. 6.3b shows an example of imputation confidence interval generated by the Bayesian model by sampling several well-optimised deterministic models. Confidence in the imputation can be quantified from corresponding boundaries; the more concentrated the result values, the smaller the confidence interval, signalling less uncertainty. As the figure shows, the estimation is generally successful in capturing the dynamic changes over time. The area of the shaded region is smaller when getting close to the observations, which is consistent with the temporal decay principle: the contribution of an observation to imputation decays with time.

6.3 Chapter Summary and Significance

This chapter introduced DEARI, a novel deep uncertainty-aware framework that takes an approach different to CSAI for handling EHR complexity through architectural depth and uncertainty quantification. While CSAI focused on incorporating domain knowledge and



(a) Model Performance and Complexity

(b) Confidence Interval for Systolic Blood Pressure

Fig. 6.3 Ablation study and DEARI's confidence intervals for systolic blood pressure. In (a): **C**: Model complexity (number of parameters); **P**: Performance (MAE loss)

temporal patterns, DEARI addresses the challenge of complex patient trajectories through a scalable deep architecture that can capture subtle patterns emerging from the interplay of hundreds of measurements over time. The model's key innovations - its deep attention mechanism and Bayesian marginalisation strategy - work together to enable reliable imputation even in scenarios with high dimensionality and complex missing patterns. The comprehensive evaluation across multiple healthcare datasets demonstrates DEARI's effectiveness, with particularly strong performance on complex datasets with high missingness and many variables. For instance, on the MIMIC dataset with 78% missingness and 89 features, DEARI achieved significant improvements both BRITS and CSAI, highlighting how its deep architecture effectively captures intricate dependencies in complex healthcare data. The model's uncertainty quantification also proved valuable for datasets with moderate complexity but limited samples, like PhysioNet, where understanding prediction reliability is crucial for clinical applications.

DEARI is currently in the final stages of verification for incorporation into `PyPOTS`. Despite the promising results, I have delayed submitting DEARI for publication until it is fully incorporated into the `PyPOTS` package.

Chapter 7

Modular Efficient Transformer for Health Outcome Discovery

Recent advances in transformer architectures have revolutionised natural language processing through large language models (LLMs), as discussed in Chapter 2.7. These innovations, from efficient attention mechanisms to advanced training strategies, have demonstrated remarkable potential for learning complex patterns in sequential data. However, their application to healthcare domains presents unique challenges, particularly in modelling patient trajectories where data distributions, temporal dependencies, and contextual relationships differ substantially from traditional language tasks.

While recent attempts to develop healthcare foundation models have shown promise in standardising patient timeline representations, current approaches often rely on generic transformer architectures that may not fully capture the patterns specific to clinical data. This chapter presents METHOD (Modular Efficient Transformer for Health Outcome Discovery), a novel architecture that adapts and extends modern transformer innovations specifically for healthcare applications. METHODS uses the tokenisation strategy of a recent medical Transformer model: ETHOS [135].

The remainder of this chapter is organised as follows: Section 7.1 presents an overview of the ETHOS model, whose tokenisation strategy is used in my work. Section 7.2 presents

the technical details of METHOD's architecture and its key components. Section 7.3 describes the enhanced data processing framework that enables effective learning from patient timelines. Section 7.4 provides comprehensive experimental validation across multiple healthcare datasets and tasks.

7.1 Overview of ETHOS

The Enhanced Transformer for Health Outcome Simulation (ETHOS) is a recent addition to the healthcare generative models [135]. It is distinguished from previous models by being the first to be specifically designed for numerical clinical data. ETHOS represents a significant advance in healthcare trajectory modelling through its innovative tokenisation strategy that transforms continuous clinical variables into discrete tokens while preserving clinical meaning. For those reasons, I have chosen ETHOS to be the baseline model against which I compare METHOD's performance. This section examines ETHOS's core methodological contributions and identifies key technical challenges that motivate the development of METHOD.

7.1.1 Core Methodological Innovations

ETHOS addresses three fundamental challenges in applying transformer architectures to healthcare data:

Clinical Event Discretisation ETHOS employs a unified decile-based discretisation strategy for all continuous variables in electronic health records. This approach transforms both clinical measurements (e.g., laboratory values, vital signs) and temporal intervals into categorical tokens through quantile-based binning. Specifically, continuous values are mapped to one of ten ordinal tokens (Q1-Q10) based on their position within the empirical distribution. This standardized approach offers computational advantages for transformer architectures while attempting to maintain relative relationships between values.

Temporal Context Integration A distinctive feature of ETHOS is its explicit encoding of temporal information through specialised separator tokens. This design directly addresses the irregular sampling and variable time gaps characteristic of medical data. By incorporating time-aware tokens, ETHOS enables transformer models to recognise both acute changes (e.g., rapid deterioration in vital signs) and gradual progression patterns (e.g., disease evolution over weeks).

Structured Event Relationships The framework preserves clinical event dependencies through a structured vocabulary that reflects established medical ontologies. This alignment with domain knowledge facilitates interpretability and enables the model to capture clinically meaningful patterns across diverse patient cohorts.

7.1.2 ETHOS Technical Limitations and Challenges

However, this universal quantile-based discretisation introduces several methodological challenges that motivate the development of METHOD:

- **Variable-Specific Information Loss**: Different clinical variables exhibit distinct distributions and clinically significant thresholds. A uniform decile-based approach may not adequately preserve critical diagnostic boundaries. This is particularly relevant for critical care scenarios where subtle changes can indicate important physiological shifts.

- **Distribution Preservation**: The transformation from continuous measurements to discrete tokens potentially alters the statistical properties of clinical variables. This distribution shift could affect the model's ability to capture true physiological relationships.

- **Scale-Dependent Sensitivity**: The sensitivity of the tokenisation varies with the scale and distribution of the underlying variable. For variables with heavy-tailed

distributions, significant clinical variations might be compressed into a single quantile, while clinically insignificant variations in the dense regions might span multiple tokens.

- **Temporal Resolution Challenges**: Applying the same decile-based discretisation to time intervals may not optimally capture the multi-scale nature of clinical events, where both rapid physiological changes (minutes to hours) and long-term disease progression (months to years) carry clinical significance. The token-based representation must balance sequence length constraints against the need to capture extended patient histories, particularly in chronic disease management.

7.1.3 Motivation for METHOD

The design of METHOD is driven by the unique characteristics of patient timelines and the challenges inherent in modelling healthcare data. Unlike conventional transformer architectures that assume uniform sequential dependencies, patient trajectories exhibit complex temporal structures, heterogeneous event distributions, and hierarchical dependencies that must be explicitly accounted for. This section presents the theoretical foundations and technical details of METHOD's key architectural innovations.

7.2 The METHOD Architecture

The design of METHOD is driven by the unique characteristics of patient timelines and the challenges inherent in modelling healthcare data. Unlike conventional transformer architectures that assume uniform sequential dependencies, patient trajectories exhibit complex temporal structures, heterogeneous event distributions, and hierarchical dependencies that must be explicitly accounted for. To address these challenges, METHOD introduces a series of architectural enhancements specifically tailored to clinical data modelling.

7.2.1 Architectural Overview

METHOD extends the transformer architecture through three primary components designed to address the challenges identified in Section 7.1:

1. A **patient-aware attention mechanism** that ensures strict isolation of patient information while enabling efficient batch processing
2. An **adaptive sliding window attention** scheme that captures multi-scale temporal dependencies
3. A **U-Net inspired architecture** with dynamic skip connections for effective long sequence processing

These components are integrated within a unified mathematical framework that preserves both computational efficiency and clinical interpretability.

7.2.2 Patient-Aware Attention Mechanism

A core innovation in METHOD is the patient-aware attention mechanism, which extends traditional transformer architectures to accommodate the hierarchical and event-driven nature of medical sequences. Standard causal attention enforces a strict autoregressive structure, which can be suboptimal for healthcare applications due to the following reasons:

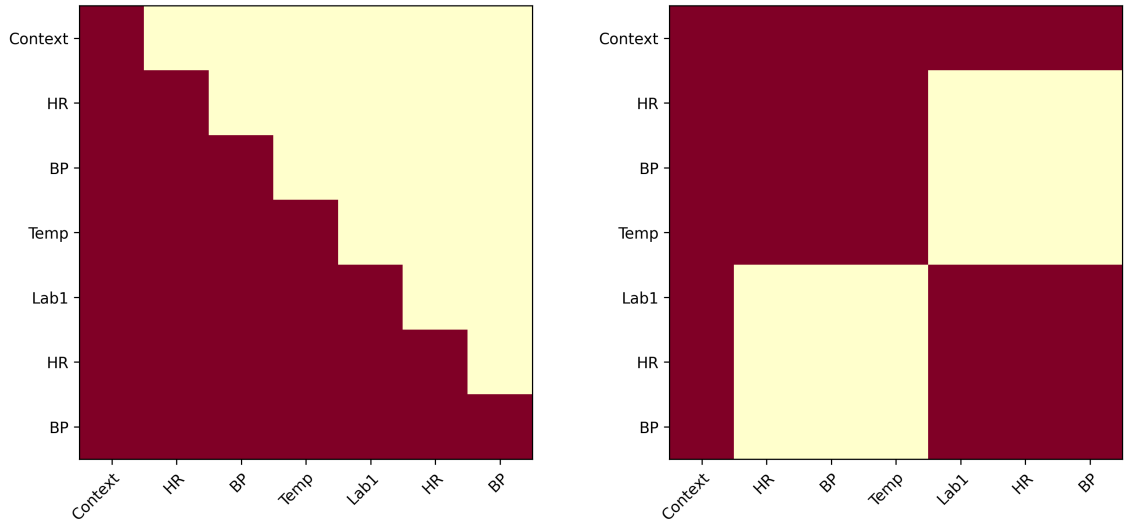
- **Contextual medical information should be globally accessible:** Static patient attributes, such as demographic factors and prior medical history, are essential for interpreting dynamic medical events and should remain available throughout a patient's sequence.
- **Inter-patient information leakage must be prevented:** Given that patient datasets are often batched together for computational efficiency, the model must ensure that information from one patient's timeline does not inadvertently influence another's.

- **Causal dependencies within a patient's trajectory must be preserved:** Medical events follow strict temporal causality (e.g., interventions occur in response to prior diagnoses), and the model should respect these constraints.

To enforce these constraints, METHOD utilises a **patient-aware block masking strategy** in conjunction with FlexAttention. The mask function ensures that:

$$M_{ij} = \begin{cases} 1 & \text{if } (i \geq j) \wedge (p_i = p_j) \\ 1 & \text{if } j \text{ is a static context token} \\ 0 & \text{otherwise} \end{cases} \tag{7.1}$$

where M_{ij} defines the attention mask between tokens at positions i and j, and p_i represents the patient identifier.



(a) Causal Mask

(b) Patient-Aware Mask

Efficient Implementation via FlexAttention To implement patient-aware attention efficiently, METHOD integrates **FlexAttention** [34] with a structured block mask:

$$\text{FlexAttn}(Q,K,V,M) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}} \odot \text{block\_mask}(M)\right)V \tag{7.2}$$

where block_mask(M) enforces patient-specific constraints and $\odot$ denotes element-wise multiplication.

Additionally, to enhance numerical stability, RMSNorm is applied to the query and key vectors before attention computation:

$$Q' = \text{RMSNorm}(Q), \quad K' = \text{RMSNorm}(K) \tag{7.3}$$

Theoretical Analysis The patient-aware attention mechanism provides several theoretical guarantees:

1. **Information Isolation**: The block masking strategy ensures zero gradient flow between patient sequences during backpropagation, mathematically guaranteeing patient privacy.

2. **Context Preservation**: Static patient information maintains consistent gradient paths to all tokens within a patient's sequence, enabling effective integration of contextual medical knowledge.

3. **Computational Efficiency**: The structured block mask enables efficient implementation through modern attention optimisations while maintaining O(n) memory complexity per patient.

This formulation provides a theoretically sound foundation for handling patient-specific temporal dependencies while maintaining computational efficiency.

7.2.3 Efficient Long Sequence Modelling

Patient timelines in electronic health records (EHRs) span extended periods, often exhibiting irregularly sampled events with variable temporal resolutions. Capturing both fine-grained short-term dependencies (e.g., sudden physiological changes) and long-range

clinical trends (e.g., chronic disease progression) is crucial for effective predictive modelling in healthcare. METHOD addresses these challenges through an integrated framework that combines *sliding window attention*, *Rotary Position Encoding (RoPE)*, and *multi-scale U-Net-inspired skip connections*, optimising computational efficiency while preserving clinically relevant temporal structures.

Sliding Window Attention for Variable-Resolution Medical Data

Medical events in patient timelines are not uniformly distributed; some clinical observations, such as vital signs, are recorded at high frequency (e.g., every few minutes in intensive care), whereas others, such as laboratory tests, are measured sporadically over days or weeks. Standard transformer architectures struggle with this variability, as they assume fixed-length token positions and uniform attention mechanisms.

To address this, METHOD employs a **sliding window attention** mechanism that restricts attention computation within a dynamically defined window:

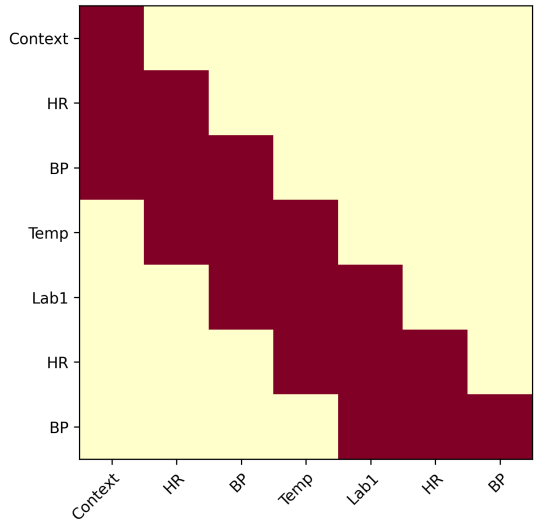
$$W_{ij} = \begin{cases} 1 & \text{if } |i-j| < w \wedge M_{ij} = 1 \\ 0 & \text{otherwise} \end{cases} \tag{7.4}$$

where w represents the sliding window size, which can also be dynamically adjusted during training as:

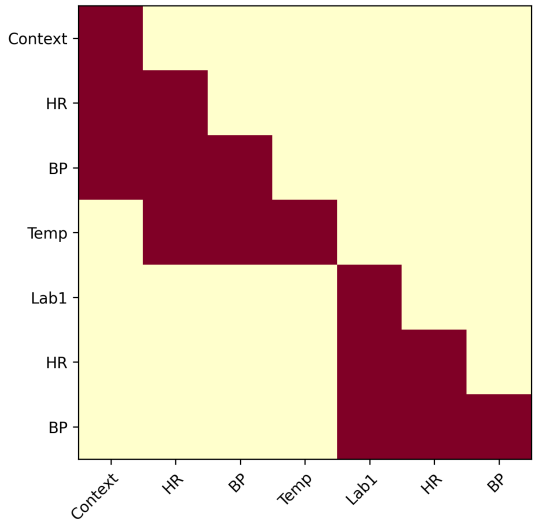
$$w_t = \min\left(w_{\text{base}} + \alpha \cdot \left\lfloor \frac{t}{L} \right\rfloor, w_{\max}\right) \tag{7.5}$$

Theoretical Properties The sliding window mechanism provides several key theoretical advantages:

1. **Local-Global Balance**: The adaptive window size enables efficient capture of both local patterns and global dependencies, with theoretical guarantees on information flow between distant tokens.



(a) Sliding Window Mask



(b) Combined METHOD Mask

2. **Memory Efficiency**: The windowed attention reduces memory complexity from $O(n^2)$ to $O(nw)$, where w is the window size, enabling the processing of longer sequences.

3. **Gradient Stability**: The restricted attention pattern provides more stable gradient paths during backpropagation, particularly beneficial for long sequence training.

Rotary Position Encoding for Clinical Sequences

Unlike traditional positional encodings that assume fixed sequence lengths, METHOD employs **Rotary Position Embeddings (RoPE)** to handle the variable temporal granularity inherent in medical data. The RoPE mechanism encodes relative positional information through rotational transformations:

$$\text{RoPE}(q,k,m) = \begin{pmatrix} \cos(m\theta) & -\sin(m\theta) \\ \sin(m\theta) & \cos(m\theta) \end{pmatrix} \begin{pmatrix} q \\ k \end{pmatrix} \tag{7.6}$$

where m represents the relative position and θ is a learnable frequency parameter. This formulation provides several advantages for medical sequence modelling:

1. **Scale Invariance**: The rotational nature of the encoding preserves relative distances regardless of sequence length, crucial for handling variable-length patient histories.
2. **Temporal Coherence**: The continuous nature of the encoding enables smooth interpolation of timelines, beneficial for irregularly sampled medical data.
3. **Hierarchical Structure**: The frequency parameter θ can adapt to different temporal scales, allowing the model to capture both rapid changes and long-term trends.

Multi-Scale Temporal Processing via U-Net Inspired Skip Connections

Medical event sequences exhibit variability in temporal granularity, requiring architectures capable of capturing both short-term and long-term dependencies effectively. To address this, METHOD integrates a **U-Net inspired architecture** with dynamically stored skip connections, ensuring robust multi-scale feature propagation across layers.

Unlike standard transformer architectures that process sequences in a strictly hierarchical manner, METHOD incorporates **learnable skip connections** that dynamically modulate information flow between encoder and decoder layers:

$$h_l = \text{RMSNorm}(\text{Attention}(x_l) + \lambda_l s_l) \tag{7.7}$$

where:

- h_l represents the hidden state at layer l
- s_l denotes the stored skip connection from earlier layers
- λ_l are **learnable weights** that adaptively balance information from different depths

The skip connections are maintained through a dynamic storage mechanism:

$$s_l = x_l \quad \text{(during encoding)} \tag{7.8}$$

$$x_l = x_l + \lambda_l s_l \quad \text{(during decoding)} \tag{7.9}$$

Theoretical Analysis The U-Net inspired architecture provides several theoretical advantages for medical sequence modelling:

1. **Multi-Resolution Feature Learning**: The hierarchical structure enables simultaneous capture of both fine-grained physiological patterns and coarse-grained clinical trajectories.

2. **Gradient Flow Enhancement**: Skip connections create additional gradient paths, mitigating the vanishing gradient problem, particularly prevalent in long medical sequences.

3. **Adaptive Information Integration**: Learnable weights λ_l allow the model to dynamically adjust the contribution of different temporal scales based on the clinical context.

7.2.4 Model Optimisations

METHOD incorporates several crucial optimisations to enhance both computational efficiency and clinical reliability:

Root Mean Square Normalisation To maintain numerical stability whilst preserving clinically relevant feature relationships, METHOD employs RMSNorm:

$$\text{RMSNorm}(x) = \frac{x}{\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}} \cdot \gamma \tag{7.10}$$

where γ represents a learnable scale parameter. This formulation offers advantages over traditional layer normalisation, particularly for medical data where preserving relative magnitudes is crucial.

Hybrid Optimisation Strategy METHOD utilises a novel hybrid optimisation approach combining the Muon optimiser with Adam. The update rule for matrix parameters follows:

$$\begin{aligned} v_t &= \beta v_{t-1} + (1-\beta) g_t \\ X_t &= \text{ZeroPower}(X_{t-1} - \eta v_t) \end{aligned} \tag{7.11}$$

where v_t represents the momentum vector and ZeroPower ensures orthogonality through Newton-Schulz iteration.

This hybrid approach provides several key advantages:

- **Enhanced Stability**: The combination of optimisers enables more stable training on heterogeneous medical data
- **Efficient Memory Usage**: Reduced memory footprint compared to standard adaptive optimisers
- **Improved Convergence**: Faster convergence on sparse, irregularly sampled medical time series

Implementation Considerations The practical implementation of METHOD requires careful attention to several technical aspects:

1. **Memory Management**: Efficient implementation of the sliding window attention mechanism through careful buffer management
2. **Numerical Stability**: Use of mixed precision training with dynamic loss scaling to maintain stability
3. **Batch Processing**: Strategic batch construction to maximise GPU utilisation while maintaining patient privacy constraints

These optimisations collectively enable METHOD to process long medical sequences efficiently whilst maintaining high prediction accuracy and clinical relevance.

7.3 Enhanced Data Processing Framework

The effectiveness of transformer-based models in healthcare depends significantly on the design of input sequences, as patient data inherently differs from conventional sequential data due to its multimodal nature, sparsity, and non-uniform temporal structure. Unlike domains where sequences exhibit regular sampling and clear segmentations, EHRs present unique challenges, requiring careful integration of static contextual data with dynamic medical events. METHOD introduces a tailored data processing framework that addresses these challenges through structured timeline construction, multi-patient sequence management, and a novel approach to temporal alignment.

7.3.1 Patient Timeline Construction

Patient timelines in EHRs are inherently **irregular**, **heterogeneous**, and **temporally asynchronous**, posing difficulties for conventional sequence models. METHOD adopts a structured approach to timeline representation that preserves the chronological and contextual integrity of medical events:

$$\mathscr{T} = \{(c, e_1, ..., e_T)\} \tag{7.12}$$

where c encapsulates static patient context (e.g., demographics, baseline conditions), and e_t represents time-stamped clinical events such as laboratory test results, vital sign measurements, and administered treatments.

Theoretical Foundations This structured representation offers several advantages:

- **Information Completeness**: Explicit separation between static characteristics and dynamic trajectories ensures no loss of contextual information
- **Temporal Coherence**: Preservation of clinical event ordering maintains causal relationships crucial for medical reasoning

- **Efficient Tokenisation**: Structured handling of irregular sampling mitigates information loss during discretisation
- **Cross-Patient Consistency**: Unified representation enables effective batch processing while maintaining patient-specific contexts

To address the challenge of boundary effects in sequence processing, METHOD introduces a circular data augmentation strategy:

$$\mathscr{C} = [\mathscr{T}_1 \xrightarrow{\text{wrap}} \mathscr{T}_2 \xrightarrow{\text{wrap}} \ldots \xrightarrow{\text{wrap}} \mathscr{T}_N \xrightarrow{\text{wrap}} \mathscr{T}_1] \tag{7.13}$$

where $\xrightarrow{\text{wrap}}$ denotes a circular connection that eliminates edge effects and ensures uniform training opportunity across all temporal positions.

Theoretical Properties The circular augmentation strategy provides several guarantees:

- **Positional Invariance**: Equal representation of all positions in the training process
- **Boundary Continuity**: Smooth handling of sequence boundaries without artificial padding effects
- **Training Stability**: Consistent gradient flow across all temporal positions during backpropagation

7.3.2 Multi-Patient Sequence Management

Beyond constructing individual patient timelines, METHOD employs a multi-patient sequence batching strategy, which enhances training efficiency while maintaining proper patient-event relationships. Given a predefined sequence length L, training sequences are constructed as:

$$\begin{aligned} &\mathscr{S} = [\mathscr{T}_1; \mathscr{T}_2; \ldots; \mathscr{T}_k] \\ &\text{s.t.} \quad \sum_{i=1}^{k} (|c_i| + |e_i|) \leq L \end{aligned} \tag{7.14}$$

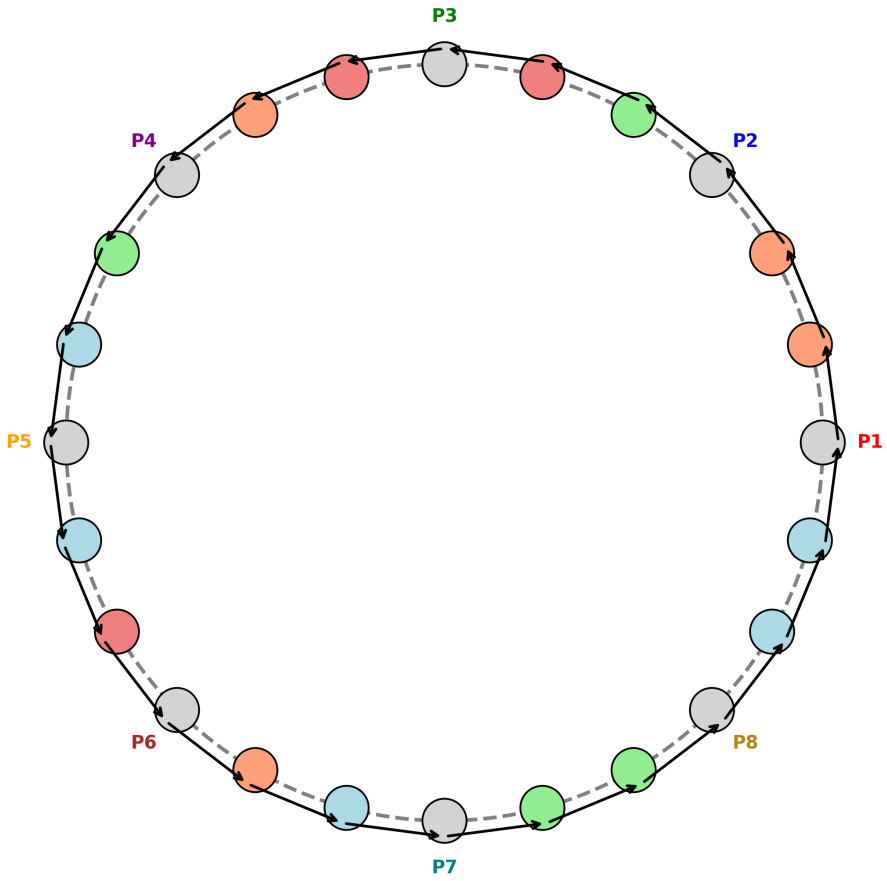


Fig. 7.3 Illustration of the multi-patient data ring used for unbiased sampling in the dataset construction. Each patient is represented as a sequence of events arranged in a circular structure. The **grey nodes** indicate the *static patient context*, which remains constant for a given patient, while the **coloured nodes** represent *time-stamped clinical events* occurring at different time points. The order and types of clinical events vary between patients. The outer arrows indicate the unbiased progression through patient sequences during data loading.

To enforce clear demarcation between patient timelines, METHOD introduces explicit indexing mechanisms:

$$\begin{aligned} p &= [p_1, \ldots, p_L] \quad \text{where } p_i \text{ denotes patient ID} \\ q &= [q_1, \ldots, q_L] \quad \text{where } q_i = 0 \text{ for context tokens} \end{aligned} \tag{7.15}$$

Theoretical Properties This formulation provides several guarantees:

- **Information Isolation**: Patient boundaries are explicitly maintained through index vectors
- **Memory Efficiency**: Optimal utilisation of computational resources through controlled sequence lengths
- **Context Preservation**: Clear distinction between static and dynamic information through the context indicator

7.3.3 Temporal Alignment and Causality

A critical challenge in medical sequence modelling is maintaining proper temporal causality while enabling efficient batch processing. METHOD addresses this through an event-aware alignment strategy where the prediction objective follows:

$$y_t = \begin{cases} x_{t+1} & \text{if } p_t = p_{t+1} \\ e_{\text{sep}} & \text{if } p_t \neq p_{t+1} \end{cases} \tag{7.16}$$

where $e_{\text{sep}} \in \mathscr{E}$ is a separator token that explicitly marks patient transitions.

Theoretical Analysis This formulation ensures several crucial properties:

1. **Causal Consistency**:
   - Strict preservation of temporal ordering within patient sequences
   - Clear separation between patient trajectories
   - Maintenance of clinical event dependencies
2. **Information Flow Control**:
   - Prevention of cross-patient information leakage
   - Proper handling of context token influence

- Controlled gradient propagation during training

3. **Training Stability**:
    - Consistent loss computation across patient boundaries
    - Stable gradient flows through separator tokens
    - Robust handling of variable-length sequences

7.3.4 Clinical Implications

The enhanced data processing framework in METHOD aligns with current best practices in medical informatics whilst addressing several key challenges:

Temporal Modelling Challenges

- **Irregular Sampling**: Robust handling of variably-sampled clinical measurements
- **Multi-scale Dependencies**: Effective capture of both acute changes and long-term trends
- **Missing Data**: Principled approach to handling gaps in clinical records

Clinical Safety Considerations

- **Patient Privacy**: Strict isolation of patient information during processing
- **Data Integrity**: Preservation of clinical relationships and temporal ordering
- **Model Interpretability**: Clear tracking of information flow through the architecture

Through this comprehensive framework, METHOD provides a theoretically sound foundation for processing medical sequences while maintaining clinical validity and computational efficiency. The framework's design choices are directly motivated by the unique challenges of healthcare data, ensuring that the resulting model can effectively capture and utilise complex clinical patterns.

7.4 Experimental Evaluation

7.4.1 Experimental Setup

We conduct our experiments on two versions of the MIMIC-IV database: version 2.2 and the latest version 3.1, details in Section 2.8. Version 3.1 introduces several significant enhancements over version 2.2, including additional hospital admission data, error corrections, and more complex clinical cases that better reflect real-world scenarios. To ensure fair comparison with existing approaches, we maintain consistent data processing pipelines across both versions, employing the same tokenisation strategy as established in previous work [135].

For the baseline ETHOS, we maintain identical hyperparameters as reported in the original paper to ensure fair comparison. For METHOD, we initially adopt these same hyperparameters to facilitate direct comparison, although this potentially understates METHOD's optimal performance. The impact of METHOD-specific hyperparameter tuning is reserved for future work. To ensure robust comparison, we:

- Reproduce ETHOS results on MIMIC-IV v2.2 using the original implementation
- Retrain ETHOS on MIMIC-IV v3.1 using identical hyperparameters
- Verify evaluation metrics through independent implementation

The baseline model was rigorously validated using the released pre-trained model and inference results, revealing notable inconsistencies in the reported SOFA score evaluation. Our reproduction yielded a *real-value MAE* of 2.2567 ($\pm$0.0053) compared to the originally reported 1.502, underscoring the critical need for transparent and standardised evaluation protocols in medical AI research [106].

7.4.2 Evaluation Metrics Design

The evaluation of METHOD presents unique challenges that extend beyond traditional performance metrics. The transformation of continuous SOFA scores into decile tokens, while necessary for the transformer architecture, introduces additional complexity in performance assessment. We propose a comprehensive evaluation framework that examines both the model's computational performance and clinical relevance.

Continuous Score Evaluation To assess clinical accuracy rigorously, we employ:

- **Mean Absolute Error (MAE)** of reconstructed SOFA scores
- **Root Mean Square Error (RMSE)** to penalise larger deviations
- **Pearson's Correlation** for trend analysis
- **High SOFA Performance Metrics** targeting critical cases (SOFA > 7)

Token-Level Assessment For evaluation in the transformed token space:

- **Ordinal Metrics**:
  - Kendall's Tau and Spearman's Rho for ordering preservation
  - Weighted MAE/MSE with exponential weights
- **Clinical Alignment Measures**:
  - Token-to-clinical mapping accuracy
  - Prediction margin analysis (Off-by-One/Two/Three)
- **High SOFA Token Performance**:
  - MAE for high severity tokens (Q8-Q10)
  - Critical case accuracy metrics
  - Error distribution analysis

Classification Performance To assess discriminative capability:

- **Token-wise AUC Scores**:
  - Per-token AUC scores (Q1-Q10)
  - Macro-averaged AUC
  - Weighted-averaged AUC
- **Support Analysis**:
  - Class-wise support statistics
  - Prediction reliability analysis

7.5 Core Performance Analysis

7.5.1 Impact of Training Sequence Length

We first examine how the training sequence length affects model performance by comparing models trained with sequences ranging from 1024 to 10240 tokens. Our analysis reveals several key insights about the relationship between sequence length and model effectiveness.

Performance Trends The relationship between sequence length and model performance reveals important insights about clinical trajectory modelling. Unlike traditional clinical prediction tasks where longer observation windows consistently yield better results, our analysis reveals a more nuanced relationship in the context of transformer-based models.

Statistical Analysis The improvement in continuous SOFA prediction from 1024 to 3072 tokens (MAE: 2.2123 $\rightarrow$ 2.1550, $p < 0.001$) suggests that longer sequences initially help capture more comprehensive clinical patterns. However, the plateauing effect beyond

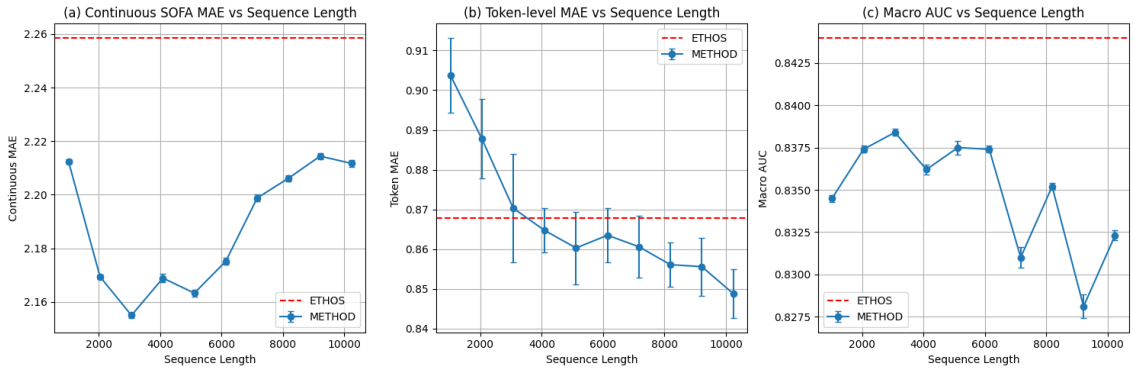


Fig. 7.4 Performance metrics across different training sequence lengths. (a) Continuous SOFA MAE shows initial improvement followed by stabilisation. (b) Token-level MAE demonstrates consistent improvement with longer sequences. (c) Macro AUC indicates the optimal performance of around 3072 tokens. Error bars indicate the standard deviation over 10 runs.

3072 tokens indicates a potential limitation in the model's ability to effectively utilise very long-term dependencies.

Of particular clinical significance is the model's performance on high-severity cases (SOFA > 7). The improvement in high SOFA MAE demonstrates METHOD's enhanced capability in critical care prediction, a crucial advantage given the disproportionate importance of accurate predictions in high-risk scenarios.

7.5.2 Clinical Semantic Alignment Analysis

Our experimental results reveal a concerning misalignment between computational and clinical performance metrics. As the training sequence length increases, we observe two contradictory trends:

- Token-level MAE shows consistent improvement, decreasing from 0.9038 ($\pm$0.0094) at 1024 tokens to 0.8488 ($\pm$0.0062) at 10240 tokens
- Continuous SOFA MAE exhibits a U-shaped curve, initially improving from 2.2123 ($\pm$0.0008) to 2.1550 ($\pm$0.0011) at 3072 tokens, but then degrading to 2.2117 ($\pm$0.0012) at 10240 tokens

Theoretical Analysis This divergence between token-level and continuous metrics suggests fundamental challenges in the current tokenisation approach:

1. **Granularity Limitations**:
    - Discretisation may be too coarse for subtle clinical variations
    - Adjacent tokens may not represent meaningful clinical gradients
    - Token boundaries may not align with clinically significant thresholds
2. **Distribution Shift Effects**:
    - Transformation alters statistical properties of clinical variables
    - Potential introduction of systematic biases
    - Loss of continuous value relationships

    Temporal Context Preservation:
    - Discrete tokens may inadequately capture temporal dynamics
    - Challenges in representing rate-of-change information
    - Potential loss of temporal dependency structure

7.5.3 Inference Length Flexibility

As shown in Figure 7.5, the model exhibits remarkable stability across different inference lengths, with performance metrics showing minimal variation. This stability is particularly evident in the Macro AUC scores, which maintain consistency (0.832 ± 0.0003) across all inference lengths. Furthermore, Figure 7.6 demonstrates that models trained on longer sequences (32768 tokens) consistently outperform their shorter counterparts while maintaining similar stability across inference lengths. Our analysis of inference length flexibility reveals remarkable stability across different sequence lengths, particularly for models trained with longer sequences (16384 and 32768 tokens). Key findings include:

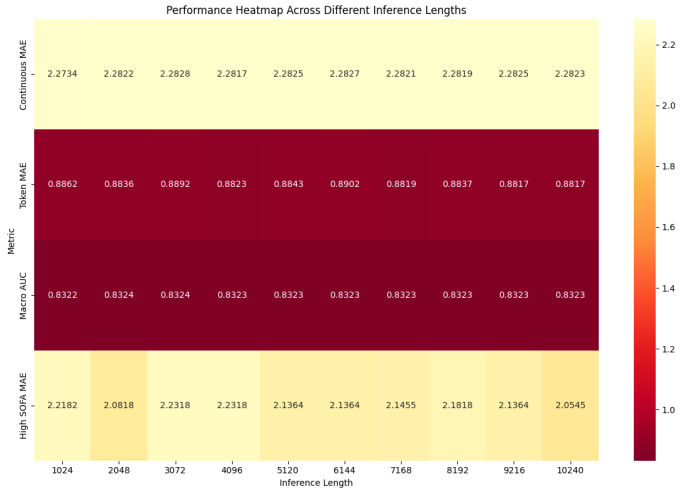


Fig. 7.5 Performance heatmap across different inference lengths for the model trained with 32768 sequence length. Darker colours indicate better performance. The relatively uniform colouring across inference lengths suggests stable performance regardless of inference sequence length.

Performance Stability

- Consistent Macro AUC scores (0.832 ± 0.0003) across all inference lengths
- Minimal variation in token-level prediction accuracy
- Robust performance on both short and long sequences

Computational Efficiency

- Linear scaling of memory usage with sequence length
- Maintained prediction speed across varying sequence lengths
- Efficient handling of long patient histories

One of METHOD's key innovations is its ability to maintain consistent performance across varying inference lengths, a crucial feature for real-world clinical deployment where patient histories vary significantly in length. The stability of performance metrics across different inference lengths (Macro AUC variation $< 1\%$) suggests robust generalisation capabilities, addressing a common limitation in clinical prediction models where performance often degrades with varying input lengths.

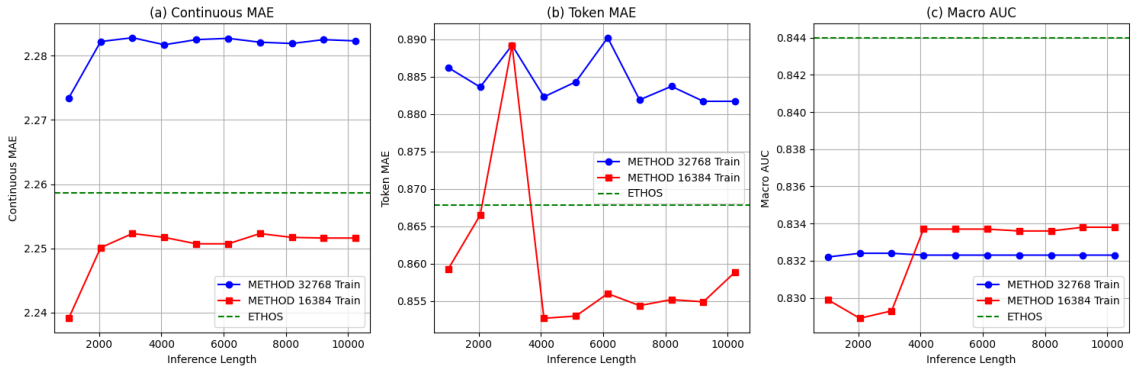


Fig. 7.6 Comparison of models trained with different sequence lengths (16384 vs 32768) across various inference lengths. (a) Continuous MAE shows consistent performance across inference lengths. (b) Token MAE demonstrates the stability of predictions regardless of inference length. (c) Macro AUC indicates robust discriminative ability across different sequence lengths.

7.5.4 Model Architecture and Clinical Reliability

As illustrated in Figure 7.6, models trained with longer sequences (32768 tokens) consistently outperform their shorter counterparts whilst maintaining similar stability across inference lengths. Figure 7.5 provides a comprehensive view of performance stability through a heatmap visualisation, where darker colours indicate better performance across different inference lengths.

The comparison between 6-layer and 12-layer architectures reveals important insights about the relationship between model capacity and clinical prediction reliability. The 12-layer model's superior performance (Continuous MAE: 2.28 $\rightarrow$ 2.10) suggests that deeper architectures better capture the complex interdependencies in clinical data. However, this improvement must be contextualised within clinical requirements:

1. **Clinical Significance**: The improvement in Kendall's Tau (0.5018 $\rightarrow$ 0.5338) indicates better preservation of clinical severity ordering, crucial for risk stratification.

2. **Uncertainty Calibration**: Deeper models show more consistent uncertainty estimates across severity levels, essential for clinical decision support.

3. **Resource Considerations**: The computational cost increase must be weighed against marginal performance gains, particularly in resource-constrained healthcare settings.

7.5.5 Analysis of High-severity Cases

The assessment of model performance in high-severity cases (SOFA > 7) is particularly crucial in clinical settings, as these cases often represent patients at greatest risk and require the most urgent interventions. Figure 7.7 demonstrates that MAE exhibits a non-monotonic relationship with sequence length, ranging from 2.33 ($\pm$0.22) at 1024 tokens to 2.39 ($\pm$0.20) at 10240 tokens, with local minima observed at 4096 and 6144 tokens.

This pattern, visualised in Figure 7.8, suggests that whilst longer sequences theoretically provide more clinical context, they may also introduce noise in critical care scenarios where recent observations carry greater prognostic weight. The confidence intervals, represented by the shaded regions, widen notably at longer sequence lengths, indicating increased prediction uncertainty—a crucial consideration for clinical deployment.

Accuracy measurements for high-severity cases demonstrate even more pronounced variability, with peak performance (0.154 $\pm$0.045) achieved at 4096 tokens. This optimal point likely represents a balance between sufficient temporal context and signal-to-noise ratio, aligning with clinical observations that diagnostic accuracy often depends on identifying relevant temporal windows rather than maximising observation periods. The wider confidence intervals at certain sequence lengths (particularly evident in the 3000-5000 token range) suggest that prediction reliability varies significantly across different patient trajectories.

7.5.6 Clinical Semantic Analysis via ICD Codes Embedding

Medical knowledge representation in foundation models presents unique challenges due to the complex hierarchical nature of clinical taxonomies and the intricate relationships between medical concepts. We conducted a detailed analysis of learnt token embeddings,

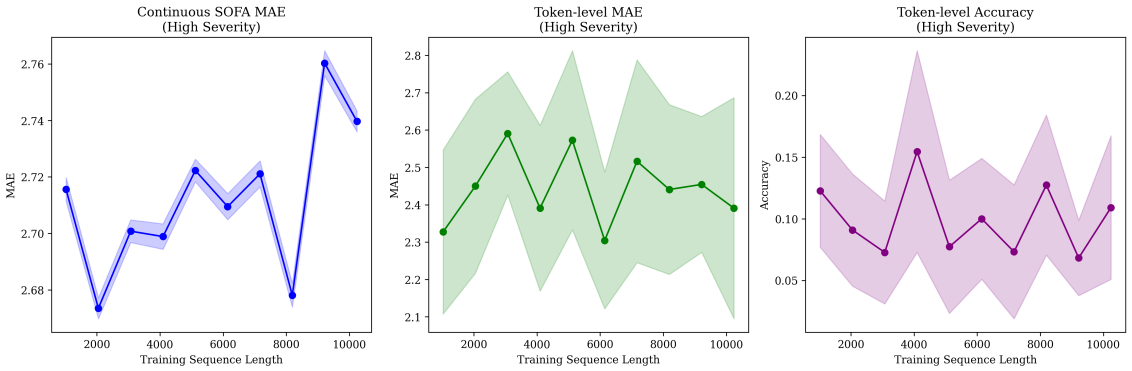


Fig. 7.7 Performance heatmap across different inference lengths for the model trained with 32768 sequence length. Darker colours indicate better performance. The relatively uniform colouring across inference lengths suggests stable performance regardless of inference sequence length.

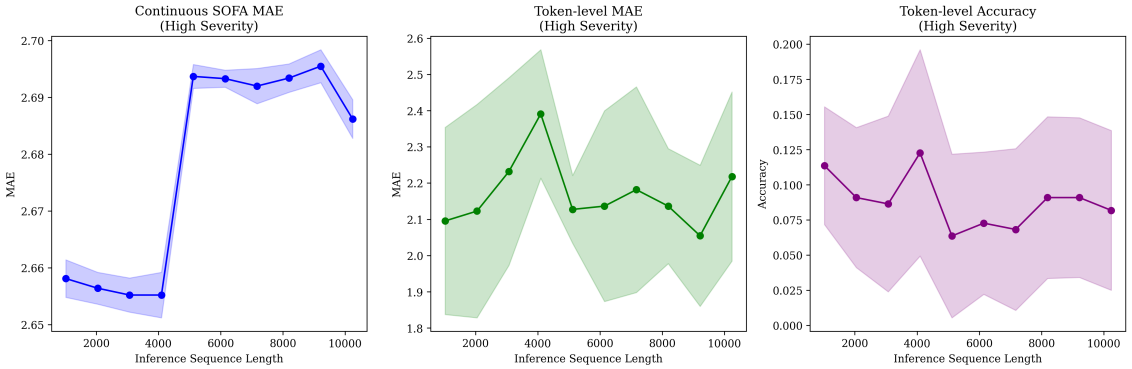
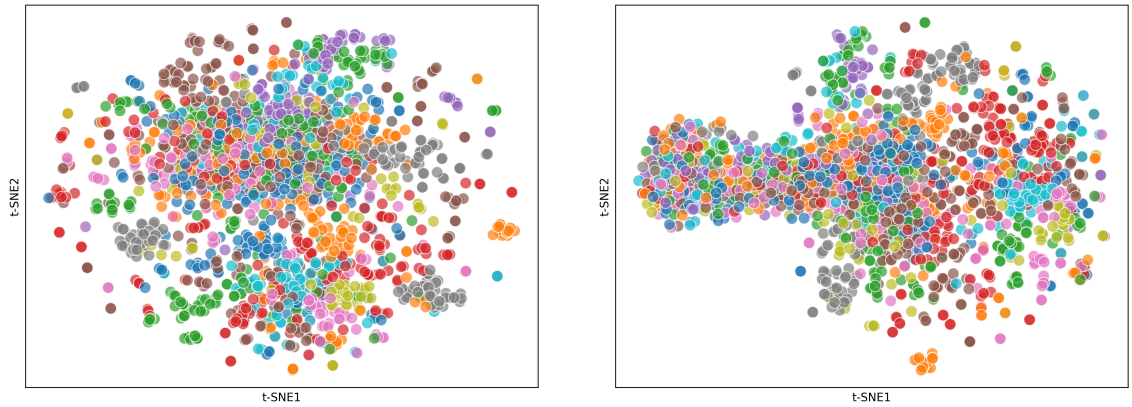


Fig. 7.8 Comparison of models trained with different sequence lengths (16384 vs 32768) across various inference lengths. (a) Continuous MAE shows consistent performance across inference lengths. (b) Token MAE demonstrates the stability of predictions regardless of inference length. (c) Macro AUC indicates robust discriminative ability across different sequence lengths.

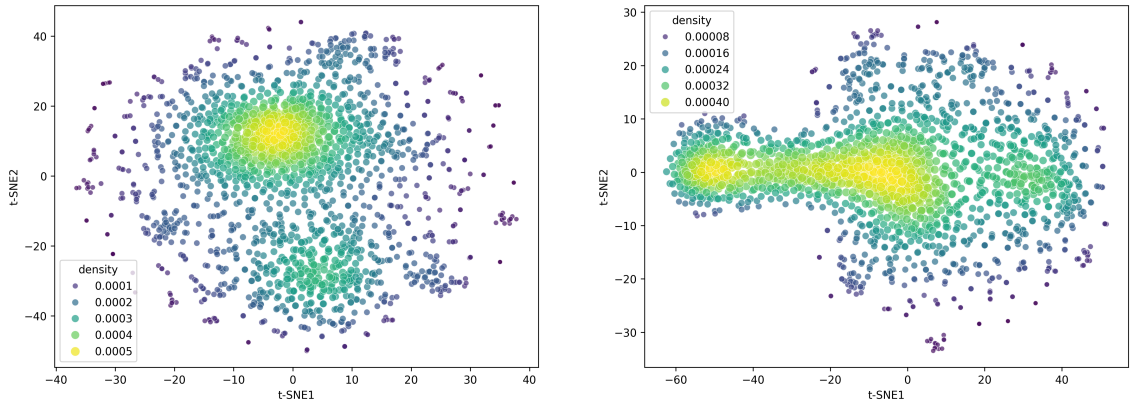
focusing particularly on ICD (International Classification of Diseases) codes, which represent a standardised medical classification system fundamental to clinical documentation and research.

We extracted embeddings for ICD tokens from both ETHOS and METHOD models post-training. The high-dimensional embeddings were projected into two dimensions using t-SNE, maintaining perplexity at 30 with 1000 iterations. This visualisation approach allows examination of how the models preserve clinically meaningful relationships between disease categories, an essential consideration for downstream clinical applications.



(a) TSNE Clustering on ETHOS (b) TSNE Clustering on METHOD

Fig. 7.9 Comparison of TSNE Clustering on ETHOS and METHOD



(a) Density Estimation of ETHOS (b) Density Estimation of METHOD

Fig. 7.10 Comparison of Density Estimation between ETHOS and METHOD

Comparative Analysis of Embedding Structures To evaluate METHOD's capacity for preserving clinical relationships, we conducted a comprehensive embedding analysis through multiple visualisation and clustering techniques. Figure 7.9 presents the t-SNE projections of ICD code embeddings, revealing distinct organisational patterns between ETHOS and METHOD.

Global Structure Analysis The density visualisations in Figure 7.10 illustrate fundamental differences in how the two models structure medical knowledge:

- **ETHOS** exhibits a spherical distribution with uniform density gradients, suggesting a generalised approach to medical concept representation. While this ensures broad

coverage of clinical relationships, it may oversimplify the complex hierarchical nature of medical knowledge.

- **METHOD** develops a manifold structure with varying density regions, indicating better preservation of clinical hierarchies. The emergence of distinct high-density clusters (shown in Figure 7.11) suggests the model has learnt to differentiate between major disease categories while maintaining relevant cross-category relationships.

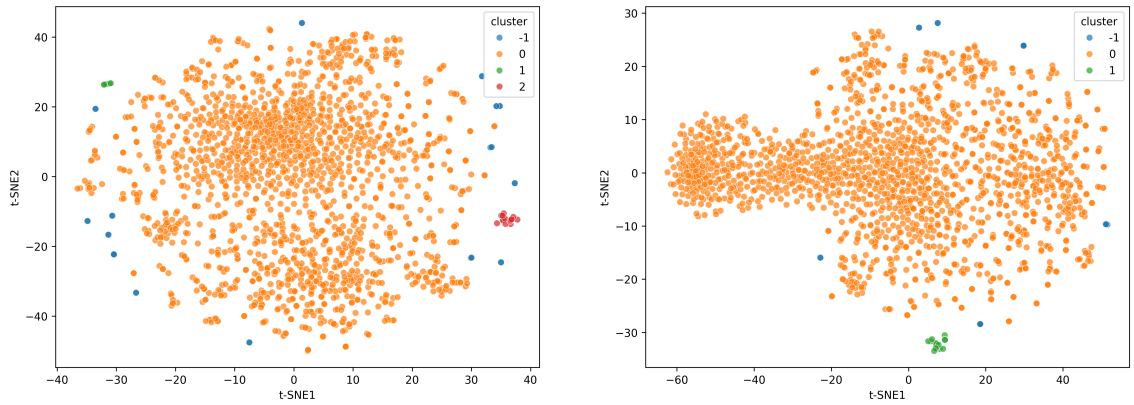
Local Pattern Analysis The similarity heatmaps in Figure 7.12 reveal fine-grained differences in how the models encode clinical relationships:

- Within high-density regions, METHOD demonstrates stronger intra-category similarities, particularly for closely related conditions within the same ICD chapter.
- The similarity difference heatmap (Figure 7.13) shows METHOD selectively strengthens certain clinical associations while weakening others, potentially reflecting real-world medical knowledge structure.

Clinical Implications These structural differences suggest both potential advantages and risks:

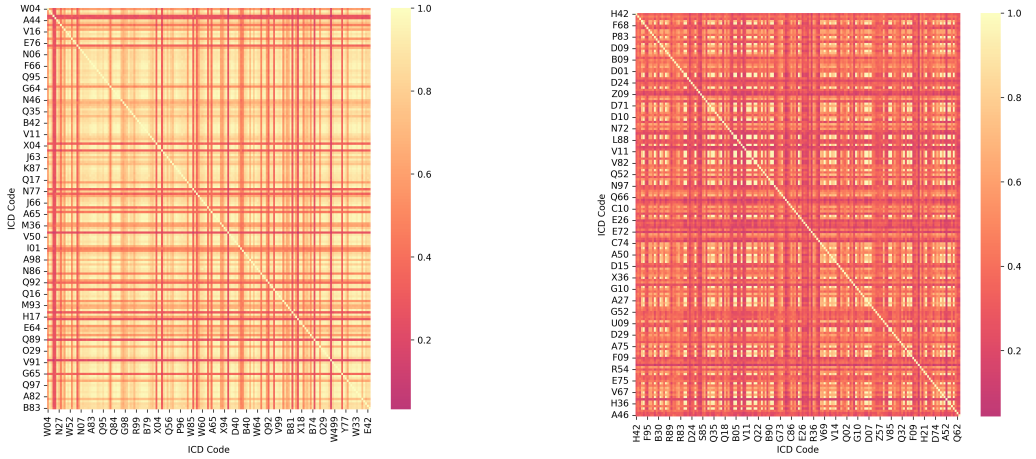
- METHOD's more structured embedding space may improve accuracy in tasks requiring fine-grained clinical discrimination, such as specific disease prediction or comorbidity analysis.
- However, the stronger clustering patterns could potentially lead to over-segmentation of the clinical space, making it important to validate the model's generalisation capabilities across different medical contexts.
- The observed patterns warrant further investigation into whether the enhanced structural organisation genuinely reflects meaningful clinical relationships or introduces unwanted biases in medical concept representation.

This analysis provides insights into how architectural choices influence medical knowledge representation, though the clinical significance of these differences requires validation through downstream task performance and expert evaluation.



(a) DBSCAN Clustering on ETHOS (b) DBSCAN Clustering on METHOD

Fig. 7.11 DBSCAN Clustering Results for ETHOS and METHOD



(a) High-Density ICD Code (ETHOS) (b) High-Density ICD Code (METHOD)

Fig. 7.12 High-Density ICD Code Similarity for ETHOS and METHOD

7.6 Chapter Summary and Significance

This chapter has introduced METHOD as a specialised transformer-based framework for clinical sequence modelling, addressing key challenges in healthcare AI. By optimising for

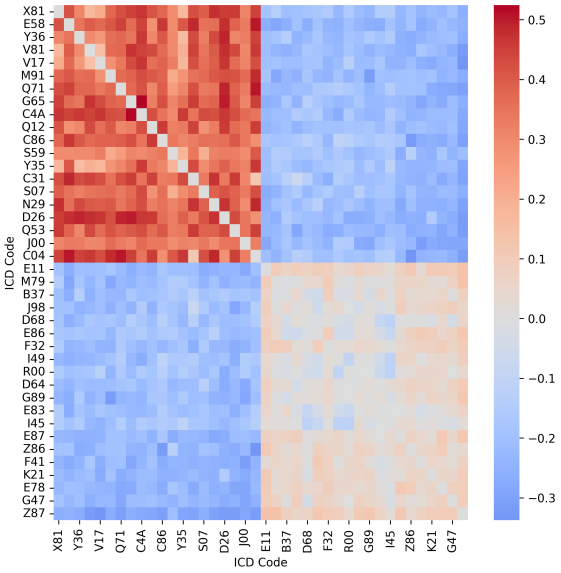


Fig. 7.13 ICD Code Similarity Difference (METHOD - ETHOS)

long-term dependencies, mitigating data heterogeneity, and enhancing computational efficiency, METHOD demonstrates significant improvements over conventional architectures.

Experimental results highlight its ability to balance predictive accuracy and clinical relevance. METHOD's patient-aware attention mechanism prevents cross-patient information leakage, while sliding window attention and multi-scale processing improve temporal representation, making it more effective for handling irregularly sampled medical data. These innovations align with broader efforts in medical AI to develop models that accommodate the complexity of patient trajectories without sacrificing interpretability.

Despite these advancements, challenges remain. Tokenisation strategies influence prediction reliability, and the computational demands of deeper architectures pose deployment concerns in resource-limited clinical environments. Further work is needed to refine model efficiency, validate clinical performance in real-world settings, and enhance interpretability for transparent decision support.

METHOD represents a significant step towards transformer architectures optimised for healthcare applications, offering a foundation for future research into clinically grounded AI models.

Chapter 8

Conclusions and Future Directions

This final chapter synthesises the contributions of this thesis, reflecting on their significance and outlining critical areas for future research while emphasising the work's broader implications. The chapter is structured as follows: Section 8.1 discusses the direct and broader impact of the thesis contributions. Here, I also highlight my ongoing publication and dissemination plan in Section 8.1.1. Section 8.2 proposes future research directions, Section 8.3 highlights key challenges and lessons learnt, and Section 8.4 concludes with final reflections.

8.1 Direct and Broader Impact of Contributions

This thesis has presented studies and models which have advanced our understanding and methodologies for dealing with the complexities of healthcare time-series through several key contributions in the domain of neural architectures. Beginning with a systematic evaluation of current approaches, Chapter 4's experimental evaluation revealed critical gaps in how neural imputation models operating on multivariate EHR time-series are evaluated, particularly highlighting the misalignment between evaluation frameworks and real-world missingness patterns. This foundational work informed the development of three novel architectures: **1) CSAI** aims to overcome some of the issues identified to better impute

EHR time-series. It does so by providing better alignment between the model's architecture and domain requirements through a novel temporal decay mechanisms. CSAI also captures multi-range temporal dynamics pervasive in EHR time series by integrating transformer attention with RNN short-term representation. **2) DEARI** addresses missingness in complex patient trajectories and promises the ability to handle larger and more complex datasets through a scalable deep architecture and uncertainty quantification. Finally, **3) METHOD** takes an orthogonal approach, bypassing the issue of imputation using a large language model. METHOD leverages transformer-based innovations to model patient timelines efficiently, incorporating patient-aware attention and multi-scale temporal processing to improve health outcome discovery.

The practical impact of this work lies in its potential to improve the handling of missing data in EHRs, which is a critical step in enabling more reliable analysis of healthcare datasets. By developing models that are better aligned with the complexities of clinical data—such as irregular sampling, heterogeneous variables, complex latent correlations, and structured missingness—this research provides tools that can support more robust data preprocessing pipelines. While the direct application of these models in clinical applications requires further validation, they represent a step forward in addressing the challenges of missing data in healthcare analytics.

The open-source nature of the developed tools ensures that researchers and practitioners can readily use, adapt, and build upon these models. My hope is by making the code and methodologies publicly available, my work contributes to the growing ecosystem of tools for healthcare machine learning, enabling more reproducible and transparent research in this area.

8.1.1 Strategic Approach to Research Dissemination

While the research has produced significant methodological advances, *I made a deliberate choice to prioritise code quality and reproducibility over immediate publication*. This

decision stemmed from my personal understanding that the practical impact of methodological innovations in healthcare analytics heavily depends on their accessibility and usability by the broader research community. CSAI has now been fully integrated into the `PyPOTS` library after rigorous code verification, API alignment, and adherence to PyPOTS' standards. Currently under review at the IEEE Journal of Biomedical and Health Informatics, my submission of CSAI to peer-review for publication was delayed until its `PyPOTS` implementation was complete and thoroughly validated. DEARI is currently in the final stages of `PyPOTS` verification, with its submission for peer review pending the completion of this integration process. This approach, while extending the timeline to publication, ensures that researchers engage with my work by not only having access to the theoretical foundations and code (available in my `GitHUb` repository), but also to robust, well-documented implementations that can be immediately applied to their own research problems via a specialised Python package. Below is a list of my dissemination plan:

8.1.2 Publications Under Review/In Preparation/Published and Model Availability

1. **How Deep is your Guess? A Fresh Perspective on Deep Learning for Medical Time-Series Imputation.**

    - **Description**: a critical review of the literature of deep imputation models. The paper presents the taxonomy I propose in Chapter 3, the experimental validation I presnt in Chapter 4, using those to highlight the current gaps in neural network imputation for multivariate EHR time-series, pointing out valuable future research directions.
    - **Status**: Submitted to IEEE Journal of Biomedical and Health Informatics (IEEE JBHI - Impact Factor: 7.1). Initial submission date: 29th of August, 2024; Revised on the 10th of January, 2025.

- **Availability:** Preprint available on arXiv (Arxiv ID: 2407.08442).
  - Link: https://arxiv.org/abs/2407.08442

2. **Knowledge Enhanced Conditional Imputation for Healthcare Time-series: The CSAI Model**

   - **Description**: Presents the CSAI architecture detailed in Chapter 5, introducing a novel approach to healthcare time-series imputation through domain-informed temporal decay mechanisms and transformer-enhanced initialisation.
   - **Status**: Submitted to IEEE Journal of Biomedical and Health Informatics (IEEE JBHI - Impact Factor: 7.1). Submission date: $4th$ of November, 2024.
   - **Availability**: Preprint available on arXiv (Arxiv ID: 2312.16713).
     - Link: https://arxiv.org/abs/2312.16713
     - Implementation repository: https://github.com/LinglongQian/CSAI
     - Package availability: available on `PyPOTS` - https://github.com/WenjieDu/PyPOTS (Please scroll down to the list of available algorithms.)

3. **Uncertainty-Aware Deep Attention Recurrent Neural Network for Heterogeneous Time Series Imputation: The DEARI Model**

   - **Description**: Introduces DEARI, detailed in Chapter 6, which advances the state-of-the-art in healthcare time-series imputation through a novel deep architecture with integrated uncertainty quantification.
   - **Status**: Manuscript prepared and submission will be made upon completion of the `PyPOTS` integration of DEARI. I intend to submit DEARI to the IEEE Transactions on Neural Networks and Learning Systems (IEEE TNNLS - Impact Factor: 14.255).
   - **Availability**: Preprint available on arXiv (Arxiv ID: 2401.02258).
     - Link: https://arxiv.org/abs/2401.02258

  - Implementation repository: https://github.com/LinglongQian/DEARI
  - Package availability: In final stages of PyPOTS integration

4. **Unveiling the Secrets: How Masking Strategies Shape Time Series Imputation in Healthcare Analytics**

   - **Description**: Presents the benchmarking results detailed in Chapter 4 using `PyPOTS`, critically examining how different masking strategies affect the performance of neural imputation models in healthcare time-series, providing insights into the relationship between evaluation frameworks and real-world missingness patterns, establishing best practices for benchmarking imputation models on healthcare time-series and advocating for open experimental paradigms using packages such as `PyPOTS`.
   - **Status**: To be submitted to the 23rd International Conference on Artificial Intelligence in Medicine (AIME 2025) under the theme: AI for signal processing and time series analysis. Submission deadline: 3*rd* of February, 2025.
   - **Availability**: Preprint available on arXiv (Arxiv ID: 2405.17508).
     - Link: https://arxiv.org/abs/2405.17508

5. **Addressing Class Imbalance in Electronic Health Records Data Imputation**

   - **Description**: Presents preliminary that has led to the development of a new non-uniform masking strategy (not discussed in this thesis). The aim is to maximise usage of the available dataset while avoiding overfitting. Non-uniform masking is currently being implemented as a masking strategy within the `PyPOTS` ecosystem. The work is being taken over by research assistants within the group for further development and validation, which is why I have not included it as part of my thesis.

- **Status**: Published in the Proceedings of the 6th International Workshop on Knowledge Discovery from Healthcare Data co-located with 32nd International Joint Conference on Artificial Intelligence (IJCAI 2023).
- **Availability**: Published in CEUR Workshop Proceedings, Vol. 3479.
  - Link: https://ceur-ws.org/Vol-3479/paper7.pdf

6. **TSI-Bench: A Systematic Benchmarking Framework for Time Series Imputation**

   - **Description**: This paper is a large-scale collaboration with the `PyPOTS` research team, of which I am a member (https://pypots.com/about/). This work transcends healthcare datasets and benchmarks state of the art neural imputers across multiple domains, focusing on how `PyPOTS` can be used to create reproducible benchmarks, introducing systematic approaches to dataset selection, performance metric evaluation, and computational efficiency assessment, and advocating for standardised benchmarking practices in the field by providing practical guidelines for researchers evaluating imputation models.
   - **Status**: We are continuously updating the manuscript. We are currently rewriting the discussion section of the paper to provide reflections on domain-specific model selection, in preparation for submission. We have not yet agreed on a publication venue.
   - **Availability**: Preprint available on arXiv (Arxiv ID: 2406.12747).
     - Link: https://arxiv.org/abs/2406.12747

7. **METHOD: A Modular Efficient Transformer for Health Outcome Discovery**

   - **Description**: Presents METHOD, detailing its innovation as described in Chapter 7 in addition to ongoing work on temporal representation and tokenisation, which has not been submitted as part of the thesis (although this work is partly described in the future research directions in Section 8.2 below).

- **Status**: Ongoing preparation.
- **Availability**: Implementation repository: https://github.com/LinglongQian/METHOD

8.2 Future Research Directions

Building on the contributions and challenges identified in this thesis, several promising directions for future research emerge:

The Structure of EHR Missingness

The taxonomy and experimental results of neural imputation methods reveal an important gap in how missingness is conceptualised and evaluated in EHR imputers. While Table 3.2 shows models claiming to handle different types of missingness (MCAR, MAR, MNAR), the experiments performed in Chapter 4 point to a more fundamental issue: the presence of *structured missingness* [112] in clinical data, where the specific modes of data collection cause missing values to exhibit associations and structural patterns. In EHRs, structural missingness naturally arises from the asynchronous and decision-driven nature of healthcare data collection [171], reflecting the complex interplay between clinical protocols, resource availability, and care delivery patterns. For instance, vital signs were measured at fixed intervals (creating regular gaps), lab tests clustered around clinical events, and observations clustered around shift ends. The patterns of structured missingness in EHRs are also shaped by the uneven data distributions of clinical outcomes. Many severe clinical events are rare [21]. For example, cardiac arrests constitute only 2.3% of ICU admissions [9], making data samples with a cardiac arrest outcome a minority. This shapes distinct missingness structures: these cases trigger intensive monitoring protocols with frequent measurements, yet their rarity means few complete examples of these measurement patterns exist. This creates a systematic relationship between missingness patterns and clinical

severity - the frequency and timing of missing values becomes informative about the underlying patient state.

The evaluation performed in Chapter 4 using different masking patterns (point, temporal, and block) shows that even sophisticated models struggle when missingness patterns become more structured, suggesting that current approaches may be oversimplified. These findings highlight that structured missingness extends beyond Rubin's classification of missingness into MCAR, MAR, and MNAR, highlighting the need for new theoretical foundations that better capture the systematic, informative nature of missing patterns in clinical data collection. Although CSAI represents a step forward to incorporating medical recording patterns into the architecture of an imputation model, it is only a starting point and more work is needed on how to incorporate domain insight into neural architectures for added robustness and domain alignment.

The Challenge of Uncertainty Quantification

While my review of the literature shows several probabilistic frameworks offer inherent uncertainty estimation, these frameworks rely on distribution-specific assumptions that may struggle to capture the heterogeneous nature of medical time series, while their computational complexity limits practical application. More concerning is that highly performing models such as BRITS and SAITS are fundamentally deterministic with no inherent capability to communicate imputation confidence. Although emerging approaches like DEARI [129] and CF-RNN [154] show promise through post-hoc uncertainty estimation, these are not general solutions. The field needs model-agnostic post-hoc uncertainty quantification approaches that can adapt to the diverse characteristics of medical time series while maintaining computational efficiency, particularly given the demonstrated importance of uncertainty measures for clinical trust [53].

The Gap Between Models and Domain Knowledge

My journey thus far uncovered a disconnect between computational sophistication and clinical expertise. Most models present in the literature (those working on the direct time-series or regarding them as text) treat medical time series as abstract mathematical constructs, overlooking the rich contextual knowledge embedded in clinical practices. This gap becomes particularly problematic when handling complex distributions and rare events common in medical data, where domain knowledge could help ensure clinically meaningful imputations and downstream results. The challenge extends beyond simple rule integration — models need to model the clinical significance of temporal patterns, ensure physiologically plausible reconstruction and generation, and maintain data integrity across diverse patient populations. Future research must develop systematic approaches to bridge this gap, ensuring imputed values are not just statistically sound but clinically meaningful.

Hybrid and Novel Generative Models for Patient Trajectories

The promising results of METHOD in handling patient trajectories using transformer architectures build on existing indications of new avenues leveraging LLMs for healthcare time-series analysis. This research field can take several equally interesting directions.

A current bottleneck in healthcare LLMs lies in the representation of temporal constraints that can significantly impact model generations. While current models demonstrate promising capabilities in handling patient trajectories, they often treat medical events as simple sequential tokens, overlooking the rich temporal and semantic structures inherent in healthcare data. Future work could explore the integration of temporal representation and temporal logic into generative architectures to enable the development of healthcare-specific tokenisation strategies that preserve temporal relationships while capturing clinically meaningful patterns. For instance, rather than treating medical events as discrete tokens in a sequence, models could leverage temporal abstraction to transform raw time-

series into semantically meaningful tokens of clinically meaningful intervals (e.g., "High Heart Rate", "Normal Blood Pressure", "Increasing Creatinine Levels"), enabling more sophisticated reasoning about temporal relationships like "before", "during", or "overlaps". Such form of representation can also provide better means for model evaluation, by offering clinically-aligned natural language descriptors of patient trajectories. Following prior work within the team [68], this is the path I have chosen for my ongoing research, using abstraction and topology to enforce and explicitly represent temporal and domain constraints as tokens. This approach would better align with how clinicians interpret patient trajectories and could lead to more interpretable and clinically relevant predictions.

Moreover, the actual architectures must explicitly account for the complex temporal relationships that often transcend simple sequential ordering. Future models can develop specialised attention mechanisms that can leverage this explicit temporal information while respecting clinical dependencies and causality constraints. This development must be balanced with practical considerations of privacy, data integrity, and computational efficiency.

Finally, the field needs robust evaluation frameworks that not only assess statistical accuracy but also clinical validity, temporal consistency, and medical plausibility. Research on evaluation methodologies for healthcare LLMs remains in its early stages, with recent work [159] highlighting the need for standardised frameworks that incorporate both temporal and clinical validity metrics.

8.3 Key Challenges, Limitations and Lessons Learned

Throughout this research, several challenges emerged that shaped the development and evaluation of the proposed models.

8.3.1 Clinical Data Quality

A major challenge in this work was the heterogeneous nature of the clinical data used. EHR data across the MIMIC and eICU datasets exhibited significant variability, both in terms of the presenting clinical phenotypes as well as the data formats and modalities. While some patient records were relatively sparse and shallow, containing only routine clinical measurements, others had broad and deep data spanning multiple chronic conditions and care episodes. This heterogeneity in data quantity and quality across patients introduced complexities in modelling and analysis and required careful processing and integration. For the sake of reproducibility and to enable comparison with existing models, I carefully followed well-documented and well-cited benchmarks (detailed in Section 2.8. It is worth noting, however, that both databases used their corresponding benchmarks are Intensive Care Unit (ICU) datasets, making the wider evaluation of my models on non-ICU data part of my ongoing work.

8.3.2 Data Access and Siloing

Another significant challenge encountered was the difficulty in accessing and integrating data from different healthcare institutions. While access to widely used datasets such as MIMIC and eICU was relatively straightforward, obtaining and working with local clinical data—particularly from King's College Hospital (KCH)—proved to be a lengthy and arduous process. Even once access was granted, KCH data was siloed and governed by strict information governance protocols, and all data extraction and harmonisation had to go through the CogStack system [70]. This added another layer of complexity and time constraints. Additionally, the lack of standardised data formats and the variability in data quality across institutions made it difficult to establish reproducible benchmarks. Given that CogStack is highly customisable and optimised for specific healthcare environments, its deployment could vary significantly across different institutions, further complicating direct comparisons. Furthermore, CogStack's focus on real-world utility and seamless

integration into existing hospital IT infrastructures often prioritised practical usability over standardised benchmarking. For these reasons, I opted to focus on publicly available datasets for benchmarking and validation. This allowed for more rigorous and reproducible testing, which was essential for the goals of this thesis. While testing on local datasets would have added value, it would not have provided the same level of comparability with existing work, and remains an area of ongoing research.

8.3.3 Compute Restrictions

The computational resources available for this research played a crucial role in enabling certain experiments and analyses, particularly the GPU computing facility available at King's high-performance computing infrastructure, CREATE. However, working with CREATE has also posed great challenges that at times delayed progress. CREATE HPC houses 52 high-specification A100 GPUs, but has stringent limits and scheduling policies. Using CREATE HPC, training a single-layer model on 2,000 MIMIC-III patients requires 6 GPU hours (CSAI) and 10 hours (DEARI) for a single fold. Expanding DEARI to 8 layers increases training time to >60 hours per fold. CREATE HPC's job queue has a 48-hour active usage limit. I have therefore followed an incremental training strategy for both CSAI and DEARI, which involved smaller MIMIC-III subsets and model check-pointing, causing significant overhead and $\tilde{1}$ month delay (rejoining the job queue) per experiment. Each peer-reviewed proof-of-concept experiment requires >24 5-fold experiments, highlighting model complexity within HPC limitations. Training METHOD took over 2 months, using the same strategy of model check-pointing.

The situation was further exacerbated by a period of downtime in the CREATE infrastructure last year, which delayed ongoing projects. Since October 2024, there have been significant issues with the /scratch storage space, which was unavailable until the 15th of January 2025. CREATE's front-facing pages remain unreachable at the time of submission of this thesis (link: https://docs.er.kcl.ac.uk/). These technical constraints have particularly

hindered my ability to run the experiments that evaluate my temporally-aware tokenisation framework (using abstracted temporal intervals, e.g. High Blood Pressure), despite the development being complete in September 2024.

8.4 Conclusions

This thesis has advanced the field of neural architectures for handling missing data in healthcare time-series through three complementary approaches. First, through CSAI, which demonstrated how incorporating domain knowledge and temporal patterns through adaptive decay mechanisms and transformer-enhanced initialisation can significantly improve imputation accuracy. Second, through DEARI, which introduced a scalable deep architecture with integrated uncertainty quantification, particularly valuable for complex patient trajectories where understanding prediction reliability is crucial. Finally, through METHOD, which adapted a modern transformer specifically for healthcare data, providing a foundation for more effective modelling of patient timelines.

The comprehensive experimental evaluation across multiple healthcare datasets has validated these contributions, with particularly strong performance on complex datasets with high missingness and many variables. For instance, on the MIMIC dataset with 78% missingness and 89 features, both CSAI and DEARI achieved significant improvements over baseline models, highlighting how their respective approaches effectively capture intricate dependencies in complex healthcare data. Furthermore, the integration of these models into the `PyPOTS` library ensures their accessibility to the broader research community, facilitating reproducible research and continued innovation in this field. Finally, METHOD has highly outperformed its predecessor, improving performance on longer sequences, where previous models have struggled, and validating the transferability of models trained on different sequence lengths.

Looking ahead, several promising directions emerge for future research. First, for models working strictly with tabular time-series, there is a need to develop more sophisticated

evaluation frameworks that better capture the structured missingness patterns inherent in EHR data. This includes exploring non-uniform masking strategies and incorporating real-world missingness scenarios into benchmarking protocols. Second, while DEARI introduced uncertainty quantification, further work is needed to develop efficient and model-agnostic uncertainty estimation techniques that can be applied to deterministic architectures. Finally, the success of METHOD in handling patient trajectories suggests promising avenues for leveraging generative models in healthcare, but indicates the need for additional frameworks that can meaningfully model the temporal interplay between data points.

Beyond the technical contributions, this work has broader implications for healthcare analytics. By developing models that better align with the complexities of clinical data—such as irregular sampling, heterogeneous variables, and structured missingness—this research provides tools that can support more reliable analysis of healthcare datasets. While the direct application of these models in clinical decision-making requires further validation, they represent a significant step forward in addressing the challenges of missing data in healthcare analytics.

In conclusion, this thesis has demonstrated that through careful consideration of domain characteristics, architectural innovation, and rigorous evaluation, it is possible to develop neural network models that effectively handle the complexities of healthcare time-series data. The open-source nature of the developed tools ensures that researchers can readily use, adapt, and build upon these models, fostering collaboration and iterative improvement in this critical area of healthcare machine learning.

References

[1] Blythe Adamson et al. “Approach to machine learning for extraction of real-world data variables from electronic health records”. In: *Frontiers in Pharmacology* (2023). DOI: 10.3389/fphar.2023.1180962.

[2] Joshua Ainslie et al. “GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints”. In: *The 2023 Conference on Empirical Methods in Natural Language Processing*. 2023.

[3] Razvan Pascanu et al. “How to Construct Deep Recurrent Neural Networks”. In: *ICLR 2014*. 2014.

[4] David J Albers et al. “Mechanistic machine learning: how data assimilation leverages physiologic knowledge using Bayesian inference to forecast the future, infer the present, and phenotype”. In: *Journal of the American Medical Informatics Association* 25.10 (2018), pp. 1392–1401.

[5] Juan Lopez Alcaraz and Nils Strodthoff. “Diffusion-based Time Series Imputation and Forecasting with Structured State Space Models”. In: *Transactions on Machine Learning Research* (2022).

[6] Ali Amirahmadi, Mattias Ohlsson, and Kobra Etminani. “Deep learning prediction models based on EHR trajectories: A systematic review”. In: *Journal of Biomedical Informatics* 144 (2023), p. 104430.

[7] Abdul Fatir Ansari et al. “Chronos: Learning the Language of Time Series”. In: *Transactions on Machine Learning Research* (2024). ISSN: 2835-8856.

[8] Silvia Arber et al. “MLP-deficient mice exhibit a disruption of cardiac cytoarchitectural organization, dilated cardiomyopathy, and heart failure”. In: *Cell* 88.3 (1997), pp. 393–403.

[9] Richard Armstrong and et al. “The incidence of cardiac arrest in the intensive care unit: A systematic review and meta-analysis”. In: *Journal of the Intensive Care Society* 20.2 (2019), pp. 144–154.

[10] Bert Arnrich et al. “Medical Event Data Standard (MEDS): Facilitating Machine Learning for Health”. In: *ICLR 2024 Workshop on Learning from Time Series For Health*. 2024.

[11] Daniel Barrejón, Pablo M Olmos, and Antonio Artés-Rodríguez. “Medical data wrangling with sequential variational autoencoders”. In: *IEEE Journal of Biomedical and Health Informatics* 26.6 (2021), pp. 2737–2745.

[12] Iz Beltagy, Matthew E Peters, and Arman Cohan. “Longformer: The long-document transformer”. In: *arXiv preprint arXiv:2004.05150* (2020).

[13] Charles Blundell et al. “Weight uncertainty in neural network”. In: *International conference on machine learning*. PMLR. 2015, pp. 1613–1622.

[14] A Bornet et al. “Comparing neural language models for medical concept representation and patient trajectory prediction”. In: *medRxiv* (2023). DOI: 10.1101/2023.06.01.23290824.

[15] Aaron Boussina et al. “Impact of a deep learning sepsis prediction model on quality of care and survival”. In: *npj Digital Medicine* 7 (2024), p. 14.

[16] George EP Box et al. *Time series analysis: forecasting and control*. 5th. John Wiley & Sons, 2015.

[17] Tom Brown et al. “Language models are few-shot learners”. In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[18] Defu Cao et al. “TEMPO: Prompt-based Generative Pre-trained Transformer for Time Series Forecasting”. In: *The Twelfth International Conference on Learning Representations*. 2024.

[19] P. Cao et al. “Large language models to facilitate pregnancy prediction after in vitro fertilization”. In: *Acta Obstetricia et Gynecologica Scandinavica* (2024). DOI: 10.1111/aogs.14989.

[20] Wei Cao et al. “BRITS: Bidirectional recurrent imputation for time series”. In: *Advances in Neural Information Processing Systems* 31 (2018).

[21] Abigail R Cartus et al. “Outcome class imbalance and rare events: An underappreciated complication for overdose risk prediction modeling”. In: *Addiction* 118.6 (2023), pp. 1167–1176.

[22] Zhengping Che et al. “Recurrent neural networks for multivariate time series with missing values”. In: *Scientific reports* 8.1 (2018), pp. 1–12.

[23] Si-An Chen et al. “TSMixer: An All-MLP Architecture for Time Series Forecast-ing”. In: *Transactions on Machine Learning Research* (2023). ISSN: 2835-8856.

[24] Liang Chen et al. *Next Token Prediction Towards Multimodal Intelligence: A Comprehensive Survey*. 2024. arXiv: 2412.18619 [cs.CL].

[25] Ricky TQ Chen et al. “Neural ordinary differential equations”. In: *Advances in neural information processing systems* 31 (2018).

[26] Yu Chen et al. “Provably convergent Schrödinger bridge with applications to probabilistic time series imputation”. In: *International Conference on Machine Learning*. PMLR. 2023, pp. 4485–4513.

[27] Rewon Child et al. “Generating long sequences with sparse transformers”. In: *arXiv preprint arXiv:1904.10509* (2019).

[28] Edward Choi, Mohammad Taha Bahadori, and Jimeng Sun. “Learning low-dimensional representations of medical concepts”. In: *AMIA Summits on Translational Science Proceedings* 2020 (2020), p. 41.

[29] Tae-Min Choi, Ji-Su Kang, and Jong-Hwan Kim. “RDIS: Random drop imputation with self-training for incomplete time series data”. In: *IEEE Access* (2023).

[30] Krzysztof Marcin Choromanski et al. “Rethinking Attention with Performers”. In: *International Conference on Learning Representations*. 2021.

[31] Andrea Cini, Ivan Marisca, and Cesare Alippi. “Filling the G_ap_s: Multivariate Time Series Imputation by Graph Neural Networks”. In: *ICLR*. 2022.

[32] Antonia Creswell et al. “Generative adversarial networks: An overview”. In: *IEEE signal processing magazine* 35.1 (2018), pp. 53–65.

[33] Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. “A survey of multilingual neural machine translation”. In: *ACM Computing Surveys (CSUR)* 53.5 (2020), pp. 1–38.

[34] Tri Dao et al. “Flashattention: Fast and memory-efficient exact attention with io-awareness”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 16344–16359.

[35] Adnan Darwiche. “Human-level intelligence or animal-like abilities?” In: *Communications of the ACM* 61.10 (2018), pp. 56–67.

[36] Ritankar Das and David J. Wales. “Machine learning landscapes and predictions for patient outcomes”. In: *Royal Society Open Science* 4 (2017), p. 170175.

[37] A Philip Dawid, Mervyn Stone, and James V Zidek. “Marginalization paradoxes in Bayesian and structural inference”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 35.2 (1973), pp. 189–213.

[38] Min Deng et al. “A Hybrid Method for Interpolating Missing Data in Heterogeneous Spatio-Temporal Datasets”. In: *ISPRS International Journal of Geo-Information* 5.2 (2016).

[39] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: 2018. URL: https://arxiv.org/abs/1810.04805.

[40] Ning Ding et al. “Parameter-efficient fine-tuning of large-scale pre-trained language models”. In: *Nature Machine Intelligence* 5.3 (2023), pp. 220–235.

[41] Li Dong et al. “Unified language model pre-training for natural language understanding and generation”. In: *Advances in neural information processing systems* 32 (2019).

[42] Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. “Attention is not all you need: pure attention loses rank doubly exponentially with depth”. In: *ICML*. 2021, pp. 2793–2803.

[43] William P. T. M. van Doorn et al. “A comparison of machine learning models versus clinical evaluation for mortality prediction in patients with sepsis”. In: *PLoS One* 16.1 (2021), e0245157.

[44] Alexey Dosovitskiy et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *International Conference on Learning Representations*. 2021.

[45] Nan Du et al. “Glam: Efficient scaling of language models with mixture-of-experts”. In: *International Conference on Machine Learning*. PMLR. 2022, pp. 5547–5569.

[46] Shengdong Du et al. “Multivariate time series forecasting via attention-based encoder–decoder framework”. In: *Neurocomputing* 388 (2020), pp. 269–279.

[47] Wenjie Du. *PyPOTS: A Python Toolbox for Data Mining on Partially-Observed Time Series*. 2023. arXiv: 2305.18811.

[48] Wenjie Du. “PyPOTS: A Python Toolbox for Data Mining on Partially-Observed Time Series”. In: *arXiv preprint arXiv:2305.18811* (2023).

[49] Wenjie Du, David Côté, and Yan Liu. "Saits: Self-attention-based imputation for time series". In: *Expert Systems with Applications* 219 (2023), p. 119619.

[50] H. Duan et al. "On Clinical Event Prediction in Patient Treatment Trajectory Using Longitudinal Electronic Health Records". In: *IEEE Journal of Biomedical and Health Informatics* (2019). DOI: 10.1109/JBHI.2019.2962079.

[51] Craig K Enders. *Applied missing data analysis*. New York: Guilford Press, 2010.

[52] Dyke Ferber et al. "Autonomous Artificial Intelligence Agents for Clinical Decision Making in Oncology". In: *arXiv preprint* (2024). DOI: 10.48550/arXiv.2404.04667.

[53] Vincent Fortuin et al. "Gp-vae: Deep probabilistic time series imputation". In: *International conference on artificial intelligence and statistics*. PMLR. 2020, pp. 1651–1661.

[54] Pedro J García-Laencina, José-Luis Sancho-Gómez, and Aníbal R Figueiras-Vidal. "Pattern classification with missing data: a review". In: *Neural Computing and Applications* 19 (2010), pp. 263–282.

[55] David Gordon et al. "TSI-GNN: Extending graph neural networks to handle missing data in temporal settings". In: *Frontiers in big Data* 4 (2021), p. 693869.

[56] A Goyal and Bengiom Y. "Inductive biases for deep learning of higher-level cognition". In: *Proceedings of the Royal Society* 478 (2022), p. 20210068.

[57] Audrunas Gruslys et al. "Memory-efficient backpropagation through time". In: *Advances in neural information processing systems* 29 (2016).

[58] Jiuxiang Gu et al. "Recent advances in convolutional neural networks". In: *Pattern recognition* 77 (2018), pp. 354–377.

[59] Hrayr Harutyunyan et al. "Multitask learning and benchmarking with clinical time series data". In: *Scientific data* 6.1 (2019), pp. 1–18.

[60] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778.

[61] James J Heckman. "Sample selection bias as a specification error". In: *Econometrica: Journal of the econometric society* (1979), pp. 153–161.

[62] L Heumos et al. "Exploratory electronic health record analysis with ehrapy". In: *medRxiv* (2023). DOI: 10.1101/2023.12.11.23299816.

[63] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models". In: *Advances in neural information processing systems* 33 (2020), pp. 6840–6851.

[64] Ronald A Howard. "Dynamic programming and markov processes." In: (1960).

[65] W. Hsu, J. Warren, and Patricia J. Riddle. "Multivariate Sequential Analytics for Treatment Trajectory Forecasting". In: *Australasian Computer Science Week*. 2019. DOI: 10.1145/3290688.3290724.

[66] Kaiyu Huang et al. "A Survey on Large Language Models with Multilingualism: Recent Advances and New Frontiers". In: *arXiv preprint arXiv:2405.10936* (2024).

[67] Marine Humbert-Droz, Pritam Mukherjee, and Olivier Gevaert. "Strategies to Address the Lack of Labeled Data for Supervised Machine Learning Training With Electronic Health Records: Case Study for the Extraction of Symptoms From Clinical Notes". In: *JMIR Medical Informatics* 10.3 (Mar. 2022), e32903.

[68] Zina M. Ibrahim, Honghan Wu, and Richard J. B. Dobson. "Modeling Rare Interactions in Time Series Data Through Qualitative Change: Application to Outcome Prediction in Intensive Care Units". In: *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)*. Vol. 325. Frontiers in Artificial Intelligence and Applications. 2020, pp. 1826–1833.

[69] Md Mohaimenul Islam et al. "Prediction of sepsis patients using machine learning approach: a meta-analysis". In: *Computer methods and programs in biomedicine* 170 (2019), pp. 1–9.

[70] Richard Jackson, Ismail Kartoglu, Clive Stringer, et al. "CogStack - experiences of deploying integrated information retrieval and extraction services in a large National Health Service Foundation Trust hospital". In: *BMC Medical Informatics and Decision Making* 18.1 (2018), p. 47. DOI: 10.1186/s12911-018-0623-9. URL: https://doi.org/10.1186/s12911-018-0623-9.

[71] Peter B Jensen, Lars J Jensen, and Søren Brunak. "Mining electronic health records: towards better research applications and clinical care". In: *Nature Reviews Genetics* 13.6 (2012), pp. 395–405.

[72] Shaoxiong Ji et al. "A Unified Review of Deep Learning for Automated Medical Coding". In: *ACM Computing Surveys* (2024). DOI: 10.1145/3664615.

[73] Ming Jin et al. "A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

[74] Ming Jin et al. "Large models for time series and spatio-temporal data: A survey and outlook". In: *arXiv preprint arXiv:2310.10196* (2023).

[75] Alistair Johnson and et al. "MIMIC-III, a freely accessible critical care database". In: *Scientific data* 3.1 (2016), pp. 1–9.

[76] Alistair EW Johnson et al. "MIMIC-IV, a freely accessible electronic health record dataset". In: *Scientific data* 10.1 (2023), p. 1.

[77] Mandar Joshi et al. "Spanbert: Improving pre-training by representing and predicting spans". In: *Transactions of the association for computational linguistics* 8 (2020), pp. 64–77.

[78] Eunji Jun et al. "Uncertainty-gated stochastic sequential model for ehr mortality prediction". In: *IEEE Transactions on Neural Networks and Learning Systems* 32.9 (2020), pp. 4052–4062.

[79] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *Proceedings of naacL-HLT*. Vol. 1. Minneapolis, Minnesota. 2019, p. 2.

[80] Mourad Khayati et al. "Mind the gap: an experimental evaluation of imputation of missing values techniques in time series". In: *Proc. VLDB Endow.* 13.5 (Jan. 2020), pp. 768–782. ISSN: 2150-8097.

[81] SeungHyun Kim et al. “Probabilistic imputation for time-series classification with missing data”. In: *International Conference on Machine Learning*. PMLR. 2023, pp. 16654–16667.

[82] Taku Kudo and John Richardson. “SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 66–71.

[83] Srivatsan Laxman and P Shanti Sastry. “A survey of temporal data mining”. In: *Sadhana* 31 (2006), pp. 173–198.

[84] Dongha Lee, Sookyung Yu, and Hwanjo Yu. “Temporal self-attention network for medical temporal sequence prediction”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.11 (2020), pp. 4048–4060.

[85] Youngjae Lee et al. “Deep Learning in the Medical Domain: Predicting Cardiac Arrest Using Deep Learning”. In: *Acute and Critical Care* 33.3 (2018), pp. 117–120.

[86] Patrick Lewis et al. “Retrieval-augmented generation for knowledge-intensive nlp tasks”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 9459–9474.

[87] Jiaming Li et al. “Imputation of missing values for electronic health record laboratory data”. In: *npj Digital Medicine* 4.1 (2021), p. 147.

[88] Steven Cheng-Xian Li and Benjamin M Marlin. “Learning from irregularly-sampled time series: A missing data perspective”. In: *International Conference on Machine Learning* (2020), pp. 5937–5946.

[89] Yikuan Li et al. “BEHRT: transformer for electronic health records”. In: *Scientific reports* 10.1 (2020), p. 7155.

[90] Yikuan Li et al. “Hi-BEHRT: hierarchical transformer-based model for accurate prediction of clinical events using multimodal longitudinal electronic health records”. In: *IEEE journal of biomedical and health informatics* 27.2 (2022), pp. 1106–1117.

[91] Yuebing Liang, Zhan Zhao, and Lijun Sun. “Memory-augmented dynamic graph convolution networks for traffic data imputation with diverse missing patterns”. In: *Transportation Research Part C: Emerging Technologies* 143 (2022), p. 103826.

[92] Jessica Lin et al. “Experiencing SAX: a novel symbolic representation of time series”. In: *Data Mining and knowledge discovery* 15 (2007), pp. 107–144.

[93] Tianyang Lin et al. “A survey of transformers”. In: *AI open* 3 (2022), pp. 111–132.

[94] Tsung-Yi Lin et al. “Feature pyramid networks for object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2117–2125.

[95] Zachary C Lipton, David Kale, and Randall Wetzel. “Directly modeling missing data in sequences with rnns: Improved classification of clinical time series”. In: *Machine learning for healthcare conference*. PMLR. 2016, pp. 253–270.

[96] Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*. Vol. 793. John Wiley & Sons, 2019.

[97] Yukai Liu et al. “Naomi: Non-autoregressive multiresolution sequence imputation”. In: *Advances in neural information processing systems* 32 (2019).

[98] Yuxi Liu et al. “Compound density networks for risk prediction using electronic health records”. In: *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE. 2022, pp. 1078–1085.

[99] Yonghong Luo et al. “E2gan: End-to-end generative adversarial network for multivariate time series imputation”. In: *Proceedings of the 28th international joint conference on artificial intelligence*. AAAI Press Palo Alto, CA, USA. 2019, pp. 3094–3100.

[100] Yonghong Luo et al. “Multivariate time series imputation with generative adversarial networks”. In: *Advances in neural information processing systems* 31 (2018).

[101] Yuan Luo. “Evaluating the state of the art in missing data imputation for clinical data”. In: *Briefings in Bioinformatics* 23.1 (2022), bbab489.

[102] David JC MacKay. “A practical Bayesian framework for backpropagation networks”. In: *Neural computation* 4.3 (1992), pp. 448–472.

[103] David JC MacKay et al. “Introduction to Gaussian processes”. In: *NATO ASI series F computer and systems sciences* 168 (1998), pp. 133–166.

[104] Pierre-Alexandre Mattei and Jes Frellsen. “MIWAE: Deep generative modelling and imputation of incomplete data sets”. In: *International conference on machine learning*. PMLR. 2019, pp. 4413–4423.

[105] Matthew McDermott et al. “Event Stream GPT: a data pre-processing and modeling library for generative, pre-trained transformers over continuous-time sequences of complex events”. In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 24322–24334.

[106] Matthew B. A. McDermott et al. “Reproducibility in machine learning for health research: Still a ways to go”. In: *Science Translational Medicine* 13.586 (2021), eabb1655.

[107] Larry R Medsker and LC Jain. “Recurrent neural networks”. In: *Design and Applications* 5.64-67 (2001), p. 2.

[108] Gaya Mehenni and Amal Zouaq. “Ontology-Constrained Generation of Domain-Specific Clinical Summaries”. In: *International Conference Knowledge Engineering and Knowledge Management*. 2024. DOI: 10.48550/arXiv.2411.15666.

[109] Xiaoye Miao et al. “Generative semi-supervised learning for multivariate time series imputation”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 35. 10. 2021, pp. 8983–8991.

[110] John A Miller et al. “A survey of deep learning and foundation models for time series forecasting”. In: *arXiv preprint arXiv:2401.13912* (2024).

[111] Riccardo Miotto et al. “Deep patient: an unsupervised representation to predict the future of patients from the electronic health records”. In: *Scientific reports* 6.1 (2016), p. 26094.

[112] Robin Mitra et al. “Learning from data with structured missingness”. In: *Nature Machine Intelligence* 5.1 (2023), pp. 13–23.

[113] Robert Moskovitch and Yuval Shahar. "Medical temporal-knowledge discovery via temporal abstraction". In: *AMIA Annual Symposium Proceedings* 2009 (2009), pp. 452–456.

[114] Travis J Moss et al. "Continuous vital sign analysis for predicting and preventing noncardiac complications after major surgery". In: *American Journal of Physiology-Heart and Circulatory Physiology* 312.4 (2017), H627–H636.

[115] Ahmad Wisnu Mulyadi, Eunji Jun, and Heung-Il Suk. "Uncertainty-aware variational-recurrent imputation network for clinical time series". In: *IEEE Transactions on Cybernetics* 52.9 (2021), pp. 9684–9694.

[116] Alfredo Nazabal et al. "Handling incomplete heterogeneous data using vaes". In: *Pattern Recognition* 107 (2020), p. 107501.

[117] Radford M Neal. *Bayesian learning for neural networks*. Vol. 118. Springer Science & Business Media, 2012.

[118] Yuqi Nie et al. "A Time Series is Worth 64 Words: Long-term Forecasting with Transformers". In: *The Eleventh International Conference on Learning Representations*. 2023.

[119] Philipp Oberdiek, Gernot Fink, and Matthias Rottmann. "Uqgan: A unified model for uncertainty quantification of deep classifiers trained via conditional gans". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 21371–21385.

[120] Mahmud Omar et al. "Applications of Large Language Models in Psychiatry: A Systematic Review". In: *medRxiv* (2024). DOI: 10.1101/2024.03.28.24305027.

[121] Chao Pang et al. "CEHR-BERT: Incorporating temporal information from structured EHR data to improve prediction tasks". In: *Machine Learning for Health*. PMLR. 2021, pp. 239–260.

[122] Sung Woo Park, Kyungjae Lee, and Junseok Kwon. "Neural markov controlled SDE: Stochastic optimization for continuous-time data". In: *International Conference on Learning Representations*. 2021.

[123] Soumen Kumar Pati et al. "Missing value estimation of microarray data using Sim-GAN". In: *Knowledge and Information Systems* 64.10 (2022), pp. 2661–2687.

[124] Rimma Pivovarov et al. "Identifying and mitigating biases in EHR laboratory tests". In: *Journal of biomedical informatics* 51 (2014), pp. 24–34.

[125] Rimma Pivovarov et al. "Temporal trends of hemoglobin A1c testing". In: *Journal of the American Medical Informatics Association* 21.6 (2014), pp. 1038–1044.

[126] Tom J Pollard et al. "The eICU Collaborative Research Database, a freely available multi-center database for critical care research". In: *Scientific data* 5.1 (2018), pp. 1–13.

[127] Ofir Press, Noah Smith, and Mike Lewis. "Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation". In: *International Conference on Learning Representations*. 2022.

[128] Sanjay Purushotham et al. "Benchmarking deep learning models on large healthcare datasets". In: *Journal of biomedical informatics* 83 (2018), pp. 112–134.

[129] Linglong Qian, Zina Ibrahim, and Richard Dobson. "Uncertainty-Aware Deep Attention Recurrent Neural Network for Heterogeneous Time Series Imputation". In: *arXiv preprint arXiv:2401.02258* (2024).

[130] Linglong Qian et al. "Addressing Class Imbalance in Electronic Health Records Data Imputation". In: *Proceedings of the 6th International Workshop on Knowledge Discovery from Healthcare Data co-located with 32nd International Joint Conference on Artificial Intelligence (IJCAI 2023)*. Vol. 3479. CEUR Workshop Proceedings. CEUR-WS.org, 2023.

[131] Linglong Qian et al. "Unveiling the Secrets: How Masking Strategies Shape Time Series Imputation". In: *arXiv preprint arXiv:2405.17508* (2024).

[132] Colin Raffel et al. "Exploring the limits of transfer learning with a unified text-to-text transformer". In: *Journal of machine learning research* 21.140 (2020), pp. 1–67.

[133] Laila Rasmy et al. "Med-BERT: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction". In: *NPJ digital medicine* 4.1 (2021), p. 86.

[134] Franois Remy, Kris Demuynck, and Thomas Demeester. "BioLORD-2023: semantic textual representations fusing large language models and clinical knowledge graph insights". In: *Oxford University Press* (2024). DOI: 10.1093/jamia/ocae029.

[135] Pawel Renc et al. "Zero shot health trajectory prediction using transformer". In: *NPJ Digital Medicine* 7.1 (2024), p. 256.

[136] Pieter Robberechts, Wannes Meert, and Jesse Davis. "Elastic Product Quantization for Time Series". In: *International Conference on Discovery Science*. Springer. 2022, pp. 157–172.

[137] David Rodbard. "Continuous glucose monitoring: a review of successes, challenges, and opportunities". In: *Diabetes Technology & Therapeutics* 18.S2 (2016), S2–3.

[138] Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. "Latent ordinary differential equations for irregularly-sampled time series". In: *Advances in neural information processing systems* 32 (2019).

[139] Donald B Rubin. "Inference and missing data". In: *Biometrika* 63.3 (1976), pp. 581–592.

[140] Mohammed Saeed et al. "Multiparameter Intelligent Monitoring in Intensive Care II: a public-access intensive care unit database". In: *Critical care medicine* 39.5 (2011), pp. 952–960.

[141] Germans Savcisens et al. "Using sequences of life-events to predict human lives". In: *Nature Computational Science* 4.1 (2024), pp. 43–56.

[142] JL Schafer. *Analysis of incomplete multivariate data*. CRC press, 1997.

[143] Joseph L Schafer and John W Graham. "Missing data: our view of the state of the art". In: *Psychological Methods* 7.2 (2002), pp. 147–177.

[144] Mona Schirmer et al. "Modeling irregular time series with continuous recurrent units". In: *International Conference on Machine Learning*. PMLR. 2022, pp. 19388–19405.

[145] Mike Schuster and Kuldip K Paliwal. "Bidirectional recurrent neural networks". In: *IEEE transactions on Signal Processing* 45.11 (1997), pp. 2673–2681.

[146] Fatemeh Shah-Mohammadi and Joseph Finkelstein. "Addressing Semantic Variability in Clinical Outcome Reporting Using Large Language Models". In: *BioMedInformatics* (2024). DOI: 10.3390/biomedinformatics4040116.

[147] Chao Shang et al. "VIGAN: Missing view imputation with generative adversarial networks". In: *2017 IEEE International conference on big data (Big Data)*. IEEE. 2017, pp. 766–775.

[148] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. "Self-attention with relative position representations". In: *arXiv preprint arXiv:1803.02155* (2018).

[149] Seyedmostafa Sheikhalishahi, Vevake Balaraman, and Venet Osmani. "Benchmarking machine learning models on multi-centre eICU critical care dataset". In: *Plos one* 15.7 (2020), e0235424.

[150] Lifeng Shen and James Kwok. "Non-autoregressive conditional diffusion models for time series prediction". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 31016–31029.

[151] Yuqi Si et al. "Deep Representation Learning of Patient Data from Electronic Health Records (EHR): A Systematic Review". In: *Journal of biomedical informatics* (2020), p. 103671.

[152] Ikaro Silva et al. "Predicting in-hospital mortality of icu patients: The physionet/computing in cardiology challenge 2012". In: *2012 Computing in Cardiology*. IEEE. 2012, pp. 245–248.

[153] Yang Song et al. "Score-Based Generative Modeling through Stochastic Differential Equations". In: *International Conference on Learning Representations*. 2020.

[154] Kamile Stankeviciute, Ahmed M Alaa, and Mihaela van der Schaar. "Conformal time-series forecasting". In: *Advances in neural information processing systems* 34 (2021), pp. 6216–6228.

[155] Ethan Steinberg et al. "Language models are an effective representation learning technique for electronic health record data". In: *Journal of biomedical informatics* 113 (2021), p. 103637.

[156] Jianlin Su et al. "Roformer: Enhanced transformer with rotary position embedding". In: *Neurocomputing* 568 (2024), p. 127063.

[157] Qiuling Suo et al. "GLIMA: Global and local time series imputation with multi-directional attention learning". In: *2020 IEEE International Conference on Big Data (Big Data)*. IEEE. 2020, pp. 798–807.

[158] Sabera J Talukder, Yisong Yue, and Georgia Gkioxari. "TOTEM: TOkenized Time Series EMbeddings for General Time Series Analysis". In: *Transactions on Machine Learning Research* (2024). ISSN: 2835-8856.

[159] Thomas Yu Chow Tam et al. "A framework for human evaluation of large language models in healthcare derived from literature review". In: *Nature Digital Medicine* 7 (2024), p. 258.

[160] Wensi Tang et al. "Rethinking 1d-cnn for time series classification: A stronger baseline". In: *arXiv preprint arXiv:2002.10061* (2020), pp. 1–7.

[161] Yusuke Tashiro et al. “CSDI: Conditional score-based diffusion models for probabilistic time series imputation”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 24804–24816.

[162] Yi Tay et al. “Efficient Transformers: A Survey”. In: *ACM Comput. Surv.* 55.6 (2022). ISSN: 0360-0300.

[163] Kristian Thygesen et al. “Fourth universal definition of myocardial infarction (2018)”. In: *European Heart Journal* 40.3 (2019), pp. 237–269.

[164] Harriet Loice Tsinale, Samuel Mbugua, and Anthony Luvanda. “Architectural Health Data Standards and Semantic Interoperability: A Comprehensive Review”. In: *International Journal of Engineering Applied Sciences and Technology* (2023). DOI: 10.33564/ijeast.2023.v08i04.002.

[165] Mehmet Ozgur Turkoglu et al. “Gating revisited: Deep multi-layer RNNs that can be trained”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.8 (2021), pp. 4081–4092.

[166] F-Y Tzeng and K-L Ma. *Opening the black box-data driven visualization of neural networks*. IEEE, 2005.

[167] Z Unger et al. “Clinical Applications and Limitations of Large Language Models in Nephrology: A Systematic Review”. In: *medRxiv* (2024). DOI: 10.1101/2024.10.30.24316199.

[168] Stef Van Buuren. *Flexible imputation of missing data*. CRC press, 2018.

[169] Nicolaas G Van Kampen. “Stochastic differential equations”. In: *Physics reports* 24.3 (1976), pp. 171–228.

[170] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017).

[171] Brian Wahl et al. “Artificial intelligence (AI) and global health: how can AI contribute to health in resource-poor settings?” In: *BMJ global health* 3.4 (2018).

[172] Amadeo Wals-Zurita et al. “The Transformative Potential of Large Language Models in Mining Electronic Health Records Data”. In: *bioRxiv* (2024). DOI: 10.1101/2024.03.07.24303588.

[173] Zhiguang Wang and Tim Oates. “Imaging time-series to improve classification and imputation”. In: *Proceedings of the 24th International Conference on Artificial Intelligence*. 2015, pp. 3939–3945.

[174] Brian J Wells et al. “Strategies for handling missing data in electronic health record derived data”. In: *Egems* 1.3 (2013).

[175] Haixu Wu et al. “TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis”. In: *The Eleventh International Conference on Learning Representations*. 2023.

[176] Zonghan Wu et al. “Connecting the dots: Multivariate time series forecasting with graph neural networks”. In: *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 2020, pp. 753–763.

[177] Qizhe Xie et al. “Unsupervised data augmentation for consistency training”. In: *Advances in neural information processing systems* 33 (2020), pp. 6256–6268.

[178] Jingwen Xu, Fei Lyu, and Pong C Yuen. "Density-aware temporal attentive step-wise diffusion model for medical time series imputation". In: *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 2023, pp. 2836–2845.

[179] Fan Yang et al. "DeepMPM: a mortality risk prediction model using longitudinal EHR data". In: *BMC Bioinformatics* 23 (2022).

[180] Yongchao Ye, Shiyao Zhang, and James J. Q. Yu. "Spatial-Temporal Traffic Data Imputation via Graph Attention Convolutional Network". In: *Artificial Neural Networks and Machine Learning – ICANN 2021*. Ed. by Igor Farkaš et al. 2021, pp. 241–252.

[181] A Yarkın Yıldız, Emirhan Koç, and Aykut Koç. "Multivariate time series imputation with transformers". In: *IEEE Signal Processing Letters* 29 (2022), pp. 2517–2521.

[182] Jinsung Yoon, William R Zame, and Mihaela van der Schaar. "Estimating missing data in temporal data streams using multi-directional recurrent neural networks". In: *IEEE Transactions on Biomedical Engineering* 66.5 (2018), pp. 1477–1490.

[183] Jinsung Yoon, William R Zame, and Mihaela van der Schaar. "Multi-directional recurrent neural networks: A novel method for estimating missing data". In: *Time series workshop in international conference on machine learning*. 2017.

[184] Manzil Zaheer et al. "Big bird: Transformers for longer sequences". In: *Advances in neural information processing systems* 33 (2020), pp. 17283–17297.

[185] Ailing Zeng et al. "Are transformers effective for time series forecasting?" In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 37. 9. 2023, pp. 11121–11128.

[186] Jiayu Zhang et al. "RETAIN: An interpretable predictive model for healthcare using reverse time attention mechanism". In: *Nature Communications* 10.1 (2019), p. 2618.

[187] Yongshun Zhang et al. "A comprehensive survey on deep learning for missing data imputation: Taxonomy, challenges, and future directions". In: *Information Fusion* 93 (2023), p. 101796.

[188] Huan Zhao et al. "Ophtha-LLaMA2: A Large Language Model for Ophthalmology". In: *arXiv preprint* (2023). DOI: 10.48550/arXiv.2312.04906.

[189] Min Zhao et al. "Deep Learning using 2D Convolutional Neural Networks for Multivariate Clinical Time Series: A Review and Comparison". In: *IEEE Journal of Biomedical and Health Informatics* 27.4 (2023), pp. 1723–1734.

[190] Jie Zhou et al. "Graph neural networks: A review of methods and applications". In: *AI open* 1 (2020), pp. 57–81.

[191] Tian Zhou et al. "One Fits All: Power General Time Series Analysis by Pretrained LM". In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023.

[192] Yizhao Zhou et al. "Missing data matter: an empirical evaluation of the impacts of missing EHR data in comparative effectiveness research". In: *Journal of the American Medical Informatics Association* 30.7 (2023), pp. 1246–1256.

[193] Yinghao Zhu et al. “REALM: RAG-Driven Enhancement of Multimodal Electronic Health Records Analysis via Large Language Models”. In: *arXiv preprint* (2024). DOI: 10.48550/arXiv.2402.07016.

[194] Ningning Zhuang et al. “On Mixture Density Networks: A Survey”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2023).

[195] Amadeo Jesus Wals Zurita et al. “The Transformative Potential of Large Language Models in Mining Electronic Health Records Data: Content Analysis”. In: *JMIR Medical Informatics* (2025). DOI: 10.2196/58457.

[196] Amadeo Wals Zurita et al. “The Transformative Potential of Large Language Models in Mining Electronic Health Records Data”. In: *medRxiv* (2024). DOI: 10.1101/2024.03.07.24303588.

Appendix A

Missing Data Mechanisms

The concept of missingness can be understood through two fundamental aspects: missing mechanisms and missing patterns. Missing mechanisms describe the underlying processes that lead to missing values, while missing patterns define how these missing values manifest within the dataset. The classical framework for discerning missingness mechanisms has been established by Rubin in the 1970s [139]. Prior to this, missing data treatments were largely ad hoc and lacked a formal mathematical foundation. Rubin introduced a systematic framework based on the relationship between the missingness of a given variable and other observed and unobserved variables within the dataset. This framework remains a cornerstone of modern missing data analysis, and has identified three distinct mechanisms through which data can be missing. Those are: Missing Completely at Random (MCAR), Missing at Random (MAR), and Missing Not at Random (MNAR).

Missing Completely at Random (MCAR)

When data are MCAR, the probability of missingness is independent of both the observed and unobserved values in the dataset. Mathematically, if $\mathbf{Y}$ represents the complete data, $\mathbf{Y}_{\text{obs}}$ the observed values, $\mathbf{Y}_{\text{mis}}$ the missing values, and $\mathbf{R}$ the missingness indicator, then:

$$P(\mathbf{R}|\mathbf{Y}) = P(\mathbf{R}) \tag{A.1}$$

This is the strongest assumption about missingness and rarely holds in practice [51]. However, when MCAR holds, complete case analysis (though not necessarily optimal) will produce unbiased estimates, albeit with reduced statistical power [143].

Missing at Random (MAR)

The MAR mechanism, despite its potentially misleading name, indicates that missingness depends only on the observed data values, not on the missing values themselves. Formally:

$$P(\mathbf{R}|\mathbf{Y}) = P(\mathbf{R}|\mathbf{Y}_{\text{obs}}) \tag{A.2}$$

MAR is a less restrictive assumption than MCAR and is often more plausible in real-world scenarios. For example, if older participants in a study are more likely to skip certain questions, but this probability depends only on their age (which is recorded), the missingness would be MAR. Most modern missing data methods, including multiple imputation and maximum likelihood approaches, rely on the MAR assumption [168].

Missing Not at Random (MNAR)

MNAR occurs when the probability of missingness depends on the unobserved values themselves, even after controlling for the observed data. In formal terms:

$$P(\mathbf{R}|\mathbf{Y}) \neq P(\mathbf{R}|\mathbf{Y}_{\text{obs}}) \tag{A.3}$$

This represents the most challenging missingness mechanism, as the missingness process cannot be inferred solely from the observed data. Instead, additional assumptions or external information are required to facilitate valid inference. Selection models [61] and pattern-mixture models [96] have been proposed to address MNAR data. Selection models explicitly model the missingness mechanism, typically through a selection equation, whereas pattern-mixture models stratify the data based on different missingness patterns

and estimate separate distributions for each. Both approaches, however, rely on untestable assumptions regarding the nature of the missingness process.

Despite the clarity the above taxonomy provides, it often falls short in capturing the complexity of missing data scenarios in modern datasets [112]. This limitation has driven the development of new frameworks [112, 31, 80] that better align with diverse real-world data.

Appendix B

Artificial Neural Networks

Artificial Neural Networks (ANNs) represent a fundamental paradigm in machine learning that draws inspiration from biological neural systems. At their core, ANNs consist of interconnected computational units (neuronees) organised in layers, where each neurone processes input signals through mathematical transformations and produces an output activation. These networks learn by adjusting the strength of connections between neurones in response to training data, gradually modifying their internal parameters to capture complex patterns and relationships in the input space. The layered architecture allows ANNs to transform raw input data into increasingly sophisticated representations, enabling them to model intricate non-linear relationships between inputs X and the target outputs that simpler models cannot capture. This hierarchical learning process, combined with their ability to automatically discover relevant features in the data, has made ANNs particularly successful across a wide range of applications, from image recognition to natural language processing.

ANNs are configured with specific connection *architectures* and computational configurations, each designed to capture distinct patterns in data. This appendix reviews fundamental neural architectures and their applications to time series imputation. I begin with feed-forward networks, establishing core concepts and theoretical foundations. I then examine specialised architectures: recurrent neural networks (RNNs) for modelling

temporal dependencies, convolutional neural networks (CNNs) for capturing hierarchical temporal patterns, and graph neural networks (GNNs) for modelling complex variable relationships. The discussion continues with the Transformer architecture and its self-attention mechanism for handling long-range dependencies. I conclude by exploring modern learning frameworks, including diffusion models and mixture density networks, which provide sophisticated approaches to modelling uncertainty in missing data scenarios.

B.1 Feed-Forward Neural Networks

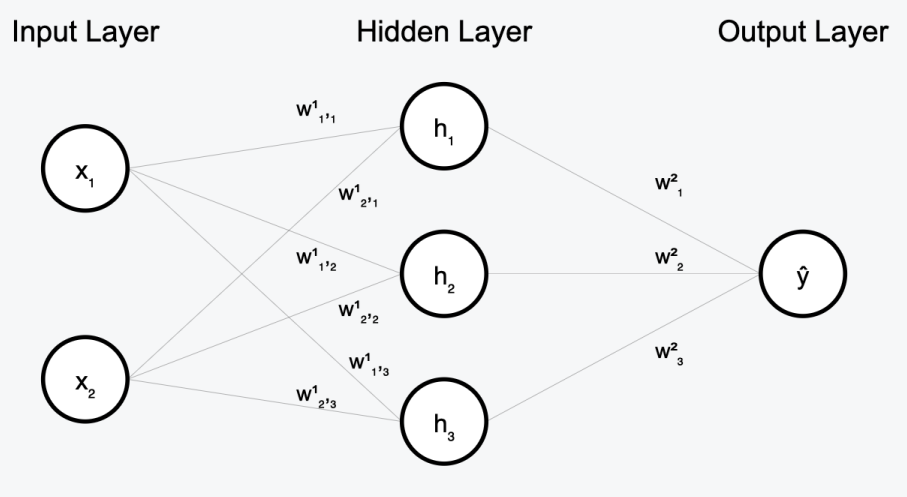


Fig. B.1 An example feed-forward neural network with one output. The network consists of an input layer with two features (x_1, x_2), a hidden layer with three neuronees, and a single output neurone for binary classification. The weights connecting each layer ($w^1_{i,j}$ and w^2_i) represent the learnable parameters of the network.

The Multilayer Perceptron (MLP), while being the simplest neural architecture, serves as a foundation for understanding more complex models. An example MLP is shown in Figure B.1. In time series imputation, MLPs process fixed-width windows of temporal data, where each neurone computes an activation based on its inputs and learnt parameters. The fundamental computation at each neurone is given by:

$$z(x) = f(W \cdot x + b) \tag{B.1}$$

where f represents a non-linear activation function, W is the weight matrix, and b is the bias term. Common activation functions in modern neural networks include:

ReLU:	Leaky ReLU:	GELU:
$f(x) = max(0,x)$	$f(x) = \begin{cases} \alpha x, x < 0 \\ x, otherwise \end{cases}$	$f(x) = x \cdot \Phi(x)$

While MLPs can learn temporal patterns within fixed windows, they lack explicit mechanisms for modelling long-range dependencies or variable-length sequences. However, they often serve as fundamental building blocks in more sophisticated architectures designed for time series analysis. In the context of time series imputation, MLPs can be effective for scenarios where the temporal dependencies are relatively short-range or when the missing data patterns follow regular structures.

B.2 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) address the limitations of MLPs in handling sequential data by maintaining an internal state that captures temporal dependencies. For time series imputation, RNNs process data sequentially, making them particularly effective at learning patterns across different timescales. The basic RNN computation at the time step t is given by:

$$h_t = f(W_{hh}h_{t-1} + W_{xh}x_t + b_h) \tag{B.2}$$

where h_t represents the hidden state at time t, W_{hh} and W_{xh} are weight matrices, and b_h is the bias term. However, basic RNNs often struggle with long-term dependencies due to

the vanishing or exploding gradient problem, which occurs during the backpropagation through time process. A basic RNN node is shown in Figure B.2.

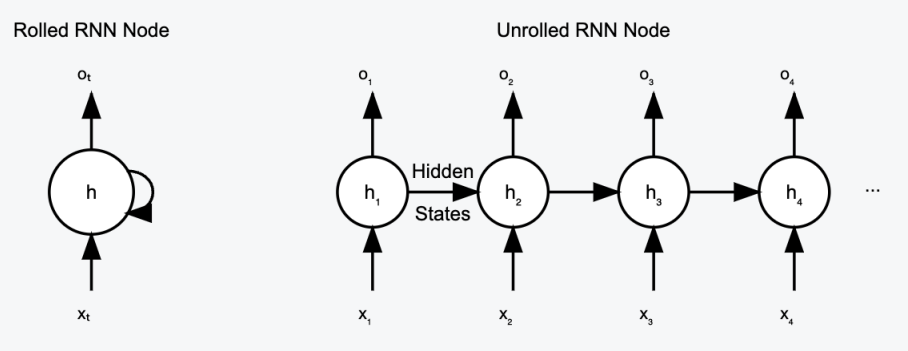


Fig. B.2 The rolled form shows the compact representation with a single recurrent node h processing inputs x_t and producing outputs o_t over time. The unrolled form demonstrates how the same network processes a sequence, with hidden states h_1 through h_4 carrying information forward through the network.

B.2.1 Advanced RNN Architectures

To address the limitations of basic RNNs, several advanced architectures have been developed. The Long Short-Term Memory (LSTM) network introduces sophisticated gating mechanisms to control information flow through the network. These gates include:

$$\begin{aligned} i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\ f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \\ o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \end{aligned} \tag{B.3}$$

where i_t, f_t, and o_t represent the input, forget, and output gates, respectively. The Gated Recurrent Unit (GRU) offers a simplified alternative to LSTM with fewer parameters, combining the input and forget gates into a single update gate while maintaining comparable performance.

B.3 Convolutional Neural Networks

While traditionally associated with image processing, Convolutional Neural Networks (CNNs) have proven effective for time series analysis through temporal convolutions. The key advantage is their ability to learn hierarchical patterns at different temporal scales through their layered architecture, as shown in Figure B.3. For time series data, 1D convolutions operate over the temporal dimension:

$$y[t] = \sum_{i=0}^{k-1} w[i] \cdot x[t-i] \tag{B.4}$$

where k represents the kernel size and w represents learnable filters. In time series imputation, CNNs can effectively capture local temporal patterns and handle missing data by learning robust feature representations that are invariant to specific types of missing patterns.

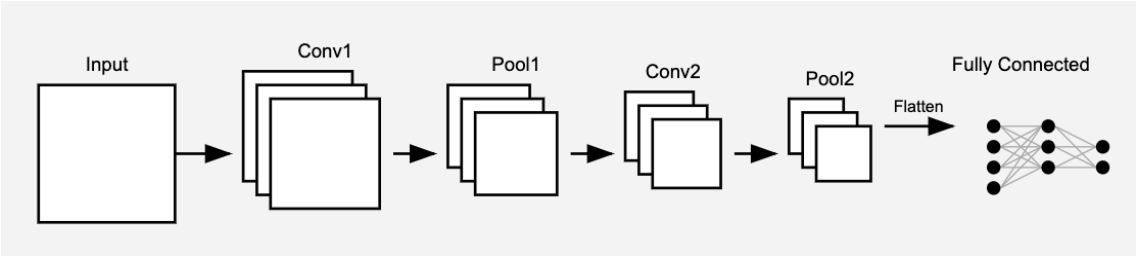


Fig. B.3 The network processes input through alternating convolution and pooling layers, which progressively reduce spatial dimensions while increasing the number of feature maps. The resulting feature maps are flattened and fed into fully connected layers in the case of classification. The decreasing size of feature maps through the network represents the hierarchical feature extraction process characteristic of CNNs.

B.4 Graph Neural Networks

For multivariate time series with complex interdependencies, Graph Neural Networks (GNNs) provide a powerful framework for modelling relationships between different variables. The graph structure can represent various types of relationships, including

temporal edges connecting time steps, feature edges connecting related variables, and domain-specific relationships encoding physical or logical constraints.

The basic message passing framework in GNNs is described by:

$$h_v^{(k+1)} = \phi(h_v^{(k)}, \{\psi(h_v^{(k)}, h_u^{(k)}, e_{uv}) : u \in \mathcal{N}(v)\}) \tag{B.5}$$

where $h_v^{(k)}$ represents the node features at layer k, ϕ and ψ are learnable functions, and $\mathcal{N}(v)$ represents the neighbourhood of node v. This formulation allows GNNs to naturally handle irregularly sampled time series and complex missing data patterns by leveraging the graph structure.

B.5 Transformer Architecture

The Transformer architecture, introduced by [170], has revolutionised sequence modelling through its self-attention mechanism. For time series imputation, Transformers offer several advantages, including parallel processing of entire sequences, direct modelling of long-range dependencies, and the ability to handle variable-length sequences. The core self-attention mechanism is computed as:

$$\text{Attention}(Q,K,V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{B.6}$$

where Q, K, and V represent the queries, keys, and values, respectively, and d_k is the dimension of the keys. In the context of time series imputation, the self-attention mechanism allows the model to dynamically focus on relevant temporal patterns when predicting missing values.

B.6 Modern Learning Frameworks

Recent advances in deep learning have introduced several powerful frameworks particularly relevant to time series imputation. These frameworks provide different approaches to modelling uncertainty and generating realistic imputations.

B.6.1 Diffusion Models

Diffusion models have emerged as a powerful framework for generating realistic imputations by learning to gradually denoise data. The forward process is defined as:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \tag{B.7}$$

where β_t represents the noise schedule. The reverse process learns to gradually denoise the data, providing a principled approach to generating realistic imputations.

B.6.2 Mixture Density Networks

Mixture Density Networks (MDNs) combine neural networks with mixture models to predict probability distributions rather than point estimates.

Appendix C

Appendix: Description of the Datasets

C.1 The eICU dataset.

C.2 The MIMIC-III dataset.

C.3 The MIMIC-IV dataset.

C.4 The PhysioNet 2012 Dataset

Feature	Missingness	Positive-Missingness	Negative-Missingness	Difference between P&N
apacheadmissiondx	0.208927	0.158346	0.215488	0.057142
ethnicity	0.208927	0.158346	0.215488	0.057142
gender	0.208927	0.158346	0.215488	0.057142
GCS Total	0.798386	0.789851	0.799493	0.009642
Eyes	0.82387	0.81548	0.824958	0.009478
Motor	0.823969	0.815628	0.825051	0.009423
Verbal	0.825821	0.820868	0.826464	0.005596
admissionheight	0.21446	0.164	0.221007	0.057007
admissionweight	0.222522	0.168045	0.229589	0.061544
age	0.208927	0.158346	0.215488	0.057142
Heart Rate	0.275438	0.216145	0.28313	0.066985
MAP (mmHg)	0.957822	0.962035	0.957275	-0.004760
Invasive BP Diastolic	0.306338	0.246144	0.314146	0.068002
Invasive BP Systolic	0.306353	0.246103	0.314169	0.068066
O2 Saturation	0.39559	0.360601	0.400129	0.039528
Respiratory Rate	0.324984	0.268013	0.332374	0.064361
Temperature (C)	0.711669	0.621156	0.723411	0.102255
glucose	0.805305	0.759432	0.811256	0.051824
FiO2	0.975856	0.946642	0.979646	0.033004
pH	0.972348	0.940238	0.976514	0.036276
Count	**0.52882195**	**0.48868825**	**0.5340282**	**0.045340**

Table C.1 Statistical information on each variable of eICU dataset.

Feature	Missingness	Positive-Missingness	Negative-Missingness	Difference between P&N
Gastric Gastric Tube	0.99321	0.990656	0.993437	0.002781
Stool Out Stool	0.998705	0.998865	0.998691	-0.000174
Urine Out Incontinent	0.995952	0.99805	0.995765	-0.002285
Ultrafiltrate Ultrafiltrate	0.9999	0.999681	0.999919	0.000238
Foley	0.741471	0.691241	0.745945	0.054704
Void	0.975109	0.99305	0.973511	-0.019539
Condom Cath	0.998753	0.999326	0.998702	-0.000624
Fecal Bag	0.999452	0.998812	0.999509	0.000697
Ostomy (output)	0.999497	0.999184	0.999525	0.000341
Chest Tube #1	0.967253	0.994379	0.964836	-0.029543
Chest Tube #2	0.99693	0.998493	0.99679	-0.001703
Jackson Pratt #1	0.995427	0.994982	0.995467	0.000485
OR EBL	0.997687	0.998688	0.997598	-0.001090
Pre-Admission	0.995546	0.994734	0.995619	0.000885
TF Residual	0.998875	0.996773	0.999062	0.002289
Albumin 5%	0.997514	0.997677	0.9975	-0.000177
Dextrose 5%	0.933518	0.906879	0.935891	0.029012
Fresh Frozen Plasma	0.997521	0.994167	0.99782	0.003653
Lorazepam (Ativan)	0.996086	0.995213	0.996163	0.000950
Calcium Gluconate	0.992298	0.98805	0.992676	0.004626
Midazolam (Versed)	0.989279	0.976578	0.990411	0.013833
Phenylephrine	0.983649	0.970337	0.984835	0.014498
Furosemide (Lasix)	0.994747	0.99367	0.994843	0.001173
Hydralazine	0.996995	0.996206	0.997065	0.000859
Norepinephrine	0.986981	0.959947	0.989389	0.029442
Magnesium Sulfate_input	0.990216	0.990089	0.990228	0.000139
Nitroglycerin	0.989616	0.997482	0.988915	-0.008567
Insulin - Regular	0.973135	0.974592	0.973005	-0.001587
Insulin - Glargine	0.998221	0.999238	0.99813	-0.001108

Table C.2 Statistical information on each variable of MIMIC-III (89f) dataset (Part 1 of 2).

Feature	Missingness	Positive-Missingness	Negative-Missingness	Difference between P&N
Insulin - Humalog	0.994348	0.993475	0.994426	0.000951
Heparin Sodium	0.995034	0.993191	0.995198	0.002007
Morphine Sulfate	0.985635	0.988918	0.985342	-0.003576
Potassium Chloride_input	0.982362	0.983918	0.982223	-0.001695
Packed Red Blood Cells	0.992456	0.991259	0.992562	0.001303
Gastric Meds	0.992479	0.983989	0.993235	0.009246
D5 1/2NS	0.995771	0.996791	0.99568	-0.001111
LR	0.979494	0.983901	0.979102	-0.004799
K Phos	0.998524	0.998032	0.998567	0.000535
Solution	0.953869	0.941418	0.954979	0.013561
Sterile Water	0.997816	0.997004	0.997888	0.000884
Metoprolol	0.99418	0.993794	0.994214	0.000420
Piggyback	0.990473	0.985691	0.990899	0.005208
OR Crystalloid Intake	0.995516	0.9975	0.995339	-0.002161
OR Cell Saver Intake	0.998547	0.99984	0.998432	-0.001408
PO Intake	0.960913	0.983865	0.958869	-0.024996
GT Flush	0.995633	0.98984	0.996149	0.006309
KCL (Bolus)	0.991311	0.993209	0.991142	-0.002067
Magnesium Sulfate (Bolus)	0.9902	0.990089	0.99021	0.000121
HEMATOCRIT	0.927576	0.923121	0.927973	0.004852
WHITE BLOOD CELLS	0.941353	0.932447	0.942147	0.009700
PLATELET COUNT	0.939807	0.930833	0.940607	0.009774
HEMOGLOBIN	0.941026	0.932411	0.941793	0.009382
MCHC	0.941488	0.93266	0.942275	0.009615
MCH	0.941516	0.932713	0.9423	0.009587
MCV	0.941517	0.932713	0.942302	0.009589
RED BLOOD CELLS	0.941513	0.932713	0.942297	0.009584
RDW	0.94153	0.932748	0.942313	0.009565
POTASSIUM	0.936206	0.92078	0.937581	0.016801
SODIUM	0.937406	0.920603	0.938903	0.018300
CHLORIDE	0.937313	0.922287	0.938651	0.016364
BICARBONATE	0.940475	0.925727	0.941788	0.016061
ANION GAP	0.94152	0.926011	0.942902	0.016891
UREA NITROGEN	0.938296	0.923351	0.939628	0.016277
CREATININE	0.937966	0.923138	0.939286	0.016148
GLUCOSE	0.94282	0.926578	0.944266	0.017688
MAGNESIUM	0.951154	0.934131	0.952671	0.018540
CALCIUM, TOTAL	0.954934	0.936223	0.956601	0.020378
PHOSPHATE	0.955243	0.936312	0.956929	0.020617
INR(PT)	0.954948	0.944273	0.955899	0.011626
PT	0.954944	0.944273	0.955895	0.011622
PTT	0.953182	0.943475	0.954047	0.010572
LYMPHOCYTES	0.980875	0.974131	0.981476	0.007345
MONOCYTES	0.980875	0.974131	0.981476	0.007345
NEUTROPHILS	0.980875	0.974131	0.981476	0.007345
BASOPHILS	0.980875	0.974131	0.981476	0.007345
EOSINOPHILS	0.980875	0.974131	0.981476	0.007345
BILIRUBIN, TOTAL	0.980629	0.97016	0.981561	0.011401
PH	0.932283	0.907216	0.934516	0.027300
BASE EXCESS	0.974563	0.974557	0.974564	0.000007
CALCULATED TOTAL CO2	0.950026	0.92695	0.952082	0.025132
PO2	0.95002	0.926933	0.952077	0.025144
PCO2	0.950026	0.926933	0.952083	0.025150
SPECIFIC GRAVITY	0.98501	0.981578	0.985316	0.003738
LACTATE	0.962123	0.939628	0.964127	0.024499
ALANINE AMINOTRANSFERASE (ALT)	0.98042	0.969894	0.981357	0.011463
ASPARATE AMINOTRANSFERASE (AST)	0.980426	0.969894	0.981364	0.011470
ALKALINE PHOSPHATASE	0.980926	0.970691	0.981838	0.011147
ALBUMIN	0.987082	0.980957	0.987628	0.006671
Pantoprazole	0.998799	0.998688	0.998809	0.000121
Count	**0.974160786**	**0.968324082**	**0.974680755**	**0.006357**

Table C.3 Statistical information on each variable of MIMIC-III (89f) dataset (Part 2 of 2).

Feature	Missingness	Positive-Missingness	Negative-Missingness	Difference between P&N
Capillary refill rate-0	0.99732	0.998332	0.997166	-0.001166
Capillary refill rate-1	0.99732	0.998332	0.997166	-0.001166
Diastolic blood pressure	0.360733	0.354211	0.361727	0.007516
Fraction inspired oxygen	0.950456	0.935295	0.952767	0.017472
Glascow coma scale eye opening-To Pain	0.765768	0.749486	0.768251	0.018765
Glascow coma scale eye opening-3 To speech	0.765768	0.749486	0.768251	0.018765
Glascow coma scale eye opening-1 No Response	0.765768	0.749486	0.768251	0.018765
Glascow coma scale eye opening-4 Spontaneously	0.765768	0.749486	0.768251	0.018765
Glascow coma scale eye opening-None	0.765768	0.749486	0.768251	0.018765
Glascow coma scale eye opening-To Speech	0.765768	0.749486	0.768251	0.018765
Glascow coma scale eye opening-Spontaneously	0.765768	0.749486	0.768251	0.018765
Glascow coma scale eye opening-2 To pain	0.765768	0.749486	0.768251	0.018765
Glascow coma scale motor response-1 No Response	0.76683	0.751058	0.769235	0.018177
Glascow coma scale motor response-3 Abnorm flexion	0.76683	0.751058	0.769235	0.018177
Glascow coma scale motor response-Abnormal extension	0.76683	0.751058	0.769235	0.018177
Glascow coma scale motor response-No response	0.76683	0.751058	0.769235	0.018177
Glascow coma scale motor response-4 Flex-withdraws	0.76683	0.751058	0.769235	0.018177
Glascow coma scale motor response-Localizes Pain	0.76683	0.751058	0.769235	0.018177
Glascow coma scale motor response-Flex-withdraws	0.76683	0.751058	0.769235	0.018177
Glascow coma scale motor response-Obeys Commands	0.76683	0.751058	0.769235	0.018177
Glascow coma scale motor response-Abnormal Flexion	0.76683	0.751058	0.769235	0.018177
Glascow coma scale motor response-6 Obeys Commands	0.76683	0.751058	0.769235	0.018177
Glascow coma scale motor response-5 Localizes Pain	0.76683	0.751058	0.769235	0.018177
Glascow coma scale motor response-2 Abnorm extensn	0.76683	0.751058	0.769235	0.018177
Glascow coma scale total-11	0.85832	0.84391	0.860517	0.016607
Glascow coma scale total-10	0.85832	0.84391	0.860517	0.016607
Glascow coma scale total-13	0.85832	0.84391	0.860517	0.016607
Glascow coma scale total-12	0.85832	0.84391	0.860517	0.016607
Glascow coma scale total-15	0.85832	0.84391	0.860517	0.016607
Glascow coma scale total-14	0.85832	0.84391	0.860517	0.016607
Glascow coma scale total-3	0.85832	0.84391	0.860517	0.016607
Glascow coma scale total-5	0.85832	0.84391	0.860517	0.016607
Glascow coma scale total-4	0.85832	0.84391	0.860517	0.016607
Glascow coma scale total-7	0.85832	0.84391	0.860517	0.016607
Glascow coma scale total-6	0.85832	0.84391	0.860517	0.016607
Glascow coma scale total-9	0.85832	0.84391	0.860517	0.016607
Glascow coma scale total-8	0.85832	0.84391	0.860517	0.016607
Glascow coma scale verbal response-1 No Response	0.766542	0.750238	0.769028	0.018790
Glascow coma scale verbal response-No Response	0.766542	0.750238	0.769028	0.018790
Glascow coma scale verbal response-Confused	0.766542	0.750238	0.769028	0.018790
Glascow coma scale verbal response-Inappropriate Words	0.766542	0.750238	0.769028	0.018790
Glascow coma scale verbal response-Oriented	0.766542	0.750238	0.769028	0.018790
Glascow coma scale verbal response-No Response-ETT	0.766542	0.750238	0.769028	0.018790
Glascow coma scale verbal response-5 Oriented	0.766542	0.750238	0.769028	0.018790
Glascow coma scale verbal response-Incomprehensible sounds	0.766542	0.750238	0.769028	0.018790
Glascow coma scale verbal response-1.0 ET/Trach	0.766542	0.750238	0.769028	0.018790
Glascow coma scale verbal response-4 Confused	0.766542	0.750238	0.769028	0.018790
Glascow coma scale verbal response-2 Incomp sounds	0.766542	0.750238	0.769028	0.018790
Glascow coma scale verbal response-3 Inapprop words	0.766542	0.750238	0.769028	0.018790
Glucose	0.777611	0.791972	0.775421	-0.016551
Heart Rate	0.337305	0.335769	0.33754	0.001771
Height	0.996436	0.997088	0.996337	-0.000751
Mean blood pressure	0.361854	0.35407	0.363041	0.008971
Oxygen saturation	0.358267	0.354926	0.358776	0.003850
Respiratory rate	0.3466	0.344067	0.346987	0.002920
Systolic blood pressure	0.360597	0.354085	0.36159	0.007505
Temperature	0.747638	0.754558	0.746582	-0.007976
Weight	0.972522	0.97183	0.972627	0.000797
pH	0.869953	0.850577	0.872908	0.022331
Count	**0.764735254**	**0.752159017**	**0.766652881**	**0.014494**

Table C.4 Statistical information on each variable of MIMIC-III (59f) dataset.

Feature	Missingness	Positive-Missingness	Negative-Missingness	Difference between P&N
DiasABP	0.461237	0.414636	0.468735	0.054099
HR	0.098892	0.080551	0.101843	0.021292
Na	0.929437	0.915162	0.931733	0.016571
Lactate	0.959183	0.938403	0.962527	0.024124
NIDiasABP	0.579054	0.597999	0.576006	-0.021993
PaO2	0.883261	0.865298	0.886152	0.020854
WBC	0.933028	0.927384	0.933936	0.006552
pH	0.877841	0.86323	0.880192	0.016962
Albumin	0.987741	0.981912	0.988679	0.006767
ALT	0.983472	0.974955	0.984842	0.009887
Glucose	0.932303	0.918923	0.934456	0.015533
SaO2	0.958875	0.95901	0.958854	-0.000156
Temp	0.628821	0.643991	0.62638	-0.017611
AST	0.983446	0.974955	0.984812	0.009857
Bilirubin	0.983394	0.97503	0.98474	0.009710
HCO3	0.929207	0.918773	0.930886	0.012113
BUN	0.927618	0.917908	0.92918	0.011272
RespRate	0.760993	0.860898	0.744917	-0.115981
Mg	0.92928	0.920126	0.930753	0.010627
HCT	0.905299	0.904069	0.905497	0.001428
SysABP	0.461174	0.414523	0.46868	0.054157
FiO2	0.839833	0.785161	0.84863	0.063469
K	0.924912	0.911327	0.927098	0.015771
GCS	0.681782	0.667456	0.684087	0.016631
Cholesterol	0.998358	0.99827	0.998372	0.000102
NISysABP	0.578569	0.597473	0.575528	-0.021945
TroponinT	0.988945	0.982814	0.989931	0.007117
MAP	0.464181	0.420352	0.471234	0.050882
TroponinI	0.997738	0.996277	0.997973	0.001696
PaCO2	0.883178	0.865223	0.886067	0.020844
Platelets	0.92693	0.921781	0.927758	0.005977
Urine	0.3076	0.317013	0.306086	-0.010927
NIMAP	0.584334	0.60082	0.581681	-0.019139
Creatinine	0.927274	0.917494	0.928847	0.011353
ALP	0.983915	0.975707	0.985236	0.009529
Count	**0.805174429**	**0.7978544**	**0.806352229**	**0.008498**

Table C.5 Statistical information on each variable of PhysioNet2012 dataset.