Imputation of clinical covariates in time series

Imputation of clinical covariates in time series

Cite this article as: Dimitris Bertsimas, Agni Orfanoudaki and Colin Pawlowski, Imputation of clinical covariates in time series, Machine Learning https://doi.org/10.1007/s10994-020-05923-2

Machine Learning manuscript No.
(will be inserted by the editor)

Imputation of Clinical Covariates in Time Series

Dimitris Bertsimas · Agni Orfanoudaki · Colin Pawlowski

Received: date / Accepted: date

Abstract Missing data is a common problem in longitudinal datasets which include multiple instances of the same individual observed at different points in time. We introduce a new approach, MedImpute, for imputing missing clinical covariates in multivariate panel data. This approach integrates patient specific information into an optimization formulation that can be adjusted for different imputation algorithms. We present the formulation for a K-nearest neighbors model and derive a corresponding scalable first-order method `med.knn`. Our algorithm provides imputations for datasets with both continuous and categorical features and observations occurring at arbitrary points in time. In computational experiments on three real-world clinical datasets, we test its performance on imputation and downstream predictive tasks , varying the percentage of missing data, the number of observations per patient, and the mechanism of missing data. The proposed method improves upon both the imputation accuracy and downstream predictive performance relative to the best of the benchmark imputation methods considered. We show that this edge is consistently present both in longitudinal and electronic health records datasets as well as in binary classification and regression settings. On computational experiments on synthetic data, we test the scalability of this algorithm on large datasets, and we show that an efficient method for hyperparameter tuning scales to datasets with $10,000$'s of observations and 100's of covariates while maintaining high imputation accuracy.

Keywords missing data imputation, time series data, electronic health records, longitudinal studies, Framingham Heart Study, K-nearest neighbors

1 Introduction

Machine learning applied to healthcare data can generate actionable insights ranging from predicting the onset of disease to streamlining hospital operations. Statistical models that leverage the variety and richness of clinical data are still relatively rare and offer an exciting avenue for further research (Callahan and Shah 2017). As an increasing amount of information becomes available the medical field expects machine learning to become an indispensable tool for clinicians (Obermeyer and Emanuel 2016).

This information will come from various clinical and epidemiological sources. Claims records, clinical trials, and data from longitudinal studies have been an invaluable resource for medical research over the past decades. In many of these datasets, data from individual subjects is gathered over time via continuous or repeated monitoring of both risk factors and health outcomes. For example, longitudinal cohort studies are used to discover relationships between exposures of interest and long term health effects including adverse events and chronic disease. By design, these studies mitigate recall bias in participants by collecting data prospectively and prior to knowledge of a possible subsequent event (Caruana et al. 2015).

Dimitris Bertsimas
Operations Research Center, E40-111, Massachusetts Institute of Technology, Cambridge, MA 02139, USA. E-mail: dbertim@mit.edu

Agni Orfanoudaki
Operations Research Center, E40-111, Massachusetts Institute of Technology Cambridge, MA 02139, USA. E-mail: agniorf@mit.edu

Colin Pawlowski
Operations Research Center, E40-111, Massachusetts Institute of Technology Cambridge, MA 02139, USA. E-mail: cpawlows@mit.edu

Another valuable source of clinical data are Electronic Health Records (EHR). Over the past years, widespread uptake of EHR has generated massive datasets that contain quantitative, qualitative, and transactional data (TB and AS 2013). Their hospital adoption has skyrocketed in part due to the Health Information Technology for Economic and Clinical Health (HITECH) Act of 2009, which provided $30 billion in incentives for hospitals and physician practices to adopt EHR systems (Birkhead et al. 2015). While primarily designed for archiving patient information and performing administrative healthcare tasks, many researchers have found secondary use of these records for various clinical informatics applications (Shickel et al. 2018). Because heterogeneous labs, measurements, and notes are recorded for patients during each visit, EHR data has a rich and complex structure with time series information.

However, it is algorithms and not merely datasets that will prove transformative for the medical field (Obermeyer and Emanuel 2016). To make progress, we need to develop new statistical tools tailored to clinical applications which address the challenges and leverage common structure encountered in healthcare data. One of the most important issues is the ubiquitous presence of missing time series data (Pedersen et al. 2017), particularly for variables requiring complex, time-sensitive, or resource-intensive procedures to collect. There are many reasons for "missingness", including missed study visits, patients lost to follow-up, missing information in source documents, lack of availability (e.g., laboratory tests that were not performed), and clinical scenarios preventing collection of certain variables (e.g., missing coma scale data in sedated patients) (CD and RJ 2015). Thus, creating a consistent dataset for individuals over multiple visits even at the same healthcare organization for a fixed set of covariates remains a challenge. Even in longitudinal studies, where a set of covariates is collected over time, missing data are pervasive and complete ascertainment of all variables is rare (Landrum and Becker 2001).

The presence of missing data poses considerable challenges in the analyses and interpretation of clinical investigations' results (Wood et al. 2004), potentially weakening their validity and leading to biased inferences. Their presence may complicate interpretation or even invalidate an otherwise important study (Ware et al. 2012). Many methods commonly used for handling missing values during data analysis can yield biased results, decrease study power, or lead to underestimates of uncertainty, all reducing the chance of drawing valid conclusions (CD and RJ 2015). As many statistical models and machine learning algorithms rely on complete datasets, it is key to handle the missing data appropriately.

1.1 Review of Methods for Handling Missing Values

In this section, we present some of the most common approaches for missing data imputation. First, we introduce fairly simple and intuitive techniques that do not require the use of sophisticated machine learning methods. We then provide brief descriptions of advanced missing data imputation algorithms, both general purpose methods as well as approaches tailored to medical records and time series.

Excluding observations that contain missing values has been a standard practice for clinical research, primarily due to the lack of interpretable, accurate machine learning methods that can be easily applied by medical researchers (Sterne et al. 2009; Janssen et al. 2010). Unsurprisingly, complete case analysis may suffer from severe bias and the reduced sample size results in lower study power (CD and RJ 2015). Recent advances in machine learning have allowed missing values to be accurately imputed prior to running statistical analyses on the complete dataset. The benefit of the latter approach is that once a set (or multiple sets) of complete data has been generated, practitioners can easily apply their own learning algorithms to the imputed dataset. In healthcare settings, often times those datasets contain numerous visits of the same person corresponding to various patterns of missing data. This special structure challenges state-of-the-art missing data methods which do not consider the connection of multiple observations to the same individual (Che et al. 2018).

A variety of machine learning approaches have been introduced in the literature to impute missing values ignoring the potential dependency between observations of the same individual. The simplest approach is the `mean` imputation that uses the mean of the observed values to replace those missing for the same covariate (Little and Rubin 2019). However, `mean` imputation underestimates the variance, ignores the correlation between the features leading to poor imputation outcomes.

Another common method called `bpca` uses the singular value decomposition (SVD) of the data matrix and information from a Bayesian prior distribution on the model parameters to impute missing values. This method outperforms basic SVD methods (Oba et al. 2003). In cases where the level of missing data is above 30%, we have found that this method reduces to `mean` imputation, leading to similar biases (Faria et al. 2018).

Joint modeling assumes the existence of a joint distribution on the entire dataset and a parametric density function on the data given model parameters. Current implementations of the method estimate the model parameters using an Expectation-Maximization (EM) approach in order to maximize the likelihood function. One widely used software package which implements this approach, `Amelia I`, assumes that data are drawn from a multivariate normal distribution (Honaker et al. 1999). In practice, healthcare data typically violate this condition (Sterne et al. 2009).

Recent review articles indicate that single imputation methods can lead to seriously misleading results and advise us to consider multiple imputation (Janssen et al. 2010; Little and Rubin 2019). This approach, implemented in the software package `mice`, allows for uncertainty about the missing data by creating several different plausible imputed datasets and appropriately combining results obtained from each of them (Schafer and Olsen 1998). The `Amelia I` package was extended to multiple imputation in the `Amelia II` algorithm (Honaker et al. 2011). Multiple imputation entails two stages: (1) generating replacement values for missing data and repeating this procedure many times, resulting in many datasets with replaced missing information, and (2) analyzing the many imputed datasets and combining the results (P et al. 2015). As a result, multiple imputation methods are slower and require pooling results, which may not be appropriate for certain applications. For example, in clinical applications, where the interpretability of the underlying model matters, a single imputed dataset and simple predictive model may be preferred.

Most recently, Bertsimas et al. (Bertsimas et al. 2018b) proposed a general optimization framework with a predictive model-based cost function that can explicitly handle both continuous and categorical variables and can be used to generate single, as well as multiple, imputations. This optimization perspective has led to new scalable algorithms for more accurate data imputation. We describe this method OptImpute in more detail in Section 2.2, which we use as a foundation for the imputation method proposed in this paper.

The algorithms above are not tailored to multivariate time series datasets despite the fact that covariates may be strongly correlated over time (Lipton et al. 2016). Preliminary work has been done demonstrating their performance in that setting (Zhang 2016). Recurrent Neural Network approaches have also been employed to handle missing values in time series among the covariates for a particular prediction task (Lipton et al. 2016; Che et al. 2018). However, these approaches differ from traditional imputation methods because they also use features derived from the missing pattern itself, and they require that the downstream learning method is a neural network. In contrast, our method produces a single imputed dataset that can be used as training data for any supervised learning method which is preferred for the downstream task.

In practice, simpler techniques are more commonly applied in the panel data setting. Researchers often opt for a moving average approach with a fixed time window using previous observations from the same individual (Flores et al. 2019). For example, the last-observation-carried-forward method is used to impute a present missing value by carrying only the last non-missing value forward for a defined time period (Siddiqui and Ali 1998). However, these techniques ignore the correlation between covariates which is leveraged by other more advanced imputation methods. There have been a few methods that give weights to instances of the same patient in temporal data. For example, this approach has been applied to adverse drug events monitoring (Zhao and Henriksson 2016). In addition, similar methods have been applied in the political science and economics fields where time-series cross-sectional data are quite common (Shor et al. 2007).

1.2 Contributions

Given multivariate time series data, we develop a novel imputation method that utilizes optimization and machine learning techniques and outperforms state-of-the-art algorithms. Our contributions are as follows:

1. We formulate the problem of missing data imputation with time series information under the MedImpute framework, extending the OptImpute framework proposed by Bertsimas et al. (2018b). Our approach can be adjusted to account for different imputation models based on predictive methods such as K-NN, SVM, and trees. We focus on a K-NN formulation to solve the problem and derive a corresponding fast first-order algorithm `med.knn`. This method provides imputations for datasets with both continuous and categorical features and observations occurring at arbitrary points in time.
2. We design a series of computational experiments on three real-world sets of data with direct clinical implications. We consider the Framingham Heart Study (FHS) and the Parkinson's Progression Markers Initiative (PPMI), two longitudinal datasets with rich time series data recorded at regular time intervals, and Electronic Health Record (EHR) data from the Dana Farber Cancer Institute (DFCI), which is less structured and more sparse time series data. We provide a comprehensive framework for our experiments that tests the performance of our method across a diverse range of scenarios, varying parameters including: (1) the percentage of missing data, (2) the number of observations per individual, and (3) the

mechanism of missing data. For the latter, we consider different mechanisms for the longitudinal and EHR datasets corresponding to the different patterns of missing data which are typically observed in real-world datasets. We demonstrate that `med.knn` obtains the best predictive performance and lowest imputation error as we vary the missing percentage from 10% to 50%. In addition, we show that for all datasets, the relative performance of `med.knn` improves as we increase the number of observations per individual. Finally, we demonstrate that `med.knn` performs well on missing patterns commonly encountered in practice for both longitudinal studies and EHR data. These improvements are relative to the best of the comparator methods among `amelia`, `moving average`, `mean`, `bpca`, `mice`, and `opt.knn`, which are described in Section 3.

3. We propose a new custom tuning procedure to efficiently learn the hyperparameters in the optimization problem avoiding the use of traditional approaches such as Grid Search. Our methodology allows for decoupling the problem into multiple parts, enabling parallel computation that can decrease the run time. We create synthetic EHR data to test the scaling performance of the algorithm as we increase the number of observations and features. Our results show that the custom tuning approach leads to both superior scaling performance and better imputation accuracy compared to standard cross-validation. The tuning procedure is described in Section 2.4 and the scaling experiments with synthetic data are provided in Section 4.

The structure of the paper is as follows. In Section 2, we describe our framework for imputation of clinical covariates in time series and proposed method `med.knn`. In Section 3, we describe computational experiments on three real-world datasets evaluating both imputation and prediction accuracy. In Section 4, we present scaling experiments on simulated clinical datasets. In Section 5, we discuss properties of our algorithm and key insights from our experiments. We conclude our work in Section 6.

2 Methods

In this section, we describe our proposed method for imputation. In Section 2.1, we define variables and notation that we use in this paper. In Section 2.2, we review the OptImpute framework for missing data imputation. In Section 2.3, we introduce our new framework for imputation MedImpute which directly models clinical covariates in time series, and we present the K-Nearest Neighbors (K-NN) based formulation. In Section 2.4, we describe a custom tuning procedure to efficiently learn the hyperparameters in the optimization problem. Finally, in Section 2.5 we provide the detailed steps of the first-order method `med.knn` that can be used to find high-quality solutions.

2.1 Variables and Notation

In this paper, we consider the single imputation problem for which our task is to fill in the missing values of dataset $\mathbf{X} \in \mathbb{R}^{n \times p}$ with n observations (rows) and p features (columns). Without loss of generality, we assume that the first p_0 features are continuous and that the next $p_1 = p - p_0$ features are categorical, and the missing and known indices are specified by the following sets:

$$\begin{aligned}
\mathcal{M}_0 &= \{(i,d) : \text{entry } x_{id} \text{ is missing, } 1 \le d \le p_0, \ 1 \le i \le n\}, \\
\mathcal{N}_0 &= \{(i,d) : \text{entry } x_{id} \text{ is known, } 1 \le d \le p_0, \ 1 \le i \le n\}, \\
\mathcal{M}_1 &= \{(i,d) : \text{entry } x_{id} \text{ is missing, } p_0 + 1 \le d \le p_0 + p_1, \ 1 \le i \le n\}, \\
\mathcal{N}_1 &= \{(i,d) : \text{entry } x_{id} \text{ is known, } p_0 + 1 \le d \le p_0 + p_1, \ 1 \le i \le n\}, \\
\mathcal{I} &= \{i : \mathbf{x}_i \text{ has one or more missing values}\}.
\end{aligned} \tag{1}$$

Here, $\mathcal{M}_0$, $\mathcal{M}_1$ are the sets of indices of the missing values in the continuous and categorical variables, respectively. Similarly, $\mathcal{N}_0$, $\mathcal{N}_1$ are the sets of indices of the known values in the continuous and categorical variables, respectively. $\mathcal{I}$ is the set of rows which contains at least one missing value.

We suppose that all of the continuous variables are normalized with unit standard deviation and that the d^{th} categorical variable takes value among k_d classes. Given this data, we introduce the decision variables $\mathbf{W} \in \mathbb{R}^{n \times p_0}$, $\mathbf{V} \in \{1, \ldots, k_{p_0+1}\} \times \ldots \times \{1, \ldots, k_{p_0+p_1}\}$ to be the matrices of imputed continuous and categorical variables, respectively. For each entry x_{id}, w_{id} is the imputed value if $d \in \{1, \ldots, p_0\}$, and v_{id} is the imputed value if $d \in \{p_0+1, \ldots, p_0+p_1\}$. We refer to the full imputation for observation $\mathbf{x}_i$ as $(\mathbf{w}_i, \mathbf{v}_i)$. For the MedImpute method, we also assume that each observation $\mathbf{x}_i$ corresponds to a particular patient with the unique ID y_i observed at time-stamp t_i.

2.2 Review of OptImpute

Next, we review the OptImpute framework for general imputation which we use as a foundation for our method. In this approach, we formulate the missing data problem as an optimization problem in which all entries are simultaneously filled in and used as covariates to predict the other entries. Our key decision variables are the imputed values $\{w_{id} : (i,d) \in \mathcal{M}_0\}$ and $\{v_{id} : (i,d) \in \mathcal{M}_1\}$. We will also introduce auxiliary decision variables $\mathbf{Z}$. For any given set of imputed values and a corresponding data $\mathbf{X}$, we associate a cost function $c(\cdot)$ to it. Thus, our objective is to solve the following optimization problem:

$$\begin{array}{lll} \min & c(\mathbf{Z},\mathbf{W},\mathbf{V};\mathbf{X}) & \\ \text{s.t.} & w_{id} = x_{id} & (i,d) \in \mathcal{N}_0, \\ & v_{id} = x_{id} & (i,d) \in \mathcal{N}_1, \\ & (\mathbf{Z},\mathbf{W},\mathbf{V}) \in \mathcal{Z}, & \end{array} \tag{2}$$

where $\mathcal{Z}$ is the set of all feasible combinations $(\mathbf{Z},\mathbf{W},\mathbf{V})$ of auxiliary vectors and imputations. In this paper, we only consider an OptImpute formulation based upon K-Nearest Neighbors (K-NN), however it is also possible to consider formulations based upon SVM and trees (Bertsimas et al. 2018b).

In the K-NN formulation, the objective is to impute the missing values so that each point is as close to its K-nearest neighbors as possible. First, we define a distance metric on the dataset. Given two observations i and j, we say that the distance between them is:

$$d_{ij} := \sum_{d=1}^{p_0} (w_{id} - w_{jd})^2 + \sum_{d=p_0+1}^{p_0+p_1} \mathbb{1}_{\{v_{id} \neq v_{jd}\}}. \tag{3}$$

In this distance metric, we weight the contributions from the continuous and categorical variables equally, but it is also possible to introduce a scaling factor to weight these terms differently. Given this distance metric, we introduce the binary variables $\mathbf{Z} \in \{0,1\}^{|\mathcal{I}| \times n}$, where

$$z_{ij} = \begin{cases} 1, \text{ if } j \text{ is among the } K\text{-nearest neighbors of } i \\ \quad \text{with respect to distance metric (3)}, \\ 0, \text{ otherwise.} \end{cases} \tag{4}$$

The OptImpute formulation with the K-NN objective function is

$$\begin{array}{lll} \min & \displaystyle\sum_{i \in \mathcal{I}} \sum_{j=1}^{n} z_{ij} \left(\sum_{d=1}^{p_0} (w_{id} - w_{jd})^2 + \sum_{d=p_0+1}^{p_0+p_1} \mathbb{1}_{\{v_{id} \neq v_{jd}\}} \right) & \\ \text{s.t.} & w_{id} = x_{id} & (i,d) \in \mathcal{N}_0, \\ & v_{id} = x_{id} & (i,d) \in \mathcal{N}_1, \\ & z_{ii} = 0 & i \in \mathcal{I}, \\ & \displaystyle\sum_{j=1}^{n} z_{ij} = K & i \in \mathcal{I}, \\ & \mathbf{Z} \in \{0,1\}^{|\mathcal{I}| \times n}, & \end{array} \tag{5}$$

where $\mathcal{I} = \{i : \mathbf{x}_i \text{ has one or more missing values}\}$. Problem (5) is non-convex with integer constraints for the categorical variables. In order to solve this problem, the authors find near optimal feasible solutions using first-order methods with random and targeted warm starts, resulting in a new imputation algorithm called `opt.knn` (Bertsimas et al. 2018b).

At a high level, the `opt.knn` algorithm works as follows. The user provides as input an incomplete data matrix $\mathbf{X}$, a convergence threshold $\delta_0 > 0$, and a warm start imputation $(\mathbf{W}^0, \mathbf{V}^0)$. The output of the algorithm is the full matrix $\mathbf{X}^{imp}$ with the imputed variables. In each iteration, we alternate updating the auxiliary variables $\mathbf{Z}$ and the imputation $(\mathbf{W},\mathbf{V})$ using either Coordinate Descent (CD) or Block Coordinate Descent (BCD). The problem of updating $\mathbf{Z}$ given an imputation reduces to a simple sorting procedure on the distances. To update $(\mathbf{W},\mathbf{V})$ in CD, we locally optimize each imputed value (w_{id} or v_{id}) one at a time. To update $(\mathbf{W},\mathbf{V})$ in BCD, for each continuous or categorical feature we solve a Quadratic Optimization problem or a Mixed-Integer Optimization problem, respectively. We continue updating these values until

the objective value stops improving by a sufficiently large amount δ_0. Notice that the objective function value is strictly decreasing by at least δ_0 at every iteration until the algorithm terminates. As a result, the number of steps required for the algorithm termination is:

$$T = \frac{1}{\delta_0} c(\mathbf{Z}^0, \mathbf{W}^0, \mathbf{V}^0; \mathbf{X}), \tag{6}$$

where $\mathbf{W}^0, \mathbf{V}^0$ are the warmstart values, $\mathbf{X}$ is data, and $\mathbf{Z}^0$ is the initialized auxiliary variables. There are no analytical guarantees that the algorithm will find the globaly optimal solution (Wright 2015). We repeat this process for multiple warm starts and take the solution with the best objective value to be the final imputation. The algorithm for a single warm start is summarized in Algorithm 1.

Algorithm 1 `opt.knn`

Input: Incomplete data matrix $\mathbf{X}$,
warm start $[\mathbf{W}^0, \mathbf{V}^0]$,
max number of iterations $T \geq 0$.
Output: $\mathbf{X}^{imp}$ a full matrix with imputed values.
Procedure:
Initialize $t \leftarrow 0$, $\mathbf{W}^* \leftarrow \mathbf{W}^0$, $\mathbf{V}^* \leftarrow \mathbf{V}^0$.
while $t < T$ **do**
① Find the K nearest neighbors for each observation i, and update $\mathbf{Z}^*$ accordingly.
② Update the imputation $(\mathbf{W}^*, \mathbf{V}^*)$, following either Block Coordinate Descent or Coordinate Descent (details in Bertsimas et al. (2018b)).
③ Increment $t \leftarrow t + 1$.
end while
return $\mathbf{X}^{imp} \leftarrow [\mathbf{W}^*; \mathbf{V}^*]$.

2.3 MedImpute

In this section, we present the MedImpute framework for imputation of clinical covariates in time series. We extend the general OptImpute framework by weighting instances of the same person in the imputation model. We focus on the K-NN classifier and provide the specific formulation to solve this problem. Our new framework takes into account the time series structure frequently encountered in healthcare data. In addition, unlike univariate time series methods, this approach leverages statistical correlations between multiple clinical covariates.

Suppose that we are given the same problem setup for single imputation as described in Section 2.2. In addition, assume that each observation i corresponds to an individual patient with unique identifier $y_i \in \{1, \ldots, M\}$ recorded at a particular time point. For datasets with multiple observations of individuals over time, we have $M < n$. Define $t_i \in \mathbb{R}^+$ as the number of (days/months/years) after a reference date that observation i was recorded. It follows that $|t_i - t_j|$ is the time difference in (days/months/years) between observations i and j. Note that this framework captures the common structure of many clinical datasets collected over time, including longitudinal studies, insurance claims, and EHR data.

For each clinical covariate $d = 1, \ldots, p$, we introduce the parameters α_d, h_d. We learn α_d and h_d via a custom tuning procedure which we describe in Section 2.4. The first learned parameter $\alpha_d \in [0, 1]$ is the relative weight given to the time series component of the objective function for variable d. At the extremes, $\alpha_d = 0$ corresponds to imputing covariate d under the OptImpute objective, and $\alpha_d = 1$ corresponds to imputing covariate d using each individual's time series information independently. The second learned parameter $h_d \in (0, \infty)$ is the halflife parameter for the covariate d. This parameter is called the "halflife" parameter because it is the halflife of an exponential decay function $f(x) = 2^{-x/h_d}$ that we use to determine the relative weights for multiple observations of the same patient.

We introduce this parameter h_d so that observations from the same individual at nearby points in time will be weighted most heavily in the imputation. We make this design decision under the assumption that each clinical covariate can be approximated as a continuous function which is relatively smooth over time. For example, Body Mass Index (BMI) is a clinical covariate with values that are relatively smooth over time. Under this model, we assume that a BMI measurement from one week ago is more predictive of a patient's current BMI than a BMI measurement from one year ago. However, we do not make any assumptions about how much more/less predictive these different measurements are, only that their relative

weights follow an exponential distribution. The halflife of this exponential distribution for covariate d is the modelling parameter that we refer to as h_d.

For each pair of observations i, j, covariate d, and corresponding halflife parameter h_d, define the two derived parameters:

$$\begin{aligned} C_{ijd} &= \begin{cases} 2^{-|t_i - t_j|/h_d}, \text{ if } y_i = y_j, \\ 0, \text{ otherwise,} \end{cases} \\ \overline{C}_{ijd} &= \frac{C_{ijd}}{\sum\limits_{\{j' : y_i = y_{j'}, j' \neq i\}} C_{ijd}}. \end{aligned} \tag{7}$$

The first derived parameter C_{ijd} is the relative weight that observation j is given for time-series based imputation of observation i in covariate d. Note that this parameters is only non-zero when $y_i = y_j$, i.e. i and j are observations from the same patient. For example, if $h_d = 7$ days, then past observations of covariate d from one week and two weeks ago from the same patient would be given relative weights 0.5 and 0.25, respectively. The second derived parameter, $\overline{C}_{ijd}$, is the normalized variation of C_{ijd}. In particular, $\overline{C}_{ijd}$ is the relative weight that observation j is given to impute observation i in covariate d, divided by the sum of all relative weights of observations from the same patient in covariate d.

The MedImpute formulation with the K-NN objective function is

$$\begin{aligned} \min \quad & \frac{1}{K} \sum_{i \in \mathcal{I}} \sum_{j=1}^{n} z_{ij} \left(\sum_{d=1}^{p_0} (1 - \alpha_d)(w_{id} - w_{jd})^2 + \sum_{d=p_0+1}^{p_0+p_1} (1 - \alpha_d) \mathbb{1}_{\{v_{id} \neq v_{jd}\}} \right) \\ & + \sum_{i \in \mathcal{I}} \sum_{j=1}^{n} \left(\sum_{d=1}^{p_0} \alpha_d \overline{C}_{ijd} (w_{id} - w_{jd})^2 + \sum_{d=p_0+1}^{p_0+p_1} \alpha_d \overline{C}_{ijd} \mathbb{1}_{\{v_{id} \neq v_{jd}\}} \right) \\ \text{s.t.} \quad & w_{id} = x_{id} \qquad (i,d) \in \mathcal{N}_0, \\ & v_{id} = x_{id} \qquad (i,d) \in \mathcal{N}_1, \\ & z_{ii} = 0 \qquad i \in \mathcal{I}, \\ & \sum_{j=1}^{n} z_{ij} = K \qquad i \in \mathcal{I}, \\ & \mathbf{Z} \in \{0,1\}^{|\mathcal{I}| \times n}, \end{aligned} \tag{8}$$

where $\mathcal{I} = \{i : \mathbf{x}_i \text{ has one or more missing values}\}$ and α_d, $\overline{C}_{ijd}$ are constants. This problem is equivalent to (5) plus a penalty term in the objective for each feature d with different weights α_d in order to account for instances of the same person in the dataset. At the optimal solution, the objective function is the sum of the distances from each point to its K-nearest neighbors with respect to distance metric (3), plus the sum of the distances from each point to other observations from the same individual.

We derive a fast algorithm to provide high quality solutions to this problem using first order methods with random restarts, alternatively updating the binary variables and the imputed values as in `opt.knn` (Bertsekas 1999). In Algorithm 2, we summarize the `med.knn` method for a single warm start. In the next section, we describe the steps of this algorithm in detail.

MedImpute provides a flexible framework that can be easily extended as well. For example, we may consider other predictive models besides K-NN such as support vector machines and decision tree based methods by adjusting the objective functions of the corresponding OptImpute formulations appropriately. We refer the reader to (Bertsimas et al. 2018b) for more discussion on these alternate formulations, which is a possible area of future work. In these cases, we add the same penalty term to the objective functions that we added in formulation (8), and we solve using first-order methods with random starts. In this manuscript, we focus on the K-NN formulation due to the method's simplicity that is close to the medical practice. The idea of imputing a patient's missing values using the mean or the mode of the covariates from the most similar individuals to that observation is intuitive. Various implementations of the heuristic K-NN approach are already widely accepted and used in practice (Crookston and Finley 2008). For these reasons, we decided to extend upon those combining the time series component and an optimization framework.

The method can also be adapted to a multiple imputation setting. However, while multiple imputation has been considered for several years to be the most accurate method for dealing with missing data (Rubin

1996), there is a tradeoff because single imputation is more interpretable. In particular, with single imputation we obtain one downstream predictive model that can be easily presented and explained to an entire clinical team, which is a critical step in the process of data-driven medical research (Shrive et al. 2006).

2.4 Learning α_d and h_d

In this section, we describe a custom tuning procedure to efficiently learn α_d and h_d, which are hyperparameters in the optimization problem (8). We run this custom tuning procedure as a pre-processing step before the `med.knn` algorithm, which allows us to learn these parameters without using cross-validation. This is a heuristic procedure which decouples the problem into multiple parts, first learning h_d for each covariate, and then learning α_d for each covariate. As a result, this custom tuning procedure is more computationally efficient and scales to larger problem sizes than cross-validation. In Section 4, we present the results from computational experiments comparing the speed and imputation accuracy of this custom tuning procedure against a traditional cross-validation method for selecting α_d and h_d.

In the first step of the custom tuning procedure, we learn the halflife parameter h_d for each covariate. As in cross-validation, we tune the halflife parameters over a discrete range of values, denoted as $\mathcal{H}$. For example, in the computational experiments, we set $\mathcal{H} = \{1, 7, 30, 90, 365, 1000\}$, representing halflife values of 1 day, 1 week, 1 month, etc. For each covariate d, we compute the leave-one-out error for each halflife value $h_d \in \mathcal{H}$. In particular, to compute the leave-one-out error for the halflife value h_d, first we derive the weights $\overline{C}_{ijd}$, then we impute the known values in covariate d using these weights, and finally we compute the sum-of-squared errors. Afterwards, we select the halflife parameter h_d which yields the lowest leave-one-out error.

For each continuous covariate $d \in \{1, \ldots, p_0\}$, the leave-one-out error is defined as:

$$\sum_{\{i:(i,d)\in\mathcal{N}_0\}} (x_{id} - \hat{w}_{id})^2, \tag{9}$$

where:

$$\hat{w}_{id} := \sum_{j=1}^{n} \overline{C}_{ijd} x_{jd}. \tag{10}$$

Here, $\hat{w}_{id}$ is equivalent to the MedImpute imputation of a continuous covariate x_{id} when $\alpha_d = 1$. For each categorical covariate $d \in \{p_0 + 1, \ldots, p_0 + p_1\}$, the leave-one-out error is defined as:

$$\sum_{\{i:(i,d)\in\mathcal{N}_1\}} \mathbb{1}_{\{x_{id} \neq \hat{v}_{id}\}}, \tag{11}$$

where:

$$\hat{v}_{id} := \arg\max_{v_{id}} \sum_{j=1}^{n} \overline{C}_{ijd} \mathbb{1}_{\{x_{jd}=v_{id}\}}. \tag{12}$$

Intuitively, $\hat{v}_{id}$ is the weighted mode of covariate d, where the weights are $\overline{C}_{ijd}$. This is equivalent to the MedImpute imputation of the categorical covariate x_{id} when $\alpha_d = 1$.

Note that we are able to learn h_d independently from α_d because the selection of $\overline{C}_{ijd}$ which minimizes the objective function (8) for any fixed value of α_d also minimizes the objective function for any choice of $\alpha_d \in [0, 1]$. Similarly, we can learn the halflife parameters $\{h_1, h_2, \ldots, h_p\}$ independently from one another, because the optimal choice of h_d which minimizes the objective function (8) does not depend upon the values of $\{h_1, \ldots, h_{d-1}, h_{d+1}, \ldots, h_p\}$. Therefore, in this custom tuning procedure, we take advantage of this fact, and tune each of the halflife parameters as an initial step.

In the second step of the custom tuning procedure, we learn the MedImpute weight parameter α_d for each covariate. As in cross-validation, we tune the MedImpute weight parameters over a discrete range of values, denoted as $\mathcal{A}$. For example, in the computational experiments, we set $\mathcal{A} = \{0, 0.05, \ldots, 0.95, 1.0\}$, denoting relative MedImpute weights of 0%, 5%, ..., 100%, respectively. For each covariate d, we compute the k-fold error for each MedImpute weight value $\alpha_d \in \mathcal{A}$. In particular, to compute the k-fold error for the MedImpute weight value α_d, first we split the dataset into k subsets (aka "folds"), next we impute each data subset using the rest of the subsets as training data, and finally we compute the total sum-of-squared errors across all of the folds. We select the MedImpute weight parameter α_d which yields the lowest k-fold error. For continuous covariates, the k-fold error is defined as:

$$\sum_{\ell=1}^{k} \sum_{\{i:(i,d)\in\mathcal{N}_0^\ell\}} (x_{id} - \hat{w}_{id}^\ell)^2, \tag{13}$$

where $\mathcal{N}_0^\ell$ are the known continuous values in the ℓth fold. The imputed values $\hat{w}_{id}^\ell$ are given by:

$$\hat{w}_{id}^\ell := (1-\alpha_d) w_{id}^{\mathrm{OPT}_\ell} + \alpha_d \sum_{\{i:(i,d)\in\mathcal{N}_0\setminus\mathcal{N}_0^\ell\}} \overline{C}_{ijd} x_{jd}, \tag{14}$$

where $w_{id}^{\mathrm{OPT}_\ell}$ is the OptImpute imputation of x_{id} using the data from the other $k-1$ folds, and $\mathcal{N}_0 \setminus \mathcal{N}_0^\ell$ are the known continuous values not in the ℓth fold. For categorical covariates, the k-fold error is defined as:

$$\sum_{\ell=1}^{k} \sum_{\{i:(i,d)\in\mathcal{N}_1^\ell\}} \mathbb{1}_{\{x_{id}\neq\hat{v}_{id}^\ell\}}, \tag{15}$$

where $\mathcal{N}_1^\ell$ are the known categorical values in the ℓth fold. The imputed values $\hat{v}_{id}^\ell$ are given by:

$$\hat{v}_{id}^\ell := \arg\max_{v_{id}} \left[(1-\alpha_d)\mathbb{1}_{\{v_{id}^{\mathrm{OPT}_\ell}=v_{id}\}} + \alpha_d \sum_{\{i:(i,d)\in\mathcal{N}_0\setminus\mathcal{N}_0^\ell\}} \overline{C}_{ijd}\mathbb{1}_{\{x_{jd}=v_{id}\}} \right]. \tag{16}$$

where $v_{id}^{\mathrm{OPT}_\ell}$ is the OptImpute imputation of x_{id} using the data from the other $k-1$ folds, and $\mathcal{N}_1 \setminus \mathcal{N}_1^\ell$ are the known categorical values not in the ℓth fold. Intuitively, $\hat{v}_{id}^\ell$ is the weighted mode of the OptImpute value and the other known values of the same covariate, where the weights are $(1-\alpha_d)$ and $\alpha_d \overline{C}_{ijd}$, respectively.

Finally, we note that there is another hyperparameter that we may tune for the `med.knn` algorithm, K, which is the number of nearest-neighbors. In the computational experiments, we fix $K = 10$, which works well for the datasets that we consider here. Previously, it has been shown that the OptImpute methods are relatively robust even if their hyperparameters are misspecified (Bertsimas et al. 2018b). Thus, while the accuracy of the `med.knn` algorithm can be improved slightly by tuning over K, the relative improvement in imputation accuracy is outweighed by the increased computational costs.

Algorithm 2 `med.knn`

Input: Incomplete data matrix $\mathbf{X}$,
warm start $[\mathbf{W}^0, \mathbf{V}^0]$,
max number of iterations $T \geq 0$,
weight parameters $\{\alpha_d\}_{d=1}^p$,
halflife parameters $\{h_d\}_{d=1}^p$.
Output: $\mathbf{X}^{imp}$ a full matrix with imputed values.
Procedure:
Initialize $t \leftarrow 1$, $\mathbf{W}^* \leftarrow \mathbf{W}^0$, $\mathbf{V}^* \leftarrow \mathbf{V}^0$.
while $t < T$ **do**
① Find the K nearest neighbors for each observation i, and update $\mathbf{Z}^*$ accordingly.
② Update the imputation $(\mathbf{W}^*, \mathbf{V}^*)$, following either Block Coordinate Descent or Coordinate Descent (details in Section 2.5).
③ Increment $t \leftarrow t + 1$.
end while
return $\mathbf{X}^{imp} \leftarrow [\mathbf{W}^*; \mathbf{V}^*]$.

2.5 The `med.knn` algorithm

In this section, we provide details for the updates in the `med.knn` imputation algorithm. This is a first-order method to find locally optimal solutions to Problem (5). As in the `opt.knn` algorithm, in this algorithm we alternatively update $\mathbf{Z}$ and $(\mathbf{W}, \mathbf{V})$ until the solution converges. The update for $\mathbf{Z}$ is identical to the one for `opt.knn`, and is computed with a simple sorting procedure on the distances. However, the update for $(\mathbf{W}, \mathbf{V})$ is modified and depends upon the MedImpute parameters α_d, C_{ijd}. As in `opt.knn`, we can update

the values of $(\mathbf{W}, \mathbf{V})$ either with Block Coordinate Descent (BCD) or Coordinate Descent (CD) which are described in the following subsections. The `opt.knn` updates for both BCD and CD are equivalent to the corresponding `med.knn` updates when $\alpha_d = 0$ for all $d = 1, \ldots, p$.

2.5.1 Block Coordinate Descent

In this approach, we update all of the imputed values at once. We call this approach BCD because we update the variables $(\mathbf{W}, \mathbf{V})$ as an entire block, keeping $\mathbf{Z}$ fixed. Our formulation Problem (8) decomposes by dimension into p_0 Quadratic Optimization problems for the continuous features and p_1 Mixed Integer Optimization problems for the categorical features. To update the imputed values $\mathbf{w}^d$ for continuous feature $d = 1, \ldots, p_0$, we solve:

$$\begin{aligned} \min_{\mathbf{w}^d} \quad & \sum_{i\in\mathcal{I}}\sum_{j=1}^{n} z_{ij}(1-\alpha_d)(w_{id}-w_{jd})^2 + \sum_{i\in\mathcal{I}}\sum_{j=1}^{n} \alpha_d \overline{C}_{ijd}(w_{id}-w_{jd})^2 \\ \text{s.t.} \quad & w_{id} = x_{id} \quad (i,d) \in \mathcal{N}_0. \end{aligned} \tag{17}$$

Taking the partial derivative of the objective function with respect to w_{id} for some missing entry $(i, d) \in \mathcal{M}_0$ and setting it to zero, we obtain after some simplifications:

$$\begin{aligned} 0 = & \left((1-\alpha_d)K + \alpha_d + \sum_{j\in\mathcal{I}}[(1-\alpha_d)z_{ji} + \alpha_d \overline{C}_{jid}]\right) w_{id} - \sum_{(j,d)\in\mathcal{M}_0} [(1-\alpha_d)(z_{ij}+z_{ji}) + \alpha_d(\overline{C}_{ijd}+\overline{C}_{jid})] w_{jd} \\ & - \sum_{(j,d)\in\mathcal{N}_0} [(1-\alpha_d)(z_{ij} + \mathbb{1}_{\{j\in\mathcal{I}\}} z_{ji}) + \alpha_d(\overline{C}_{ijd} + \mathbb{1}_{\{j\in\mathcal{I}\}}\overline{C}_{jid})] x_{jd}. \end{aligned} \tag{18}$$

This follows directly from equation (9) in (Bertsimas et al. 2018b). For each feature $d = 1, \ldots, p_0$, we have a system of equations of the above form which we can solve to determine the optimal imputed values $w_{id}, (i, d) \in \mathcal{M}_0$. Simplifying the notation, suppose that the missing values for the dimension d are $\widetilde{\mathbf{w}}^d := (w_{1d}, \ldots, w_{ad})$ and the known values are $\mathbf{x}^d := (x_{(a+1)d}, \ldots, x_{nd})$. Then the set of optimal imputed values $w^d_{id}, (i, d) \in \mathcal{M}_0$ is the solution to the linear system

$$((1-\alpha_d)\mathbf{Q} + \alpha_d \mathbf{P})\widetilde{\mathbf{w}}^d = ((1-\alpha_d)\mathbf{R} + \alpha_d \mathbf{Y})\mathbf{x}^d, \tag{19}$$

where the matrices $\mathbf{Q}$, $\mathbf{P}$, $\mathbf{R}$, and $\mathbf{Y}$ are defined as

$$\mathbf{Q} = \begin{bmatrix} K + \sum_{j\in\mathcal{I}} z_{j1} - 2z_{11} & -z_{12} - z_{21} & \ldots & -z_{1a} - z_{a1} \\ -z_{21} - z_{12} & K + \sum_{j\in\mathcal{I}} z_{j2} - 2z_{22} & \ldots & -z_{2a} - z_{a2} \\ \vdots & \vdots & \ddots & \vdots \\ -z_{a1} - z_{1a} & -z_{a2} - z_{2a} & \ldots & K + \sum_{j\in\mathcal{I}} z_{ja} - 2z_{aa} \end{bmatrix}, \tag{20}$$

$$\mathbf{P} = \begin{bmatrix} \sum_{j\in\mathcal{I}} \overline{C}_{j1d} - 2\overline{C}_{11d} & -\overline{C}_{12d} - \overline{C}_{21d} & \ldots & -\overline{C}_{1ad} - \overline{C}_{a1d} \\ -\overline{C}_{21d} - \overline{C}_{12d} & \sum_{j\in\mathcal{I}} \overline{C}_{j2d} - 2\overline{C}_{22d} & \ldots & -\overline{C}_{2ad} - \overline{C}_{a2d} \\ \vdots & \vdots & \ddots & \vdots \\ -\overline{C}_{a1d} - \overline{C}_{1ad} & -\overline{C}_{a2d} - \overline{C}_{2ad} & \ldots & \sum_{j\in\mathcal{I}} \overline{C}_{jad} - 2\overline{C}_{aad} \end{bmatrix}, \tag{21}$$

$$\mathbf{R} = \begin{bmatrix} z_{1(a+1)} + \mathbb{1}_{\{(a+1)\in\mathcal{I}\}} z_{(a+1)1} & \ldots & z_{1n} + \mathbb{1}_{\{n\in\mathcal{I}\}} z_{n1} \\ \vdots & & \vdots \\ z_{a(a+1)} + \mathbb{1}_{\{(a+1)\in\mathcal{I}\}} z_{(a+1)a} & \ldots & z_{an} + \mathbb{1}_{\{n\in\mathcal{I}\}} z_{na} \end{bmatrix}, \tag{22}$$

$$\mathbf{Y} = \begin{bmatrix} \overline{C}_{1(a+1)d} + \mathbb{1}_{\{(a+1)\in\mathcal{I}\}} \overline{C}_{(a+1)1d} & \ldots & \overline{C}_{1nd} + \mathbb{1}_{\{n\in\mathcal{I}\}} \overline{C}_{n1d} \\ \vdots & & \vdots \\ \overline{C}_{a(a+1)d} + \mathbb{1}_{\{(a+1)\in\mathcal{I}\}} \overline{C}_{(a+1)ad} & \ldots & \overline{C}_{and} + \mathbb{1}_{\{n\in\mathcal{I}\}} \overline{C}_{nad} \end{bmatrix}. \tag{23}$$

Without loss of generality, there exists a closed-form solution

$$\widetilde{\mathbf{w}}^d = ((1-\alpha_d)\mathbf{Q} + \alpha_d \mathbf{P})^{-1}((1-\alpha_d)\mathbf{R} + \alpha_d \mathbf{Y})\mathbf{x}^d \tag{24}$$

to this system of equations for each feature $d = 1, \ldots, p_0$. To update the imputed values $\mathbf{v}^d$ for each categorical feature $d = (p_0+1), \ldots, p$, we solve the following mixed-integer optimization problem:

$$\begin{aligned} \min_{\mathbf{v}^d} \quad & \sum_{i\in\mathcal{I}}\sum_{j=1}^{n}((1-\alpha_d)z_{ij} + \alpha_d \overline{C}_{ijd})y_{ij} \\ \text{s.t.} \quad & v_{id} = x_{id} && (i,d) \in \mathcal{N}_1, \\ & v_{id} - v_{jd} \le y_{ij}k_d && i = 1,\ldots,n, j = 1,\ldots,n, \\ & v_{jd} - v_{id} \le y_{ij}k_d && i = 1,\ldots,n, j = 1,\ldots,n, \\ & y_{ij} \in \{0,1\}^{|\mathcal{I}|\times n}. \end{aligned} \tag{25}$$

This is a Mixed Integer Optimization problem, which is practically solvable as the BCD update for `opt.knn`. Since the BCD update step requires inverting a matrix with $O(n^2)$ entries and solving an optimization problem with $O(n^2)$ binary variables, this method works best for smaller problem sizes $n \le 10,000$.

2.5.2 Coordinate Descent

In CD, we update the imputed values one at a time. In order to update the imputed value for x_{id}, we fix all of the variables in Problem (8) except for w_{id} or v_{id} and solve the corresponding one-dimensional optimization problem. This results in fast, closed-form updates for both the continuous and categorical variables. Each $w_{id}, (i,d) \in \mathcal{M}_0$ is imputed as the minimizer of the following:

$$\min_{w_{id}} \quad \sum_{r\in\mathcal{I}}\sum_{j=1}^{n} z_{rj}\sum_{d=1}^{p_0}(1-\alpha_d)(w_{rd}-w_{jd})^2 + \sum_{r\in\mathcal{I}}\sum_{j=1}^{n}\sum_{d=1}^{p_0}\alpha_d\overline{C}_{rjd}(w_{rd}-w_{jd})^2. \tag{26}$$

Solving the above gives the closed-form solution for every $(i,d) \in \mathcal{M}_0$:

$$w_{id} = \frac{\sum_{j=1}^{n}((1-\alpha_d)z_{ij} + \alpha_d\overline{C}_{ijd})w_{jd} + \sum_{j\in\mathcal{I}}((1-\alpha_d)z_{ji} + \alpha_d\overline{C}_{jid})}{K + \sum_{j=1}^{n}\alpha_d\overline{C}_{ijd} + \sum_{j\in\mathcal{I}}((1-\alpha_d)z_{ji} + \alpha_d\overline{C}_{jid})}. \tag{27}$$

Similarly, each categorical variable $v_{id}, (i,d) \in \mathcal{M}_1$ is imputed as the minimizer of the following:

$$\min_{v_{id}} \quad \sum_{r\in\mathcal{I}}\sum_{j=1}^{n} z_{rj}\sum_{d=p_0+1}^{p_0+p_1}(1-\alpha_d)\mathbb{1}_{\{v_{rd}\neq v_{jd}\}} + \sum_{r\in\mathcal{I}}\sum_{j=1}^{n}\sum_{d=p_0+1}^{p_0+p_1}\alpha_d\overline{C}_{rjd}\mathbb{1}_{\{v_{rd}\neq v_{jd}\}}. \tag{28}$$

Suppose that the value of categorical variable v_{id} is one of k_d distinct categories $\{1, 2, \ldots, k_d\}$. Then, the solution to problem (28) is

$$\arg\max_{k\in\{1,\ldots,k_d\}}\left[\sum_{j=1}^{n}\left((1-\alpha_d)z_{ij} + \alpha_d\overline{C}_{ijd}\right)\mathbb{1}_{\{v_{jd}=k\}} + \sum_{j\in\mathcal{I}}\left((1-\alpha_d)z_{ji} + \alpha_d\overline{C}_{ijd}\right)\mathbb{1}_{\{v_{jd}=k\}}\right]. \tag{29}$$

Here, we set the imputed variable to be the value with the highest frequency in the neighborhood, with instances of the same person i receiving additional weight calibrated by the parameters $\{\overline{C}_{ijd}\}_{j=1}^{n}$ and α_d.

This approach scales to large problem sizes (n in the 100,000's), and it is the method that we implement for the computational experiments.

3 Computational Experiments on Real-World Clinical Datasets

In this section, we run a series of computational experiments testing the performance of `med.knn` imputing missing values in real-world clinical datasets. In Section 3.1, we provide an overview of the three datasets and their baseline characteristics. In Section 3.2, we describe the mechanisms for generating Missing Not at Random (MNAR) data that are used in some of the experiments. In Section 3.3, we describe the setup of the computational experiments, and we describe the imputation methods that we run for comparison across all of the computational experiments. In Section 3.4, we report the results of the experiments on the imputation tasks. In Section 3.5, we report the results of the experiments on the downstream predictive tasks. In Section 3.6 we discuss the results and major takeaways from the computational experiments.

3.1 Description of Real-World Clinical Datasets

In this section we describe the three real-world clinical datasets used in the computational experiments. In Section 3.1.1, we describe the FHS dataset. In Section 3.1.2, we describe the DFCI dataset. Finally, in Section 3.1.3, we describe the PPMI dataset.

3.1.1 Framingham Heart Study (FHS) dataset

The FHS was started in 1948 with the goal of observing a large population of healthy adults over time to better understand the factors that lead to cardiovascular disease. Over 80 variables were collected from 5,209 people at a time for more than 40 years. The FHS is arguably the most influential longitudinal study in the field of cardiovascular and cerebrovascular research. This data has now been used in more than 2,400 studies and is considered one of the top 10 cardiology advances of the twentieth century alongside the electrocardiogram and open-heart surgery (Daniel Levy 2006).

In our computational experiments, we consider all individuals from the FHS Original Cohort (National Heart, Lung, and Blood Institute, Boston University 2012) with 10 or more observations, which includes $M = 1,107$ unique patients. For each patient, we take the 10 most recent observations, so the dataset has $n = 11,070$ observations total. We include $p = 13$ continuous (Age, Body Mass Index, Systolic Blood Pressure, High-Density Lipoproteins, Hematocrit, Blood Glucose levels) and categorical covariates (Gender, Smoking, presence of Cardiovascular Disease, presence of Atrial Fibrillation, presence of diabetes, currently under prescription of antihypertensive medication, presence of Left Ventricular Hypertrophy from ECG results).

Overall, there are 12.56% missing values in the FHS dataset. The percentage of missing values in each covariate is shown in Table 5 in Appendix 7.1. Due to the design of the longitudinal study, the 10 observations for each patient occur at regular intervals spaced 2 years apart, for a total span of 18 years. For the imputation tasks, we add in additional missing values to the FHS dataset, and evaluate the accuracy of `med.knn` and comparison methods against the ground-truth values. For the downstream tasks, we evaluate classification models which predict 10-year risk of stroke given the imputed training data.

3.1.2 Dana Farber Cancer Institute (DFCI) dataset

The DFCI dataset was obtained from a recently published work on predicting mortality in late-stage cancer patients (Bertsimas et al. 2018a). In this study, the authors retrospectively obtained patient data from EHR and linked Social Security Administration mortality data for cancer patients at the Dana Farber Cancer Institute / Brigham and Women's Cancer Center from 2004 through 2014. Predictive models were fit for the entire population and individual cancers, including breast, lung, colorectal, kidney, and prostate cancer. Study eligibility required adult patients that have received at least one anticancer treatment over the course of their care, including chemotherapy, immunotherapy, and targeted therapy.

In our computational experiments, we consider all patients with late-stage breast cancer from the DFCI dataset. Each observation corresponds to a patient initiating an anticancer regimen which was systematically recorded in the hospital's database. As a result, for every patient who followed more than one regimen, multiple observations were collected. For each patient, we include all of their observations in either the training set or testing set, respectively. In total, we have 12,206 observations that correspond to 5,987 unique patients. This includes 3,228 individuals who have just one line of therapy and therefore only appear once in this dataset. For each observation, there are 106 covariates which describe the patient at that point in time, including demographics, lab tests, vital signs, current medications, medical history, biomarkers, and variables derived from the patient's temporal EHR history.

Overall, there are 10.79% missing values in the DFCI dataset. The percentage of missing values in each covariate is shown in Table 6 in Appendix 7.1. Due to the nature of this observational study, the observations for each patient occur at irregular intervals, which correspond to hospital visits. In addition, in the dataset each patient has anywhere from 1 to 12 observations. In Appendix 7.1, we provide some more details on the DFCI dataset, including the distribution of observations per patient (see Figure 13) and summary statistics of the time intervals between each visit (see Table 8). For the imputation tasks, we add in additional missing values to the DFCI dataset, and evaluate the accuracy of `med.knn` and comparison methods against the ground-truth values. For the downstream tasks, we evaluate classification models which predict 60-day risk of mortality given the imputed training data.

3.1.3 Parkinson's Progression Markers Initiative (PPMI) dataset

The PPMI (Marek et al. 2011) was a landmark observational clinical study with the aim to comprehensively evaluate patient cohorts using imaging, biologic sampling as well as clinical and behavioral data to identify biomarkers of Parkinson's disease progression.

In our computational experiments, we consider data from the PPMI baseline examination as well as the following three years of follow-up. In this longitudinal study, 20 patients appeared only in one follow-up examination, 33 in two while the rest of the population participated in all 352 clinical evaluations. As a result, in total we have 1,547 observations corresponding to 405 distinct patients. For each observation, there are 116 covariates which describe the demographic characteristics, the results of behavioral tests, clinical test results, as well as the presence or absence of genetic mutations related to the disease.

Overall, there are 2.61% missing values in the PPMI dataset. The percentage of missing values in each covariate is shown in Table 7 in Appendix 7.1. Due to the design of the longitudinal study, the 4 observations for each patient occur at regular intervals spaced 1 year apart, for a total span of 4 years. For the imputation tasks, we add in additional missing values to the PPMI dataset, and evaluate the accuracy of `med.knn` and comparison methods against the ground-truth values. For the downstream tasks, we evaluate regression models which predict the Montreal Cognitive Assessment (MoCA) score one year in advance. The MoCA score is a rapid screening instrument for mild cognitive dysfunction, a clinical state that often progresses to dementia (Nasreddine et al. 2005).

3.2 Mechanisms for Generating Missing Not at Random (MNAR) data

Missing data can either be Missing Completely At Random (MCAR), Missing At Random (MAR), or Missing Not At Random (MNAR) (Little and Rubin 2019). The type of missingness can be determined through an understanding of the specific feature and what systematic biases may exist in its collection process. Different types of missingness must be treated differently for meaningful analysis. In reality, missing data are most commonly associated with the MNAR category where the presence of unknown values is systematically related to unobserved factors.

In this section, we describe mechanisms for generating Missing Not at Random (MNAR) data for our computational experiments. We consider different mechanisms for the longitudinal and EHR datasets corresponding to the different patterns of missing data which are typically observed in real-world datasets. In section 3.2.1, we describe the missing data mechanism that we use for the MNAR experiments on the two longitudinal datasets: FHS and PPMI. In section 3.2.2, we describe the missing data mechanism that we use for the MNAR experiments on the EHR dataset: DFCI.

For all MNAR experiments, the total percentage of missing data is fixed to 30%. For each individual experiment, we assume that the dataset is (γ30% MNAR, $(1-\gamma)$30% MCAR), where γ is a constant that we select between 0 and 1. To generate the missing data patterns, first we generate the γ30% MNAR patterns, and then we randomly select an additional $(1-\gamma)$30% subset of the data to be Missing Completely at Random (MCAR). In the following two sections, we describe the specific ways that we generate MNAR data for longitudinal studies and EHR data, which are influenced by real-world missing data mechanisms.

3.2.1 MNAR Mechanism for Data from Longitudinal Studies

In longitudinal studies, missing data patterns often result from changes in the experiment design. Researchers may decide to include an additional set of variables as the study progresses over time due to new information from other investigations. Thus, it is common for feature d to be missing for the first t_d rounds of long-term longitudinal studies. For example, ECG results were only first recorded in the FHS study 14 years after the study began (D'Agostino et al. 2013; Mahmood et al. 2014).

To generate γ30% MNAR patterns under this mechanism, we use the following process. First, we randomly select a covariate d and a discrete uniform random variable $t_d \in \{1, 2, \ldots, N\}$, where $N = 10$ for the FHS dataset and $N = 4$ for the PPMI dataset. The value t_d corresponds to the last round of the longitudinal study that covariate d is missing. For example, if $t_d = 2$ for the covariate "Left Ventricular Hypertrophy" (LVH), then the value for LVC will be missing for all observations in the two first clinical examinations. We continue this process until we have introduced γ30% MNAR missing values. Afterwards, we introduce additional MCAR missing values to the remaining dataset in order to obtain the final dataset with 30% missing values.

3.2.2 MNAR Mechanism for Data from EHR

In EHR data, missing data patterns may be correlated with the severity of patient's condition. Consider the case of a patient whose physician suspects the existence of chronic kidney disease. The associated record is more likely to have a recorded value for Glomerular Filtration Rate since it is a direct indication of the kidney's functional status (Levey et al. 2005). Therefore, observed values are more likely to be below the threshold of 60mL/min/1.73 m^2 since they correspond to sicker patients.

To generate $\gamma 30\%$ MNAR patterns under this mechanism, we suppose that missing indicators are independent Bernoulli random variables where the probability that entry x_{id} is missing equals the probability that a normal random variable $N(x_{id}, \epsilon)$ is greater than a particular threshold for covariate d. The threshold for each covariate d is the quantile of $\mathbf{X}^d$ which corresponds to the desired missing percentage level $\gamma 30\%$. Then, we introduce additional MCAR missing values to the remaining dataset in order to obtain the final dataset with 30% missing values total for this experiment.

3.3 Experimental Setup

In this section, we describe the setup of computational experiments that compare `med.knn` to other state-of-the-art imputation methods. We use data from three distinct sources to test the performance of our algorithm on both longitudinal cohort study and EHR datasets. The codebase for the computational experiments is publicly available at `https://github.com/colin78/medimpute_computational_experiments`.

In our experiments, we take the full dataset to be the ground truth. First, we normalize the data so that each continuous covariate has mean zero and standard deviation equal to one. Then, we run some of the most commonly-used and state-of-the-art methods for imputation to predict the missing values and compare against `med.knn`. The methods that we compare are as follows:

1. **Mean** (`mean`): This is the simplest method. For each continuous feature, we impute the mean of the observed values and, for each categorical feature, we impute the mode of the observed values (Little and Rubin 2019).
2. **Moving Average** (`moving.avg`): This method takes into account only observations of the same entity (i.e., patient) and imputes their averages under a given time window. In cases where only one observation per entity is available, the method reduces to the `mean`. For each dataset, we consider a different time horizon depending on the relative scale of the data (i.e, years, months, or days). Implemented in the Julia programming language.
3. **Bayesian Principal Component Analysis** (`bpca`): This method takes a singular value decomposition (SVD) of the data matrix and information from a Bayesian prior distribution on the model parameters to impute missing values (Oba et al. 2003). Implemented using the `pcaMethods` package in the R programming language.
4. **Multivariate Imputation via Chained Equations** (`mice`): In this multiple imputation method, we begin from m random starts and iteratively update each one to produce m independent imputations. In each iteration, we update the imputed values in feature d by drawing from a distribution conditional on all other features (van Buuren and Groothuis-Oudshoorn 2011). We use Classification Trees for the categorical features and Regression Trees for the continuous features. Implemented using the `mice` package in in the R programming language.
5. **Multiple Imputation with Boostrap Expectation Maximization (`Amelia II`):** This is another multiple imputation method that builds upon the `Amelia I` framework, which assumes that the data is jointly distributed as multivariate normal and uses an expectation-maximization (EM) algorithm with bootstrapping (Honaker et al. 2011; King et al. 2001). In addition, a newer version of the method allows for the imputation of cross-sectional time series data. It can build a general model of patterns within variables across time by creating a sequence of polynomials of the time index. Thus, it is able to capture variables that are recorded over time within a cross-sectional unit and are observed to vary smoothly over time. Implemented using the `amelia` package in the R programming language.
6. **OptImpute under K-NN Objective** (`opt.knn`): This method finds a high quality solution to Problem (5) minimizing the sum of distances from each point to its K-Nearest Neighbors (Bertsimas et al. 2018b). We find solutions to this problem using Algorithm 1 with the CD update. Fixing $K = 10$, we use several warm and random restarts and select the imputation with the best objective value. Implemented using the `OptImpute` package in the Julia programming language.
7. **MedImpute under K-NN Objective** (`med.knn`): This method finds a high quality solution to Problem (8) minimizing the sum of distances from each point to its K-Nearest Neighbors and other instances

of the same individual. We find solutions to this problem using Algorithm 2 with the CD update. For each feature d, we perform cross-validation to tune the parameters α_d, h_d with the rest of the MedImpute parameters set equal to zero. Fixing $K = 10$, we use several warm and random restarts and select the imputation with the best objective value. Implemented in the Julia programming language.

For each experiment, we evaluate the imputation accuracy of each method using the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) metrics, which are extended to accommodate both continuous and categorical covariates. Let $\mathcal{M}_0^{test}$, $\mathcal{M}_1^{test}$ be the hold-out sets for the missing continuous and categorical covariates, respectively. We define the MAE and RMSE metrics to be:

$$\text{MAE} := \frac{1}{|\mathcal{M}_0^{test}|} \sum_{(i,d)\in\mathcal{M}_0^{test}} |w_{id} - x_{id}| + \frac{1}{|\mathcal{M}_1^{test}|} \sum_{(i,d)\in\mathcal{M}_1^{test}} \mathbb{1}_{\{v_{id}\neq x_{id}\}}, \tag{30}$$

$$\text{RMSE} := \sqrt{\frac{1}{|\mathcal{M}_0^{test}|} \sum_{(i,d)\in\mathcal{M}_0^{test}} (w_{id} - x_{id})^2 + \frac{1}{|\mathcal{M}_1^{test}|} \sum_{(i,d)\in\mathcal{M}_1^{test}} \mathbb{1}_{\{v_{id}\neq x_{id}\}}}. \tag{31}$$

In addition to comparing the accuracy of each method on the imputation task, we also compare their performance on downstream predictive tasks which are tailored for each dataset. In these experiments, we use the imputation methods to fill in the missing values of the datasets, and then we train machine learning models with the data from completed datasets. By comparing the accuracy of the predictive models on the downstream tasks, we can see the relative impact of using one imputation method versus another in a machine learning pipeline. For the FHS dataset, the downstream task is to predict 10-year risk of stroke, a classification task. For the DFCI dataset, the downstream task is to predict 60-day risk of mortality, which is also a classification task. For the PPMI dataset, the downstream task is to predict the Montreal Cognitive Assessment (MoCA) score for next year, which is a regression task.

To evaluate the accuracy on the downstream predictive task, first we split the patients from the completed dataset into a training and testing set using a 75%/25% ratio. For the longitudinal datasets (FHS and PPMI) we include only one visit per patient, the most recent one. Thus, the time series component of the dataset is only present in the missing data imputation process but not in the supervised learning part of the experiment. This setup allows us to quantify the relative benefit of `med.knn` per individual. For the EHR dataset (DFCI), we include all of the observations from each patient in either the training or testing set for the supervised learning task.

Next, we train predictive models on the training set and report the out-of-sample accuracy on the testing set. For the classification tasks, we train ℓ_1-regularized logistic regression models and report the out-of-sample Area Under the Receiver Operator Characteristic Curve (AUC). For the regression task, we train ℓ_1-regularized linear regression models and report the out-of-sample Mean Absolute Error (MAE). These two metrics are commonly used evaluation criteria in machine learning (Hastie et al. 2009). We repeat all experiments for 25 random seeds and average the results. Each iteration corresponds to a different random split of the patients into the training and testing sets, a random warmstart, and a randomly generated missing data pattern. In particular, we note that the patient IDs and the time stamps corresponding to each row of the dataset are maintained across the different random seeds, so that the temporal sequence of the records remains the same as the original dataset.

We artificially created missing data under different mechanisms and random patterns to compare the imputation accuracy of the proposed method. The missing data generation process was independently applied to each column. For a fixed missing percentage $f\%$, we remove the necessary number of known values for each feature to reach the $f\%$ target. The patient ID y_i was not factored in the missing data generation process and all rows were considered independent observations. If the existing percent of missing data for a column was higher than the target $f\%$, we do not generate any artificial missing values for the covariate, and thus the feature does not contribute to the estimation of the imputation accuracy metrics.

Given this framework for evaluating imputation methods on both imputation and downstream tasks, we conduct a variety of experiments which vary the pattern of the missing data. In particular, we conduct three different types of experiments that correspond to variations in the form of missing data that we frequently encounter in medical datasets:

1. **Percentage of Missing Data:** We generate patterns of missing data for various percentages ranging from 10% to 50% under the missing completely at random (MCAR) mechanism. Given a target proportion of missing data f (i.e., $f = 20\%$), we generate among all observed data f missing values at each column independently from the rest completely at random.

2. **Number of Observations Per Patient:** With the missing percentage fixed at 50% MCAR, we vary the time frame during which patient observations are included in the imputation task. Our goal is to quantify the effect of the time series component as we vary its intensity.
3. **Mechanism of Missing Data:** With the missing percentage fixed at 30%, we vary the missing data mechanism from Missing Completely At Random (MCAR) to Missing Not At Random (MNAR) on a gradient scale. In particular, we suppose that the missing pattern is (γ30% MNAR, $(1-\gamma)$30% MCAR), where γ varies from 0 to 1. We consider two different MNAR mechanisms that correspond to distinct missing data patterns observed in longitudinal studies and EHR.

The objective of the first set of experiments is to determine which imputation methods perform best at high and low levels of missing data. For these experiments, we also report the results from statistical hypothesis tests (Friedman Rank and pairwise t-tests) to evaluate whether the rankings and differences between the imputation algorithms are statistically significant. The objective of the second set of experiments is to determine how the performance of `med.knn` and other imputation methods varies as the amount of time series information available on each patient fluctuates. Finally, the objective of the third set of experiments is to determine how robust each imputation method is with respect to the missing data mechanism. In the previous section, we describe the two mechanisms for generating MNAR data for the third set of experiments. Below, we summarize all of the steps required to run one of the computational experiments for a single random seed:

1. Fix a random seed s, a dataset, a desired missingness percentage level f%, a missing data imputation method, and a value for the γ parameter.
2. Generate a random missing data pattern in the given dataset using the targeted percentage of missing values f%, the random seed s, and the value of the γ parameter.
3. Impute the missing values in the provided dataset using the specified algorithm (i.e. `med.knn`, `mean`, `bpca`).
4. Calculate the imputation error using the MAE and RMSE metrics (see Equations 30-31) on the artificially generated missing data.
5. Split the patients in the dataset into a training and testing set using a 75%/25% ratio. For the longitudinal datasets, only include the most recent observation from each individual in the training and testing sets. For the EHR (DFCI) dataset, include all of the observations from each individual in the training or testing set.
6. Train a downstream predictive model on the training set using the `cv.glmnet` function from the R `glmnet` package (Friedman et al. 2009). For the FHS and DFCI datasets which have binary outcomes variables, train a logistic regression model with l_1 regularization. For the PPMI dataset which has a continuous outcome variable, train a linear regression model with l_1 regularization.
7. Report the out-of-sample performance of the trained model on the testing set. For the classification tasks, report the out-of-sample AUC, and for the regression task, report the out-of-sample MAE.

3.4 Imputation Results

In this section, we provide the results from all experiments on the imputation tasks. In particular, we present the imputation results from the 1) Percentage of Missing Data, 2) Number of Observations Per Patient, and 3) Mechanism of Missing Data experiments.

Percentage of Missing Data In Figure 1, we show the MAE imputation accuracy results from the first set of experiments in which we vary the percentage of missing data from 10% to 50%, and the missing data mechanism is fixed to MCAR. We present the exact values and standard errors in this plot in the Appendix in Table 9. Across all of the datasets, `med.knn` achieves the lowest average MAE for all of the missing percentages tested. On the FHS longitudinal dataset with 50% MCAR data, `med.knn` has an average MAE of 0.289 compared to the next best method `opt.knn` with an average MAE of 0.503, a 42.54% reduction. Similarly, on the PPMI longitudinal dataset with 50% MCAR data, `med.knn` has an average MAE of 1.286 compared to the next best method `opt.knn` with an average MAE of 1.99, a 35.37% reduction. On the DFCI dataset with 50% MCAR data, `med.knn` has an average MAE of 3.568 compared to the next best method `mean` with an average MAE of 4.367, a 22.39% reduction.

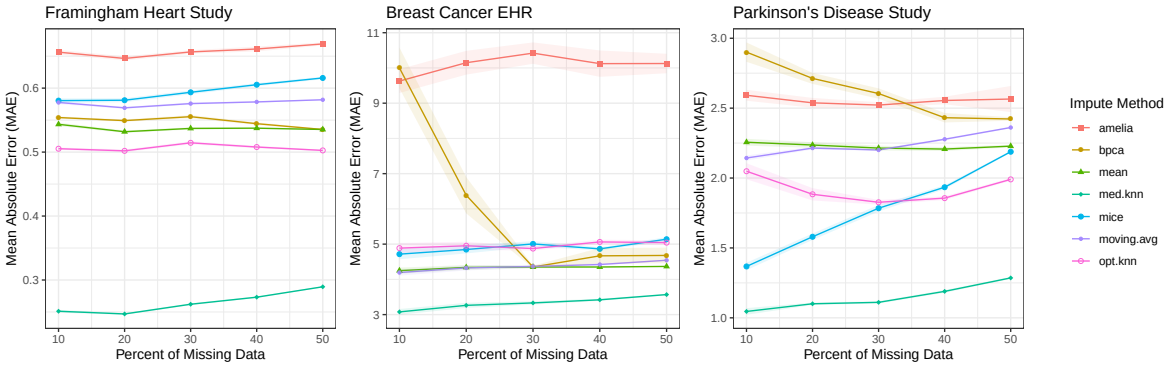

Fig. 1: Imputation errors for each method using the MAE metric on the FHS, DFCI, and PPMI datasets, varying the percentage of missing data from 10% to 50%. The missing data mechanism is fixed to MCAR.

In Figure 2, we present the RMSE imputation accuracy results. In general, the results are similar to the MAE imputation accuracy results, and `med.knn` produces the imputation with the lowest RMSE across all experiments. One notable difference is on the DFCI dataset, the relative improvement of `med.knn` compared to `bpca`, `moving.avg`, and `mean` is much smaller. Because the `mean` imputation method performs relatively well, this suggests that there are some difficult-to-impute covariates in the DFCI dataset which are resulting in large RMSE values for all of the more complex methods.

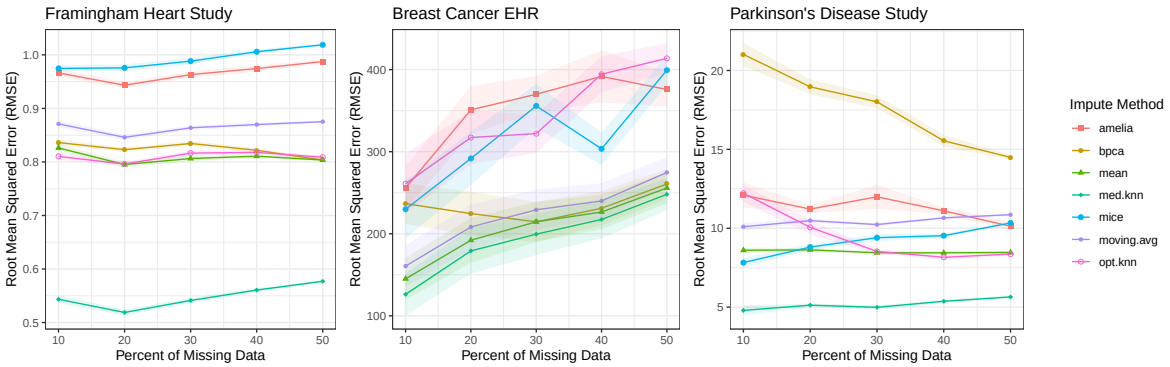

Fig. 2: Imputation errors for each method using the RMSE metric on the FHS, DFCI, and PPMI datasets, varying the percentage of missing data from 10% to 50%. The missing data mechanism is fixed to MCAR.

In Table 1, we present the results from the Friedman Rank test for each of the Missing Data imputation experiments. In this statistical test, we compare the relative rank of `med.knn` against the relative ranks of the comparator methods for each of the 25 random seeds. These results demonstrate that the `med.knn` method is consistently ranked higher than the others across each of the experiments.

In Table 2, we present the results from the pairwise t-test for each of the experiments. In this statistical test, we evaluate the differences in MAE between `med.knn` and each of the comparison methods. In all of the experiments, we observe that the differences in MAE are statistically significant with p-values less than 0.001. In most cases, we observe that the relative improvement of `med.knn` decreases as the percentage of missing data increases. This is because the comparator methods perform similarly across all levels of missing data from 10-50%, while the `med.knn` performs best at the lowest missing percentages. One exception is `mice` on the PPMI dataset, which declines in performance rapidly as the percentage of missing data increases. Another exception is the `bpca` method, which surprisingly improves in performance as the percentage of missing data increases for the DFCI and PPMI datasets. One explanation for these results could be that `bpca` is overfitting on the datasets which have few missing values.

In the Appendix, we present the MedImpute hyperparameters which were selected in Missing Percentage experiments for the FHS dataset. In Table 15, we show the median halflife parameters that were selected for each covariate at each missing percentage. We observe that most of the halflife parameters are consistent

χ^2 statistic (adjusted p-value)				χ^2 statistic (adjusted p-value)			
%	**FHS**	**DFCI**	**PPMI**	**%**	**FHS**	**DFCI**	**PPMI**
10	130 (<0.001***)	210 (<0.001***)	75 (<0.001***)	10	130 (<0.001***)	210 (<0.001***)	75 (<0.001***)
20	130 (<0.001***)	220 (<0.001***)	53 (<0.001***)	20	130 (<0.001***)	220 (<0.001***)	53 (<0.001***)
30	130 (<0.001***)	260 (<0.001***)	74 (<0.001***)	30	130 (<0.001***)	260 (<0.001***)	74 (<0.001***)
40	110 (<0.001***)	230 (<0.001***)	58 (<0.001***)	40	110 (<0.001***)	230 (<0.001***)	58 (<0.001***)
50	140 (<0.001***)	270 (<0.001***)	71 (<0.001***)	50	140 (<0.001***)	270 (<0.001***)	71 (<0.001***)

(a) MAE (b) RMSE

Table 1: The Friedman Rank test results for the imputation tasks varying the percentage of missing data from 10-50% MCAR, using either the MAE or RMSE metric for comparison. Each table shows the value of Friedman's Chi-squared statistic and p-value for the hypothesis test comparing `med.knn` against the benchmark methods for each experiment.

FHS

	Δ MAE (adjusted p-value)					
Missing %	`mice`	`moving.avg`	`amelia`	`bpca`	`mean`	`opt.knn`
10	-0.33 (<0.001***)	-0.33 (<0.001***)	-0.41 (<0.001***)	-0.30 (<0.001***)	-0.29 (<0.001***)	-0.25 (<0.001***)
20	-0.33 (<0.001***)	-0.32 (<0.001***)	-0.40 (<0.001***)	-0.30 (<0.001***)	-0.28 (<0.001***)	-0.26 (<0.001***)
30	-0.33 (<0.001***)	-0.31 (<0.001***)	-0.39 (<0.001***)	-0.29 (<0.001***)	-0.27 (<0.001***)	-0.25 (<0.001***)
40	-0.33 (<0.001***)	-0.31 (<0.001***)	-0.39 (<0.001***)	-0.27 (<0.001***)	-0.26 (<0.001***)	-0.23 (<0.001***)
50	-0.33 (<0.001***)	-0.29 (<0.001***)	-0.38 (<0.001***)	-0.25 (<0.001***)	-0.25 (<0.001***)	-0.21 (<0.001***)

DFCI

	Δ MAE (adjusted p-value)					
Missing %	`mice`	`amelia`	`moving.avg`	`bpca`	`mean`	`opt.knn`
10	-1.64 (<0.001***)	-6.55 (<0.001***)	-1.92 (<0.001***)	-6.92 (<0.001***)	-1.17 (<0.001***)	-1.81 (<0.001***)
20	-1.58 (<0.001***)	-6.89 (<0.001***)	-1.86 (<0.001***)	-3.12 (<0.001***)	-1.08 (<0.001***)	-1.69 (<0.001***)
30	-1.67 (<0.001***)	-7.09 (<0.001***)	-1.84 (<0.001***)	-1.02 (<0.001***)	-1.02 (<0.001***)	-1.54 (<0.001***)
40	-1.46 (<0.001***)	-6.71 (<0.001***)	-1.81 (<0.001***)	-1.26 (<0.001***)	-0.93 (<0.001***)	-1.62 (<0.001***)
50	-1.57 (<0.001***)	-6.56 (<0.001***)	-1.77 (<0.001***)	-1.11 (<0.001***)	-0.80 (<0.001***)	-1.48 (<0.001***)

PPMI

	Δ MAE (adjusted p-value)					
Missing %	`mice`	`amelia`	`moving.avg`	`bpca`	`mean`	`opt.knn`
10	-0.32 (<0.001***)	-1.55 (<0.001***)	-1.10 (<0.001***)	-1.86 (<0.001***)	-1.21 (<0.001***)	-1.00 (<0.001***)
20	-0.48 (<0.001***)	-1.44 (<0.001***)	-1.10 (<0.001***)	-1.61 (<0.001***)	-1.14 (<0.001***)	-0.78 (<0.001***)
30	-0.67 (<0.001***)	-1.36 (<0.001***)	-1.09 (<0.001***)	-1.49 (<0.001***)	-1.10 (<0.001***)	-0.72 (<0.001***)
40	-0.75 (<0.001***)	-1.37 (<0.001***)	-1.08 (<0.001***)	-1.24 (<0.001***)	-1.02 (<0.001***)	-0.67 (<0.001***)
50	-0.90 (<0.001***)	-1.40 (<0.001***)	-1.07 (<0.001***)	-1.14 (<0.001***)	-0.94 (<0.001***)	-0.70 (<0.001***)

Table 2: Pairwise t-tests between `med.knn` and benchmark methods for imputation tasks varying the percentage of missing data from 10-50% MCAR, using the MAE metric for comparison. The p-values are adjusted for multiple comparisons.

across different levels of missing data, and for many of the covariates the highest halflife parameter of 1000 days was selected. This suggests that for these covariates, a measurement from 1000 days ago may be used to significantly inform the measurement for the same patient today. In addition, we may be able to improve the performance of this method by considering even longer halflife values. In Table 16, we show the median alpha parameters that were selected for each covariate at each missing percentage, from the validation. In all cases, the alpha parameter is at least 0.5, and in many cases equals 1. This suggests that for these covariates, the time series part of the objective function is more important for the imputation than the K-nearest neighbors part of the objective function. In addition, we observe that the alpha parameter selected generally decreases or remains the same as the percentage of missing data increases. This suggests that as the percentage of missing data increases, the time series part of the objective function should be weighted less heavily in the imputation because there is less time series information available for each observation in the dataset.

Number of Observations Per Patient In Figure 3, we present the MAE imputation accuracy results from the experiments in which we vary the number of observations per patient. We present the exact values and standard errors in this plot in the Appendix in Table 11. Across all of the experiments, we observe that as the time horizon increases, the performance of `med.knn` generally improves. This is expected, because as the time horizon increases, we include more observations per patient in the dataset, so there is more time series information that can be leveraged during the imputation process.

Similarly, the imputation accuracy of the `moving.avg` method generally improves as the time horizon increases. One notable exception is in the FHS dataset, the MAE of the `moving.avg` method increases as the time horizon increases from 10 to 20 years, while the MAE of `med.knn` remains relatively constant. From this, we can deduce that past observations of patients in the FHS dataset from 10 to 20 years prior have little predictive power for the other imputed values, which causes simple time series methods such as `moving.avg` to perform worse with more data. In contrast, the `med.knn` method has an exponential halflife parameter that we can tune so that observations from 10+ years ago are weighted less heavily in the imputation, so the performance remains about the same with the additional data.

One surprising trend that we observe in these graphs is the performance of `amelia`, which is another imputation method that takes into account time series information. On the DFCI dataset, as the time horizon increases, the imputation error increases. In addition, on the FHS dataset, as time horizon increases, the imputation error remains about the same. Only in the PPMI dataset does the performance of `amelia` noticeably improve as the time horizon increases.

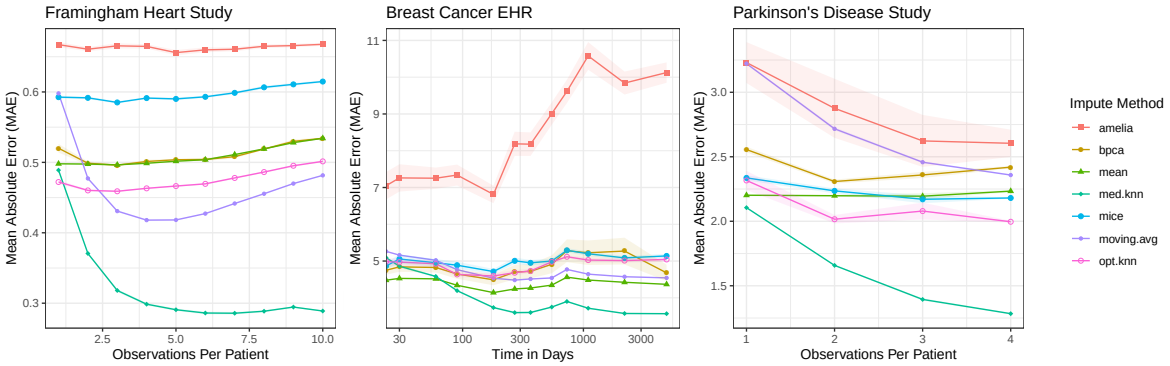

Fig. 3: Imputation errors for each method using the MAE metric on the FHS, DFCI, and PPMI datasets, varying the time horizon which determines the number of observations per patient. The missing data mechanism is fixed to MCAR, and the total percentage of missing data is fixed to 50%.

In Figure 4, we present the RMSE imputation accuracy results for the Observations Per Patient experiments. The results are similar to the MAE imputation accuracy results, and `med.knn` produces the imputation with the lowest RMSE across all experiments. One characteristic of the RMSE results is that they are much noisier, and in particular on the DFCI dataset the RMSE values do not decrease monotonically in a smooth fashion. Since the RMSE metric is more sensitive to outliers than the MAE metric, this suggests that there may be some outliers in the DFCI data which are added into the dataset at different time horizons.

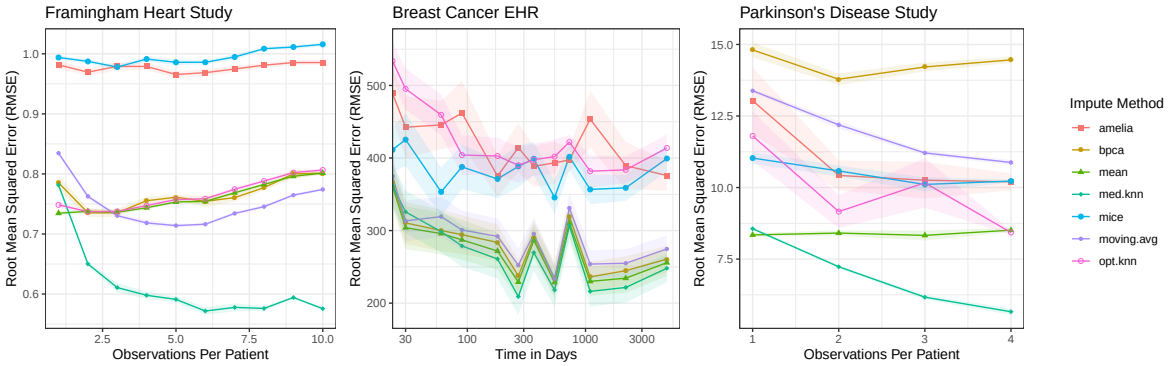

Fig. 4: Imputation errors for each method using the RMSE metric on the FHS, DFCI, and PPMI datasets, varying the time horizon which determines the number of observations per patient. The missing data mechanism is fixed to MCAR, and the total percentage of missing data is fixed to 50%.

In addition to evaluating the imputation accuracy of `med.knn` on datasets with varying numbers of observations per patient, we can also evaluate the imputation accuracy on subsets of patients within the DFCI dataset which have varying numbers of observations. In Figure 5, we present the imputation errors for `med.knn` on the DFCI dataset with 30% MCAR missing data, for subgroups of patients which have $1, 2, \ldots, 12$ observations per patient in the dataset. Overall, the MAE for the entire dataset is 3.331. For patients with one visit, and therefore one observation in the dataset, the average MAE is almost 3.5. In contrast, for patients with 10 or more visits, the average MAE is below 2.5. This suggests that in datasets with heterogeneous numbers of observations per patient, the `med.knn` imputation may be most accurate for the patients with the most observations in the dataset.

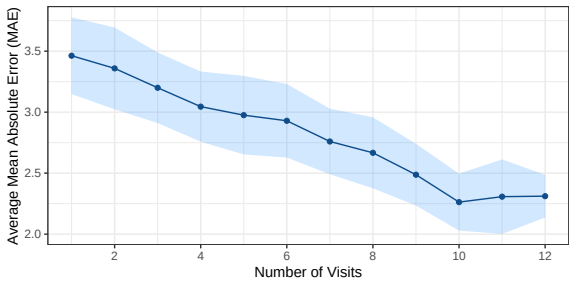

Fig. 5: Imputation errors for `med.knn` on the DFCI dataset with 30% MCAR missing data for subgroups of patients which have varying numbers of visits in the dataset.

Overall, from the Observations Per Patient experiments, we can conclude that `med.knn` method performs best with the additional time series information. As the time horizon increases, the imputation accuracy of `med.knn` generally improves or remains the same, while in a few cases the other time series methods `moving.avg` and `amelia` perform significantly worse with additional time series data. In addition, the imputation accuracy of the methods which do not take into account time series information (`bpca`, `mean`, `mice`, `opt.knn`) remains relatively constant as the time horizon varies. Furthermore, within a dataset that has heterogeneous numbers of observations per patient, such as EHR datasets, we may expect `med.knn` to most accurately impute values for the patients with the most observations in the dataset.

In the Appendix, we present the MedImpute hyperparameters which were selected in Observations Per Patient experiments for the FHS dataset. First, in Table 17, we show the median halflife parameters that were selected for each covariate for each experiment. For OPP ≤ 2, the selection of the halflife parameter does not impact the imputation, so the halflife parameter is set to 1 for each covariate. For OPP ≥ 3, the halflife parameters remain relatively constant for each covariate as the observations per patient varies. In Table 18, we show the median alpha parameters that were selected for each covariate. When the OPP $= 1$, there is no time series information in the dataset, so the alpha parameter is set to 0 for each covariate. For

OPP ≥ 2, the alpha parameters selected remain relatively constant for each covariate, with a few gradual trends for some of the covariates. For some covariates such as Age, Body Mass Index, and Systolic Blood Pressure, the selected alpha parameter gradually increases as OPP increases, and for other covariates such as Blood Glucose and High-Density Lipoproteins, the selected alpha parameter gradually decreases as OPP increases. This suggests that the addition of more time series data may change the `med.knn` imputation of each covariate differently.

Mechanism of Missing Data In Figure 6, we present the MAE imputation accuracy results from the experiments in which we vary the mechanism of missing data. We present the exact values and standard errors in this plot in the Appendix in Table 13. Across all of these experiments, we observe that `med.knn` has the best average MAE values by a significant margin.

In general, the imputation accuracy of all of the imputation methods increases or remains the same as the proportion of MNAR data increases. Two exceptions are the `moving.avg` method on the FHS dataset and the `amelia` method on the DFCI experiments, which both improve in performance at first as a small proportion of MNAR data is added. One possible explanation for this is that the MNAR data acts as a regularizer which helps these methods avoids overfitting to the dataset. However, in most cases the imputation error increases or remains constant as the percentage of MNAR data increases.

In the FHS MNAR experiments, the performance of all of the methods remains relatively constant, however the imputation error of `moving.avg` improves at $\gamma = 0.1$. Because `moving.avg` is the second-best performing method in these experiments, this means that the edge of the `med.knn` method slightly decreases in these experiments. In the PPMI MNAR experiments, the imputation error of all methods increases approximately linearly as the proportion of MNAR data increases. In the DFCI MNAR experiments, the imputation error for all methods except for `amelia` increases sharply at $\gamma = 0.1$, and then increases linearly afterwards as γ increases. As a result, for the experiments on the DFCI and PPMI datasets, the absolute improvement of `med.knn` over the comparator methods remains about the same as the proportion of MNAR data increases.

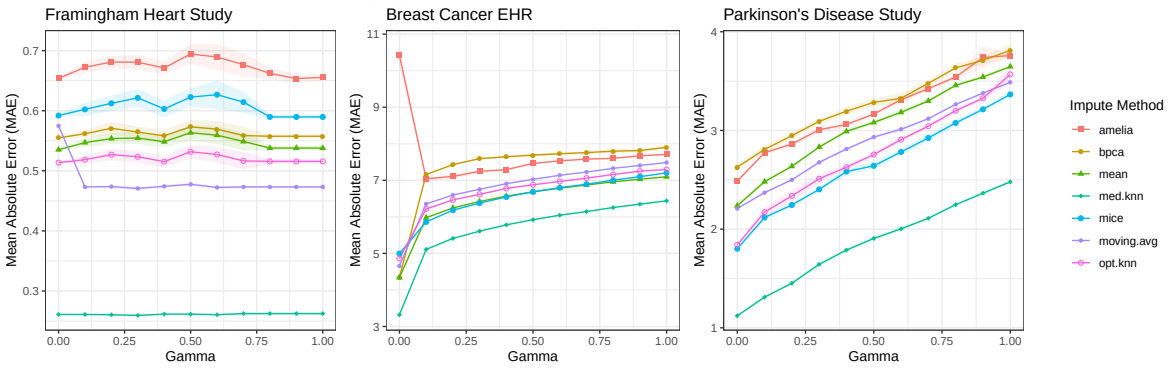

Fig. 6: Imputation errors for each method using the MAE metric on the FHS, DFCI, and PPMI datasets, varying the ratio of the missing data mechanism from $\gamma = 0$ (30% MCAR, 0% MNAR) to $\gamma = 1$ (0% MCAR, 30% MNAR). The total percentage of missing data is fixed to 30%.

In Figure 7, we present the RMSE imputation accuracy results for the missing data mechanism experiments. The results are largely consistent with the MAE imputation accuracy results. In particular, `med.knn` produces the imputation with the lowest RMSE by a significant margin across all experiments.

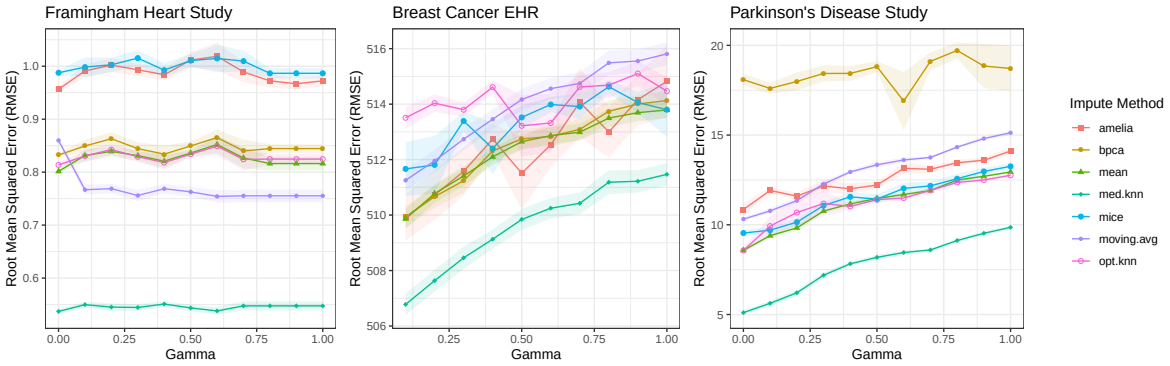

Fig. 7: Imputation errors for each method using the RMSE metric on the FHS, DFCI, and PPMI datasets, varying the ratio of the missing data mechanism from $\gamma = 0$ (30% MCAR, 0% MNAR) to $\gamma = 1$ (0% MCAR, 30% MNAR). The total percentage of missing data is fixed to 30%.

Overall, these experiments demonstrate that the `med.knn` method performs well relative to the other imputation methods even as the mechanism of missing data changes. In the MNAR experiments for the longitudinal datasets, FHS and PPMI, the relative imputation accuracy of the comparator methods remains approximately the same with the `med.knn` method performing best, with the exception of the `moving.avg` method which performs significantly worse. Thus, we can conclude that the `med.knn` method is well suited for imputing missing values according to the particular MNAR mechanism designed for longitudinal datasets which is described in Section 3.2.1. In the MNAR experiments for the EHR dataset DFCI, the relative imputation accuracy of the comparator methods remains approximately the same with the `med.knn` method performing best, with the exception of the `amelia` method which performs significantly better. Therefore, we can also conclude that the `med.knn` is suitable for imputing missing values according to the MNAR mechanism for EHR datasets which is described in Section 3.2.2.

In the Appendix, we present the MedImpute hyperparameters which were selected in Mechanism of Missing Data experiments for the FHS dataset. In Tables 19 and 20, we show the median halflife and alpha parameters that were selected for each covariate for each experiment, respectively. Across all of the experiments, we observe that the parameters selected during the validation procedure remain almost exactly constant. We conclude that varying the missing data mechanism for the FHS dataset according to the approach outlined in Section 3.2.1 has little impact on the `med.knn` imputation for this dataset.

3.5 Prediction Results

In this section, we provide the results from all experiments on the downstream prediction tasks. In particular, we present the downstream prediction results from the 1) Percentage of Missing Data, 2) Number of Observations Per Patient, and 3) Mechanism of Missing Data experiments. For the FHS and DFCI datasets, in which we train and evaluate classification models, we report the average out-of-sample AUC results. For the PPMI dataset, in which we train and evaluate regression models, we report the average out-of-sample MAE results.

Percentage of Missing Data In Figure 6, we present the performance on the downstream tasks from the experiments in which we vary the percentage of missing data. We present the exact values and standard errors in this plot in the Appendix in Table 10. Across all of the datasets, the `med.knn` method performs best, and the downstream performance of all methods generally declines as the missing level increases. In particular, the AUC values generally decrease for the classification tasks and the MAE values generally increase for the regression tasks as the percentage of missing data increases.

For the FHS dataset, while the downstream performance of all methods declines as the percentage of missing data increases, the downstream performance of `med.knn` declines least rapidly. In particular, with 20% missing data, the downstream AUC of `med.knn` is 0.897, compared to downstream AUC of 0.861 from the second-best method `bpca` and the baseline AUC of 0.901 with no additional missing data. With 50% missing data, the downstream AUC of `med.knn` is 0.864, compared to 0.826 for the second-best method `moving.avg`.

	χ^2 statistic (adjusted p-value)		
%	FHS	DFCI	PPMI
10	130 (<0.001***)	210 (<0.001***)	75 (<0.001***)
20	130 (<0.001***)	220 (<0.001***)	53 (<0.001***)
30	130 (<0.001***)	260 (<0.001***)	74 (<0.001***)
40	110 (<0.001***)	230 (<0.001***)	58 (<0.001***)
50	140 (<0.001***)	270 (<0.001***)	71 (<0.001***)

Table 3: The Friedman Rank test results for the downstream predictive tasks varying the percentage of missing data from 10-50% MCAR. The table shows the value of Friedman's Chi-squared statistic and p-value for the hypothesis test comparing `med.knn` against the benchmark methods for each experiment. The p-values are adjusted for multiple comparisons.

Similarly, for the DFCI dataset, the `med.knn` method performs best across all levels of missing data, and the downstream AUC values generally decrease as the missing level increases. The only exception is for the `amelia` method, where we do not observe a smooth trend because this method does not converge in some cases. In addition, the relative improvement of `med.knn` compared to the other imputation methods is lower for this dataset. At 50% missing data, the downstream AUC of `med.knn` is 0.889, compared to 0.884 for the second-best method `bpca` and the baseline AUC of 0.92 with no additional missing data.

Lastly, in the PPMI dataset, we observe the same trends that the `med.knn` method performs best, and the performance of all methods declines as the missing level increases. In this case, the downstream MAE for each method increases as the percentage of missing data increases. Across all levels of missing data, `med.knn` achieves the lowest downstream MAE. At 50% missing data, the downstream MAE of `med.knn` is 1.917, compared to 2.092 for the second-best method `opt.knn` and the baseline MAE of 1.170 with no additional missing data.

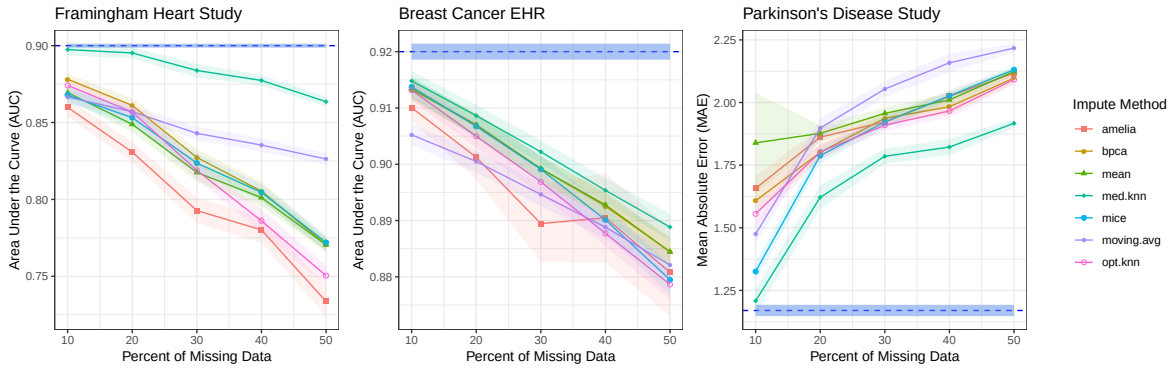

Fig. 8: Downstream accuracy results for each method on the FHS, DFCI, and PPMI datasets, varying the percentage of missing data from 10% to 50% according to the MCAR mechanism. On each plot, we overlay the downstream accuracy of a baseline model trained with no additional missing data as a dotted blue line (shaded with standard error bars).

In Table 3, we present the results from the Friedman Rank tests for each of the downstream predictive tasks varying the percentage of missing data. Similar to Friedman Rank tests for the imputation tasks, each test is significant with a p-value less than 0.001. These results demonstrate that the `med.knn` method is consistently ranked higher than the others for each of the downstream predictive tasks.

In Table 4, we present the results from the pairwise t-tests for each of the experiments. In this statistical test, we evaluate the differences in downstream predictive performance between `med.knn` and each of the comparison methods. We consider the differences in downstream AUC for the classification tasks, and we consider the differences in downstream MAE for the regression tasks. In most of the experiments, we observe that the differences in downstream AUC/MAE are statistically significant with p-values less than 0.001. These results demonstrate that the relative improvement in imputation accuracy for the `med.knn` method carries over to a relative improvement in performance on the downstream predictive tasks with different levels of MCAR data. Between the two classification tasks, we observe that the `med.knn` gives

FHS: Predicting 10-year Risk of Stroke

	Δ AUC (adjusted p-value)					
Missing %	`mice`	`moving.avg`	`amelia`	`bpca`	`mean`	`opt.knn`
10	0.0296 (<0.001***)	0.0309 (<0.001***)	0.0378 (<0.001***)	0.0193 (<0.001***)	0.0280 (<0.001***)	0.0233 (<0.001***)
20	0.0421 (<0.001***)	0.0382 (<0.001***)	0.0645 (<0.001***)	0.0341 (<0.001***)	0.0464 (<0.001***)	0.0384 (<0.001***)
30	0.0602 (<0.001***)	0.0408 (<0.001***)	0.0908 (<0.001***)	0.0566 (<0.001***)	0.0663 (<0.001***)	0.0649 (<0.001***)
40	0.0728 (<0.001***)	0.0420 (<0.001***)	0.0997 (<0.001***)	0.0720 (<0.001***)	0.0762 (<0.001***)	0.0913 (<0.001***)
50	0.0915 (<0.001***)	0.0373 (<0.001***)	0.1266 (<0.001***)	0.0931 (<0.001***)	0.0931 (<0.001***)	0.1132 (<0.001***)

DFCI: Predicting 60-day Risk of Mortality

	Δ AUC (adjusted p-value)					
Missing %	`mice`	`amelia`	`moving.avg`	`bpca`	`mean`	`opt.knn`
10	0.0010 (0.234)	0.0050 (<0.001***)	0.0196 (<0.001***)	0.0015 (0.261)	0.0013 (0.407)	0.0016 (<0.001***)
20	0.0019 (0.004**)	0.0060 (0.092)	0.0181 (<0.001***)	0.0016 (0.318)	0.0018 (0.234)	0.0037 (<0.001***)
30	0.0031 (0.003**)	0.0114 (0.052)	0.0176 (<0.001***)	0.0030 (0.037*)	0.0030 (0.037*)	0.0053 (<0.001***)
40	0.0056 (<0.001***)	0.0046 (0.075)	0.0169 (<0.001***)	0.0033 (0.032*)	0.0030 (0.044*)	0.0081 (<0.001***)
50	0.0094 (<0.001***)	0.0077 (0.065)	0.0167 (<0.001***)	0.0044 (0.003**)	0.0044 (0.003**)	0.0102 (<0.001***)

PPMI: Predicting the MoCA score

	Δ MAE (adjusted p-value)					
Missing %	`mice`	`amelia`	`moving.avg`	`bpca`	`mean`	`opt.knn`
10	-0.117 (0.027*)	-0.435 (<0.001***)	-0.288 (<0.001***)	-0.399 (<0.001***)	-0.631 (0.027*)	-0.347 (<0.001***)
20	-0.167 (0.004**)	-0.249 (0.002**)	-0.329 (<0.001***)	-0.180 (<0.001***)	-0.255 (<0.001***)	-0.181 (0.004**)
30	-0.137 (<0.001***)	-0.167 (<0.001***)	-0.296 (<0.001***)	-0.152 (<0.001***)	-0.171 (<0.001***)	-0.124 (<0.001***)
40	-0.204 (<0.001***)	-0.153 (<0.001***)	-0.362 (<0.001***)	-0.161 (<0.001***)	-0.188 (<0.001***)	-0.144 (0.002**)
50	-0.214 (<0.001***)	-0.207 (<0.001***)	-0.312 (<0.001***)	-0.181 (<0.001***)	-0.207 (<0.001***)	-0.175 (<0.001***)

Table 4: Pairwise t-tests between `med.knn` and benchmark methods for imputation tasks varying the percentage of missing data from 10-50% MCAR. The p-values are adjusted for multiple comparisons.

larger improvements in AUC on the FHS dataset than the DFCI dataset. In addition, we observe that as the percentage of missing data increases, the relative improvement of `med.knn` increases in general. These results are expected because as the percentage of missing data increases, the impact of the imputation method on the training data and the final prediction task increases as well. Since `med.knn` provides substantial improvements in imputation accuracy for all levels of missing data, having larger amounts of missing data generally leads to larger gains in downstream predictive accuracy. There are a few exceptions to this, for example `amelia`, `bpca`, `mean`, and `opt.knn` on the PPMI dataset, and `moving.avg` on the DFCI dataset. In these cases, the largest improvement for `med.knn` occurs at the 10% missing level. For these several examples, it follows that `med.knn` does a much better job at simulating the training dataset with 10% missing data, but the other methods begin to catch up as the percentage of missing data increases.

Number of Observations Per Patient In Figure 9, we present the performance on the downstream tasks from the experiments in which we vary the time horizon which determines the number of observations per patient. We present the exact values and standard errors in this plot in the Appendix in Table 12. Across all of the experiments, we observe that the downstream performance of `med.knn` tends to improve as the time horizon increases, so that the dataset includes more observations per patient. However, for each dataset, after a certain point there are diminishing returns, so that adding more observations per patient to the dataset does not improve the performance on the downstream task.

For the FHS dataset, in which the task is to predict 10-year risk of stroke, the downstream AUC of `med.knn` plateau starts to plateau at a time horizon of 6 years. For the DFCI dataset, in which the task is to predict 60-day risk of mortality, the downstream AUC of `med.knn` starts to plateau around 3 years. Similarly, for the PPMI dataset, in which the task is to predict the next year MoCA score, the downstream MAE reaches a minimum value at 3 years.

In comparison to the other methods, we observe that `med.knn` tends to perform relatively better with more observations per patient in the dataset. This indicates that the `med.knn` method is able to leverage the additional time series information more efficiently than the other methods. The only exception to this is `amelia` on the DFCI dataset, which outperforms `med.knn` with time horizons of 3 and 5 years, respectively. However, we observe that the `amelia` method is more unstable, and `med.knn` outperforms this method for the longest time horizon of 10 years.

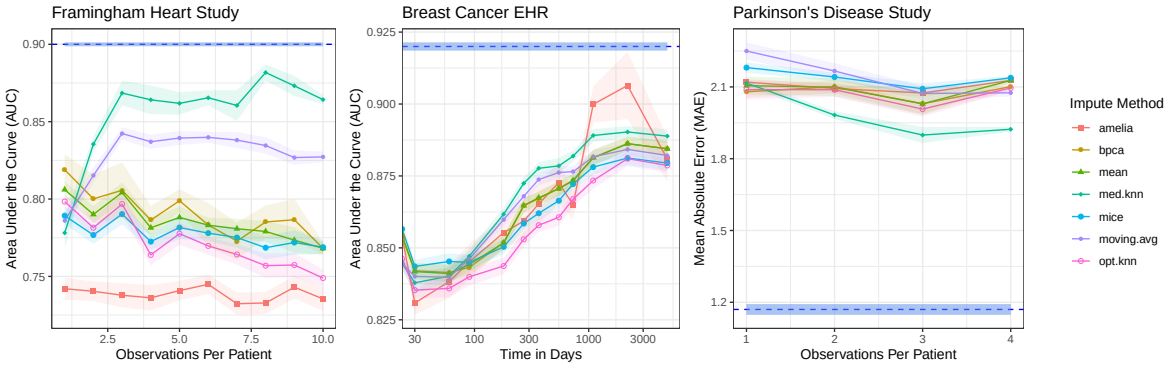

Fig. 9: Downstream accuracy results for each method on the FHS, DFCI, and PPMI datasets, varying the time horizon which determines the number of observations per patient. In these experiments, the missing data mechanism is fixed to MCAR, and the total percentage of missing data is fixed to 50%. On each plot, we overlay the downstream accuracy of a baseline model trained with no additional missing data as a dotted blue line (shaded with standard error bars).

Mechanism of Missing Data In Figure 10, we present the performance on the downstream tasks from the experiments in which we vary the mechanism of missing data. We present the exact values and standard errors in this plot in the Appendix in Table 14. In all of the experiments, we observe that the `med.knn` acheives the best downstream accuracy, typically by a substantial margin.

In the FHS dataset, the average AUC for `med.knn` remains around 0.89 and above across all proportions of MNAR data, while the second-best performing method `moving.avg` has an average AUC below 0.87. In the PPMI dataset, the downstream MAE values for all of the methods increases approximately linearly as the ratio of MNAR data increases. As a result, the relative improvement of `med.knn` on downstream tasks remains large for all of the MNAR experiments on longitudinal datasets.

On the other hand, the relative improvement of `med.knn` on downstream tasks is more varied for the MNAR experiments on EHR data. In the DFCI dataset, the downstream AUC values for each of the methods increases significantly when gamma = 0.1, and then decreases gradually as gamma increases further. These results are somewhat counterintuitive because the imputation errors for most of these methods increase significantly at gamma = 0.1, and then increase gradually afterwards. One possible explanation is that the DFCI dataset has some outlier values that tend to be missing under the MNAR mechanism for electronic health record data (described in Section 3.2.2), which typically skew the downstream prediction results. At the peak when gamma = 0.1, the relative improvement of `med.knn` is very small, with a downstream AUC of 0.916 compared to the next best method `mice` which has a downstream AUC of 0.915. At the extreme when gamma = 1, the downstream AUC of `med.knn` is 0.912 compared to 0.904 for the next best methods (`mice` and `bpca`).

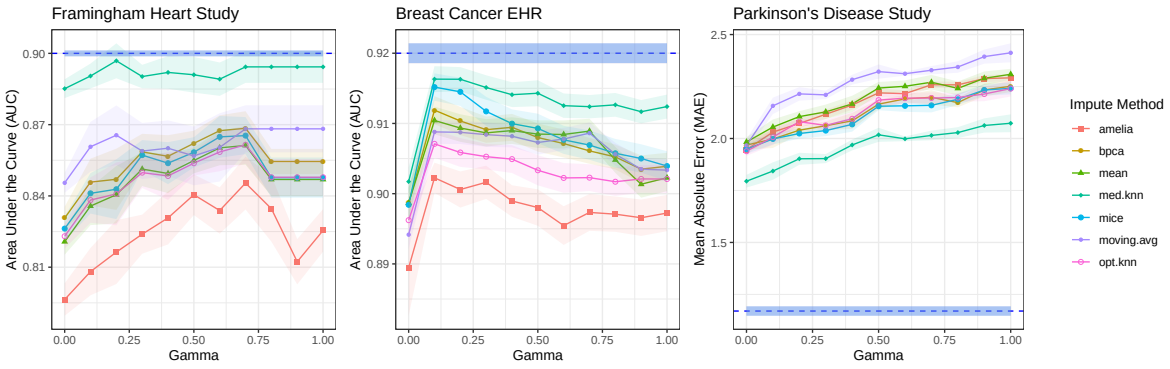

Fig. 10: Downstream accuracy results for each method on the FHS, DFCI, and PPMI datasets, varying the ratio of the missing data mechanism from $\gamma = 0$ (30% MCAR, 0% MNAR) to $\gamma = 1$ (0% MCAR, 30% MNAR). On each plot, we overlay the downstream accuracy of a baseline model trained with no additional missing data as a dotted blue line (shaded with standard error bars).

3.6 Discussion of the Computational Experiments on Real-World Clinical Datasets

In this section, we discuss the major takeaways from the computational experiments on real-world clinical datasets. For each dataset, we consider downstream models to predict patient outcomes that are clinically relevant, in order to simulate the performance of `med.knn` in practical applications. For the FHS and PPMI datasets, which are longitudinal studies, the clinical outcomes of interest are 10-year risk of stroke and next year MoCA score, which can be predicted using the most recent observation for each patient. For the DFCI dataset, which is an EHR dataset, the clinical outcome of interest is 60-day risk of mortality for late-stage cancer patients, which requires us to train models using all of the observations from each patient (using the latest observation for each patient would bias the results). As a result, the evaluation of the downstream models is different between the datasets. Furthermore, we conduct non-identical experiments on each dataset due to inherent dissimilarities in the time series structure.

Due to the significant differences between each dataset, we can draw separate conclusions from each one as a separate case study. The FHS dataset is a long term longitudinal study with many patients, few covariates, and a downstream classification task. In contrast, the PPMI dataset is a shorter longitudinal study with fewer patients, more covariates, and a downstream regression task. Finally, the DFCI dataset is an EHR dataset with irregularly recorded observations, the most patients, the most covariates, and a downstream classification task. The results from the computational experiments demonstrate that `med.knn` performs well across this range of diverse case studies. In particular, we show that this method performs well on datasets with: 1) large or small numbers of patients, 2) large or small numbers of covariates, and 3) regularly or irregularly recorded observations. Moreover, the application of `med.knn` for imputation led to improved downstream predictive performance on two binary classification tasks and one regression task.

Prior to training the downstream models, we do not perform any further preprocessing on the imputed data, so we preserve the correlation structure of the original dataset. As a result, since these are real-world datasets, there may be unexpected correlations between the predictors which impact the accuracy of the downstream models. One could apply PCA or another dimensionality-reduction method to transform the feature space prior to training downstream models on the imputed datasets. However, this analysis is outside of the scope of this set of computational experiments.

In the Percentage of Missing Data experiments, we observe that increased imputation accuracy does not always translate into increased downstream model accuracy. For example, on the DFCI dataset, `bpca` performs poorly on the imputation task (see Figure 1), but is one of the top-performing methods on the downstream predictive task (see Figure 8). This is possible because in the downstream predictive task, some features are more significant than others, so having a large imputation error on the insignificant features may only result in a small decline in downstream model accuracy. However, we also observed that in all datasets, `med.knn` consistently performed best on both the imputation and downstream tasks, by a significant margin in most cases. These results suggest that for all three of the real-world datasets considered here, `med.knn` leads to improvements in imputation accuracy on the clinically significant covariates in each downstream model.

In the OPP experiments, the major trend that we observe is that the `med.knn` method performs significantly better with more time series data. For example, in the FHS dataset, the imputation accuracy and downstream performance of `med.knn` improves dramatically as OPP increases from one to four. This makes sense because as we include more observations per patient in the dataset, there is more relevant information available to impute the missing covariates for each patient. We expect that this explains why the relative improvement of `med.knn` is less significant on the DFCI dataset for several of the experiments. In this dataset, over half of the patients have a single observation, so there is limited time series available to fill in the missing values for these patients. In contrast, in the FHS dataset, every patient has 10 observations in the full dataset, so there is more data available to aid the imputation.

In the MNAR experiments, we demonstrate that `med.knn` works under missing data mechanisms that are frequently encountered in practice. Longitudinal studies often contain systematic missing information on some clinical examinations based on decisions made by the designers of the study. For example, the Framingham Heart Study dataset has expanded over time as clinicians have incorporated more and more variables that are suspected to be correlated with heart disease (Mahmood et al. 2014). However, since some of these variables were not recorded initially, they are systematically missing from this dataset. In EHR datasets, clinical covariates recorded for each visit typically vary based the health condition of the patient. Patients at higher risk are likely to undergo more detailed medical examinations, resulting in fewer missing values. Through the MNAR experiments for each case study, we show that `med.knn` is an effective method for imputing missing values under these specific mechanisms of missing data for longitudinal studies and EHR datasets.

4 Scaling Experiments on Simulated Clinical Datasets

In this section, we present scaling experiments on simulated clinical datasets. In Section 4.1, we describe the data generation process which allows us to construct simulated longitudinal clinical datasets with $10,000$'s of observations and 100's of features. In Section 4.2, we describe the experimental setup of the scaling experiments, which considers two variations of the `med.knn` method. In Section 4.3, we report the results of the scaling experiments, including the imputation accuracy and timing results.

4.1 Simulated Data: Synthea

We create synthetic EHR to test the performance of the algorithm in higher instances of both the number of observations and the number of features using the Synthea synthetic patient population simulator. It constitutes an open-source, synthetic patient generator that aims to model the medical history of patients using specific demographic information (Walonoski et al. 2018). Patient records are generated using simulation processes that follow disease progression patterns published in the medical literature. For each synthetic patient, Synthea data contains a complete medical history, including medications, allergies, medical encounters, and social determinants of health. We pre-processed the records combining them into a single dataset that contains a summary of all the information available at each visit.

Since we leverage this data source for experiments testing the scalability of the algorithm, we do not limit the amount of observations to a specific number. Each patient in the data is associated on average with 20 distinct visits (observations). We aggregate the EHR into 344 distinct features. Each experiment randomly samples a subset of these features to compare the computational time needed by the algorithm. The covariates that comprise the data include demographic characteristics, diagnosis and procedure codes, medical prescriptions, and lab test results. We do not include any downstream prediction task.

4.2 Experimental Setup for the Scaling Experiments

In this section, we go over the experimental setup for the scaling experiments. We use synthetically generated data for EHR varying both the number of observations n and the number of features p. Our goal is to evaluate the scaling performance and accuracy of the algorithm comparing the two proposed methods for tuning the hyperparameters α_d and h_d.

One of the most well-established approach for hyperparameter tuning in machine learning is K-fold cross-validation (Kohavi et al. 1995). In the time series setting, Bergmeir et al. (2018) showed that this technique is applicable for time series models, in particular for the case of autoregression models. However, due to the large number of combinations of different values for α_d and h_d, in the case of `med.knn`, the computation time for the K-fold cross-validation scales at an quadratic rate as the number of covariates increases. For this reason, we propose a custom tuning procedure to select the hyperparameters. We conduct a series of experiments comparing the following hyperparameter selection processes:

1. **Grid Search**: This approach uses the well-established 10-fold cross-validation process to determine the hyperparameters h_d and α_d for every variable. Prior to solving the algorithm, 10% of the values of each feature are artificially removed. A set of values is defined and all their combinations are evaluated for each feature individually when solving the reduced version of the dataset. The grid for α_d was set to $[0.0, 0.1, \ldots, 1.0]$ and for h_d to $[90, 180, 365, 1000]$.
2. **Custom Tuning**: The custom tuning procedure proposed in Section 2.4. This is a heuristic method to decompose the problem into multiple parts, first learning h_d for each covariate, and then learning α_d for each covariate. This approach does not involve cross-validation and allows for parallel computations as the problem is fully decoupled.

For each experiment, we evaluate the imputation accuracy of each approach using the MAE and RMSE metrics, as defined in Equations 30 and 31. In addition, we also compare their scaling performance by measuring the average time needed for completion. In these experiments, we did not consider the prediction task as in Section 3. Here, we limit the types of experiments only to Percentage of Missing Data following the experimental set up of Section 3.3.

We vary the number of features between $[50, 100, 200, 300]$ and the number of observations between $[1000, 12500, 25000, 50000, 75000]$. These bounds were chosen as they represent the most common spectra of problem sizes that we encounter in healthcare applications. We repeat all experiments for five random seeds and average the results.

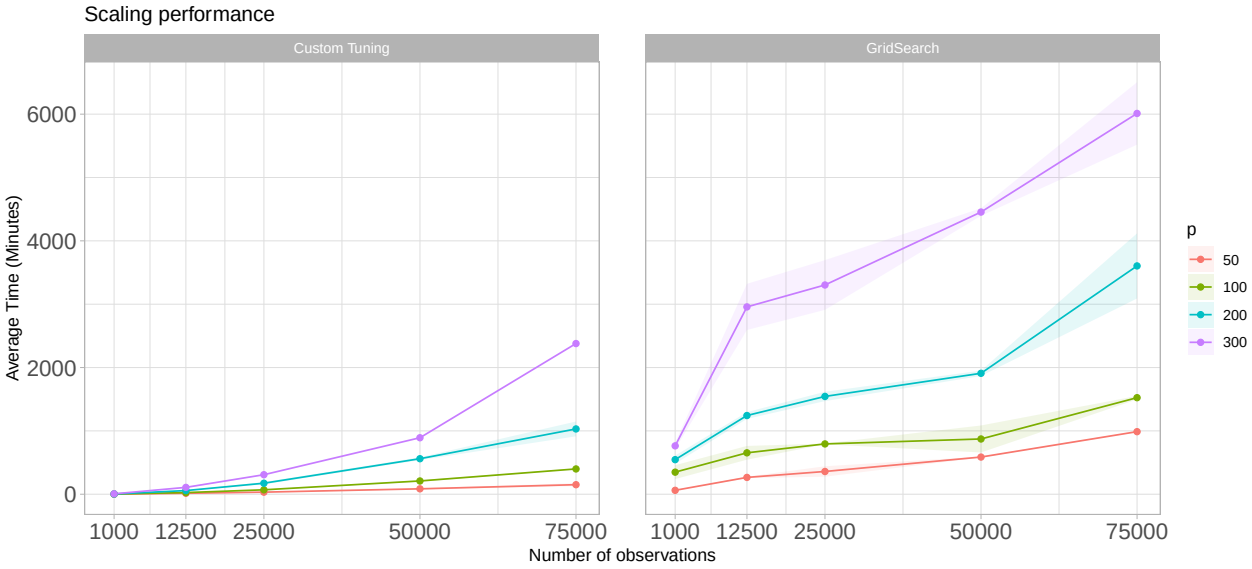

Fig. 11: Average time for MedImpute methods to complete imputation tasks on the Synthea dataset using different procedures for hyperparameter tuning, with varying numbers of observations n and features p in the dataset.

4.3 Results of the Scaling Experiments

In this section, we present the results from the scaling experiments. In Figure 11, we demonstrate the timing results. While both the methods scale to the largest problem size with $n = 75000$ observations and $p = 300$ features, the Custom Tuning procedure is -60.42% faster than Grid Search; the traditional cross-validation procedure. Across all experiments, Custom Tuning is on average -87.05% faster than Grid Search. We notice that for the lower problem sizes, the Custom Tuning approach leads almost instantaneous algorithm completion while Gridsearch requires up to 12 hours to solve.

Figure 12 presents the results referring to imputation accuracy. The two procedures lead to minimal differences in imputation performance. Across all experiments, the Custom Tuning procedure is slightly more accurate than the GridSearch procedure, with an average improvement of -4.36% in MAE. The gap between the two processes is larger when $n \in [25000, 50000]$ leading to an average reduction of -8.81% of the imputation error. We also note that only when $n = 1000$, GridSearch as the MAE is increased on average by 2.82% by the new method. In all other combinations, Custom Tuning leads to more accurate results with the maximum improvement reaching a reduction of 10.48% ($n = 50000$, $p = 100$). Detailed results for the RMSE metric are provided in Figure 14 at the Appendix.

4.4 Discussion of the Scaling Experiments on Simulated Clinical Datasets

The results from the scaling experiments demonstrate that the custom tuning procedure for the MedImpute hyperparameters α_d and h_d is highly effective and efficient. In particular, the proposed method significantly reduces the computational time required, while also giving a slight improvement in imputation accuracy as well compared to traditional cross-validation. Using the methodology, we are able to scale the algorithm to higher problem instances without sacrificing its imputation performance.

An analysis of the runtime complexity of the two hyperparameter selection methods provides further insights into these results. The key bottleneck of the `med.knn` algorithm is computing the K-NN assignment on $\mathbf{X}$ to update $\mathbf{Z}$ in each coordinate descent step, which requires $\mathcal{O}(n \log n)$ operations. The Grid Search procedure requires $\mathcal{O}(p^2)$ iterations to identify the best values for α_d and h_d, so the complete runtime for this method is $\mathcal{O}(np^2 \log n)$. On the other hand, the Custom tuning procedure only requires $\mathcal{O}(p)$ iterations because each hyperparameter for each covariate can be computed independently of the remaining covariates. As a result, this method scales in a linear fashion with respect to the number of covariates, and the full runtime is $\mathcal{O}(np \log n)$.

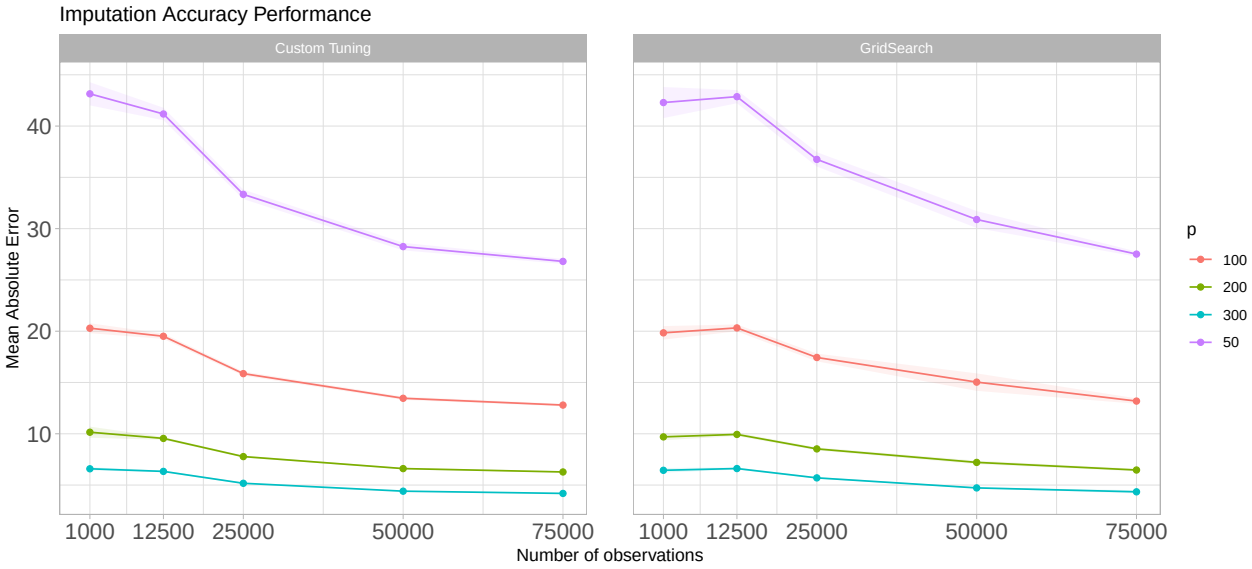

Fig. 12: Average MAE imputation errors for MedImpute methods on the Synthea dataset using different procedures for hyperparameter tuning, with varying numbers of observations n and features p in the dataset.

Despite these theoretical asymptotic runtime guarantees, we recognize that the `med.knn` method with the Custom Tuning procedure for hyperparameter tuning still takes up to 16 hours in datasets with $n \sim 50{,}000$ observations. However, given that the imputation task usually takes place once in the pre-processing part of the data analysis, we believe that the time cost is not significantly high. Moreover, the Custom tuning process allows for decoupling the problem in smaller instances. Thus, the application of parallel computing techniques can further improve the scaling performance of the algorithm.

5 Discussion

MedImpute is an extension of the OptImpute framework introduced by Bertsimas et al. (2018b). MedImpute uses the same optimization approach to solving the missing data problem. However, the optimization formulation is significantly different and more general than the OptImpute formulations in order to incorporate additional time series information present in cross-sectional data. The new formulation provides a structured way of accounting for observations from the same entity and re-weighting the objective function to incorporate time series information. As a result, the resulting imputation algorithm `med.knn` from the MedImpute framework outperforms `opt.knn` from the OptImpute framework and other benchmark imputation methods on real-world clinical datasets with patients observed over time.

In the MedImpute formulation, two new parameters are introduced, α_d, h_d, that are specific to each covariate d. The proposed Custom Tuning procedure allows for learning the values of these parameters more efficiently compared to a traditional Grid Search approach. In addition, these parameters are interpretable in a clinical context, yielding insights regarding the significance of time in their determination. For example, in the FHS dataset, we learn different values of α_d for chronic disease indicators such as Type 2 Diabetes Mellitus (T2DM) and lab values such as Systolic Blood Pressure (SBP). It is likely that an individual diagnosed with T2DM will continue to have this diagnosis regardless of the other covariates (American 2010), so MedImpute finds α_{d} relatively close to 1 for this feature. On the other hand, the lab measurement of SBP may vary significantly during a single day (Millar-Craig et al. 1978), so previous observations of this covariate from the same individual provide relatively less information. For this feature, MedImpute finds α_d closer to 0 so that the K-nearest neighbors are weighted more heavily in the imputation. In addition, we learn h_d to determine the relative weights that we give to observations of feature d from the same individual based on time elapsed. MedImpute selects higher values of h_d for features that change slowly over time such as the Body Mass Index and lower values for features that change rapidly over time such as SBP.

Beyond the healthcare setting, cross-sectional datasets are also quite common in other areas such as finance and economics. Our algorithm can be generalized and applied to any data where there is a time series component and multiple observations are tied to the same entity. The entity may represent a patient,

as we portray in this work, or something else that is observed over time such as a financial organization, region, or country. Therefore, the MedImpute imputation framework and the associated `med.knn` algorithm may be applied to impute missing values in other domains as well.

6 Conclusions

In this paper, we propose the optimization framework MedImpute that addresses the missing data problem for multivariate data in time series encountered in medical applications. We introduce a new imputation algorithm `med.knn` that yields high quality solutions using optimization techniques combined with fast first-order methods. Through computational experiments on three real-world clinical datasets, including two longitudinal studies and one EHR dataset, we show that `med.knn` offers statistically significant gains in imputation quality over state-of-the-art imputation methods, which leads to improved out-of-sample performance on downstream tasks. Through scaling experiments on a synthetic EHR dataset, we demonstrate that `med.knn` can be applied to complete datasets with 10,000's of observations and 100's of features. As a flexible, accurate, and intuitive approach, MedImpute has the potential to become an indispensable tool for applications with longitudinal missing data. Promising areas for future work include: (1) applications of this method to longitudinal datasets that are not related to healthcare, (2) additional experiments to assess the performance on downstream predictive tasks with transformed feature spaces, (3) extensions of the optimization framework to incorporate more specialized structure that is present in longitudinal healthcare datasets.

Acknowledgements We would like to thank the reviewers for their detailed feedback and suggestions which have improved the quality of this manuscript.

References

American DA (2010) Standards of medical care in diabetes—2010. Diabetes Care 33(Supplement 1):S11–S61, DOI 10.2337/dc10-S011, URL `http://care.diabetesjournals.org/content/33/Supplement_1/S11`, `http://care.diabetesjournals.org/content/33/Supplement_1/S11.full.pdf`

Bergmeir C, Hyndman RJ, Koo B (2018) A note on the validity of cross-validation for evaluating autoregressive time series prediction. Computational Statistics and Data Analysis 120:70 – 83, DOI https://doi.org/10.1016/j.csda.2017.11.003, URL `http://www.sciencedirect.com/science/article/pii/S0167947317302384`

Bertsekas D (1999) Nonlinear Programming. Athena Scientific

Bertsimas D, Dunn J, Pawlowski C, Silberholz J, Weinstein A, Zhuo YD, Chen E, Elfiky AA (2018a) Applied informatics decision support tool for mortality predictions in patients with cancer. JCO Clinical Cancer Informatics (2):1–11, DOI 10.1200/CCI.18.00003, URL `https://doi.org/10.1200/CCI.18.00003`, `https://doi.org/10.1200/CCI.18.00003`

Bertsimas D, Pawlowski C, Zhuo YD (2018b) From predictive methods to missing data imputation: An optimization approach. Journal of Machine Learning Research 18(196):1–39, URL `http://jmlr.org/papers/v18/17-073.html`

Birkhead GS, Klompas M, Shah NR (2015) Uses of electronic health records for public health surveillance to advance public health. Annual Review of Public Health 36(1):345–359, DOI 10.1146/annurev-publhealth-031914-122747, URL `https://doi.org/10.1146/annurev-publhealth-031914-122747`, pMID: 25581157, `https://doi.org/10.1146/annurev-publhealth-031914-122747`

van Buuren S, Groothuis-Oudshoorn K (2011) MICE: Multivariate imputation by chained equations in R. Journal of Statistical Software, Articles 45(3):1–67, DOI 10.18637/jss.v045.i03, URL `https://www.jstatsoft.org/v045/i03`

Callahan A, Shah NH (2017) Chapter 19: Machine learning in healthcare. In: Sheikh A, Cresswell KM, Wright A, Bates DW (eds) Key Advances in Clinical Informatics, Academic Press, pp 279 – 291, DOI https://doi.org/10.1016/B978-0-12-809523-2.00019-4, URL `http://www.sciencedirect.com/science/article/pii/B9780128095232000194`

Caruana EJ, Roman M, Hernández-Sánchez J, Solli P (2015) Longitudinal studies. Journal of thoracic disease 7(11):E537–40, DOI 10.3978/j.issn.2072-1439.2015.10.63, URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4669300/`

CD N, RJ L (2015) Missing data: How to best account for what is not known. JAMA 314(9):940–941, DOI 10.1001/jama.2015.10516, URL +http://dx.doi.org/10.1001/jama.2015.10516, /data/journals/jama/934374/jgm150007.pdf

Che Z, Purushotham S, Cho K, Sontag D, Liu Y (2018) Recurrent neural networks for multivariate time series with missing values. Scientific Reports 8(1):6085, DOI 10.1038/s41598-018-24271-9, URL https://doi.org/10.1038/s41598-018-24271-9

Crookston NL, Finley AO (2008) yaimpute: an r package for knn imputation. Journal of Statistical Software 23 (10) 16 p

D'Agostino RB, Pencina MJ, Massaro JM, Coady S (2013) Cardiovascular disease risk assessment: Insights from Framingham. Global Heart 8(1):11 – 23, DOI 10.1016/j.gheart.2013.01.001, URL http://www.sciencedirect.com/science/article/pii/S2211816013000057, framingham Legacy Issue.

Daniel Levy SB (2006) A change of heart: Unraveling the mysteries of cardiovascular disease. New York : Vintage

Faria JC, Demétrio CGB, Allaman IB (2018) bpca: Biplot of Multivariate Data Based on Principal Components Analysis. UESC and ESALQ, Ilheus, Bahia, Brasil and Piracicaba, Sao Paulo, Brasil

Flores A, Tito H, Silva C (2019) Local average of nearest neighbors: Univariate time series imputation. International Journal of Advanced Computer Science and Applications 10(8):45–50

Friedman J, Hastie T, Tibshirani R (2009) glmnet: Lasso and elastic-net regularized generalized linear models. R package version 1(4)

Hastie T, Tibshirani R, Friedman J (2009) The elements of statistical learning: data mining, inference, and prediction. Springer Science & Business Media

Honaker J, Joseph A, King G, Scheve K, Singh N (1999) Amelia: A program for missing data. Department of Government, Harvard University

Honaker J, King G, Blackwell M (2011) Amelia II: A program for missing data. Journal of Statistical Software, Articles 45(7):1–47, DOI 10.18637/jss.v045.i07, URL https://www.jstatsoft.org/v045/i07

Ismail Fawaz H, Forestier G, Weber J, Idoumghar L, Muller PA (2019) Deep learning for time series classification: a review. Data Mining and Knowledge Discovery 33(4):917–963

Janssen KJ, Donders ART, Harrell FE, Vergouwe Y, Chen Q, Grobbee DE, Moons KG (2010) Missing covariate data in medical research: To impute is better than to ignore. Journal of Clinical Epidemiology 63(7):721 – 727, DOI https://doi.org/10.1016/j.jclinepi.2009.12.008, URL http://www.sciencedirect.com/science/article/pii/S0895435610000193

King G, Honaker J, Joseph A, Scheve K (2001) Analyzing incomplete political science data: An alternative algorithm for multiple imputation. American political science review 95(1):49–69

Kohavi R, et al. (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Ijcai, Montreal, Canada, vol 14, pp 1137–1145

Landrum MB, Becker MP (2001) A multiple imputation strategy for incomplete longitudinal data. Statistics in Medicine 20(17-18):2741–2760, DOI 10.1002/sim.740, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.740, https://onlinelibrary.wiley.com/doi/pdf/10.1002/sim.740

Levey AS, Eckardt KU, Tsukamoto Y, Levin A, Coresh J, Rossert J, Zeeuw DD, Hostetter TH, Lameire N, Eknoyan G (2005) Definition and classification of Chronic Kidney Disease: A position statement from kidney disease: Improving global outcomes (kdigo). Kidney International 67(6):2089 – 2100, DOI 10.1111/j.1523-1755.2005.00365.x, URL http://www.sciencedirect.com/science/article/pii/S0085253815506984

Lipton ZC, Kale D, Wetzel R (2016) Directly modeling missing data in sequences with RNNs: Improved classification of clinical time series. In: Doshi-Velez F, Fackler J, Kale D, Wallace B, Wiens J (eds) Proceedings of the 1st Machine Learning for Healthcare Conference, PMLR, Children's Hospital LA, Los Angeles, CA, USA, Proceedings of Machine Learning Research, vol 56, pp 253–270, URL http://proceedings.mlr.press/v56/Lipton16.html

Little RJ, Rubin DB (2019) Statistical Analysis with Missing Data, vol 793. Wiley

Mahmood SS, Levy D, Vasan RS, Wang TJ (2014) The Framingham Heart Study and the epidemiology of cardiovascular disease: A historical perspective. The Lancet 383(9921):999 – 1008, DOI 10.1016/S0140-6736(13)61752-3, URL http://www.sciencedirect.com/science/article/pii/S0140673613617523

Marek K, Jennings D, Lasch S, Siderowf A, Tanner C, Simuni T, Coffey C, Kieburtz K, Flagg E, Chowdhury S, et al. (2011) The parkinson progression marker initiative (ppmi). Progress in neurobiology 95(4):629–635

Millar-Craig M, Bishop C, Raftery E (1978) Circadian variation of blood-pressure. The Lancet 311(8068):795 – 797, DOI 10.1016/S0140-6736(78)92998-7, URL http://www.sciencedirect.com/

science/article/pii/S0140673678929987, originally published as Volume 1, Issue 8068.

Nasreddine ZS, Phillips NA, Bédirian V, Charbonneau S, Whitehead V, Collin I, Cummings JL, Chertkow H (2005) The montreal cognitive assessment, moca: a brief screening tool for mild cognitive impairment. Journal of the American Geriatrics Society 53(4):695–699

National Heart, Lung, and Blood Institute, Boston University (2012) The Framingham Heart Study: 50 years of research success. Framingham, MA: Framingham Heart Study Https://biolincc.nhlbi.nih.gov/studies/fhs/?q=framingham

Oba S, Sato Ma, Takemasa I, Monden M, Matsubara Ki, Ishii S (2003) A Bayesian missing value estimation method for gene expression profile data. Bioinformatics 19(16):2088–2096, DOI 10.1093/bioinformatics/btg287, URL http://dx.doi.org/10.1093/bioinformatics/btg287, /oup/backfile/content_public/journal/bioinformatics/19/16/10.1093/bioinformatics/btg287/2/btg287.pdf

Obermeyer Z, Emanuel EJ (2016) Predicting the future: Big data, machine learning, and clinical medicine. New England Journal of Medicine 375(13):1216–1219, DOI 10.1056/NEJMp1606181, URL https://doi.org/10.1056/NEJMp1606181, pMID: 27682033, https://doi.org/10.1056/NEJMp1606181

P L, EA S, DB A (2015) Multiple imputation: A flexible tool for handling missing data. JAMA 314(18):1966–1967, DOI 10.1001/jama.2015.15281, URL +http://dx.doi.org/10.1001/jama.2015.15281, /data/journals/jama/934661/jgm150014.pdf

Pedersen AB, Mikkelsen EM, Cronin-Fenton D, Kristensen NR, Pham TM, Pedersen L, Petersen I (2017) Missing data and multiple imputation in clinical epidemiological research. Clinical Epidemiology 9, DOI 10.2147/CLEP.S129785, URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC5358992/

Rubin DB (1996) Multiple imputation after 18+ years. Journal of the American Statistical Association 91(434):473–489, DOI 10.1080/01621459.1996.10476908, URL https://amstat.tandfonline.com/doi/abs/10.1080/01621459.1996.10476908, https://amstat.tandfonline.com/doi/pdf/10.1080/01621459.1996.10476908

Schafer JL, Olsen MK (1998) Multiple imputation for multivariate missing-data problems: A data analyst's perspective. Multivariate Behavioral Research 33(4):545–571, DOI 10.1207/s15327906mbr3304_5, URL https://doi.org/10.1207/s15327906mbr3304_5, pMID: 26753828, https://doi.org/10.1207/s15327906mbr3304_5

Shickel B, Tighe PJ, Bihorac A, Rashidi P (2018) Deep EHR: A survey of recent advances in deep learning techniques for electronic health record (EHR) analysis. IEEE Journal of Biomedical and Health Informatics 22(5):1589–1604, DOI 10.1109/JBHI.2017.2767063

Shor B, Bafumi J, Keele L, Park D (2007) A bayesian multilevel modeling approach to time-series cross-sectional data. Political Analysis 15(2):165–181, DOI 10.1093/pan/mpm006

Shrive FM, Stuart H, Quan H, Ghali WA (2006) Dealing with missing data in a multi-question depression scale: A comparison of imputation methods. BMC Medical Research Methodology 6(1):57, DOI 10.1186/1471-2288-6-57, URL https://doi.org/10.1186/1471-2288-6-57

Siddiqui O, Ali MW (1998) A comparison of the random-effects pattern mixture model with last-observation-carried-forward (locf) analysis in longitudinal clinical trials with dropouts. Journal of biopharmaceutical statistics 8(4):545–563

Sterne JA, White IR, Carlin JB, Spratt M, Royston P, Kenward MG, Wood AM, Carpenter JR (2009) Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls. Bmj 338:b2393

TB M, AS D (2013) The inevitable application of big data to health care. JAMA 309(13):1351–1352, DOI 10.1001/jama.2013.393, URL +http://dx.doi.org/10.1001/jama.2013.393, /data/journals/jama/926712/jvp130007_1351_1352.pdf

Walonoski J, Kramer M, Nichols J, Quina A, Moesel C, Hall D, Duffett C, Dube K, Gallagher T, McLachlan S (2018) Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record. Journal of the American Medical Informatics Association 25(3):230–238

Ware JH, Harrington D, Hunter DJ, D'Agostino RB (2012) Missing data. New England Journal of Medicine 367(14):1353–1354, DOI 10.1056/NEJMsm1210043, URL https://doi.org/10.1056/NEJMsm1210043, https://doi.org/10.1056/NEJMsm1210043

Wood AM, White IR, Thompson SG (2004) Are missing outcome data adequately handled? A review of published randomized controlled trials in major medical journals. Clinical Trials 1(4):368–376, DOI 10.1191/1740774504cn032oa, URL https://doi.org/10.1191/1740774504cn032oa, pMID: 16279275, https://doi.org/10.1191/1740774504cn032oa

Wright SJ (2015) Coordinate descent algorithms. Mathematical Programming 151(1):3–34

Zhang Z (2016) Multiple imputation for time series data with Amelia package. Annals of Translational Medicine 4(3), URL http://atm.amegroups.com/article/view/8846

Zhao J, Henriksson A (2016) Learning temporal weights of clinical events using variable importance. BMC medical informatics and decision making 16(2):71

7 Appendix

7.1 Detailed Properties of Real-World Clinical Datasets

In this section, we provide additional details regarding the real-world clinical datasets which are used in the computational experiments. In Tables 5, 7, and 6, we show the percentage of missing data in each covariate for the FHS, PPMI, and the DFCI datasets, respectively. In Figure 13, we show a histogram of the number of observations per patient in the DFCI dataset. Table 8 shows summary statistics of inter-visit time intervals (in days) for the DFCI dataset.

Feature	**% Missing**
HDL Cholesterol Levels	70.54
Hematocrit Levels	30.53
History of Diabetes	21.14
Blood Glucose Levels	14.91
Smoking	10.5
BMI	7.18
History of Cardiovascular Disease (CVD)	3.88
Presence of Left Ventricular Hypertrophy (LVH)	2.15
Prescripion of Antihypertensive Medication (AHT)	1.93
Systolic Blood Pressure (SBP)	0.51
Age	0.07
Gender	0
History of Atrial Fibrilation (Afib)	0

Table 5: Percentage of missing data in each covariate in the original FHS dataset.

Feature	**% Missing**
History of Psoriatic arthritis	99.98
Creatinine Clearance Levels	99.96
AFP Mutation	99.91
CA.19.9 Mutation	98.74
ALT Mutation	96.92
AST Mutation	96.92
CA.125 Mutation	96.84
Direct Bilirubin Levels	77.73
CEA Mutation	69.32
CA.27.29 Mutation	62.91
Percentage Difference in Weight from Previous Measurement	50.88
Hispanic Race	29
Hematocrit Levels	25.15
Creatinine Levels	24.59
Total Bilirubin Levels	23.69
Albumin Levels	23.61
White Blood Cells Count	22.86
Systolic Blood Pressure Levels	5.56
Pulse	5.56
Weight	0.46
History of Myocardial Infarction	0.1
History of Congestive Heart Failure	0.1
History of Peripheral Artery Disease	0.1
History of Stroke	0.1
History of Dementia	0.1
History of Pulmonary Disease	0.1
History of Rheumatic Disease	0.1
History of Peptic Ulcer Disease	0.1
History of Mild Liver Disease	0.1

Table 6: Percentage of missing data in each covariate in the original DFCI dataset.

Feature	% Missing
History of Diabetes Mellitus	0.1
DMcx	0.1
History of Paralysis	0.1
History of Renal Failure	0.1
History of Severe Liver Disease	0.1
History of Metabolic Disease	0.1
History of HIV	0.1
Immunotherapy Prescription	0
Targetted Therapy Prescription	0
Number of Drugs Prescribed	0
Number of Blood Transfusions	0
Total Number of Inpatient Visits	0
Total Number of Outpatient Visits	0
Line of Therapy Prescribed	0
General Cancer Stage	0
Pathological Cancer Stage	0
Clinical Cancer Stage	0
Number of Diagnoses	0
Gender	0
White Race	0
Black Race	0
Age at Diagnosis	0
Age at Treatment	0
Divorced	0
Married	0
ACETAMINOPHEN	0
ALARIS	0
AMBULATORY	0
APREPITANT	0
ATROPINE	0
BEVACIZUMAB	0
BORTEZOMIB	0
CARBOPLATIN	0
CISPLATIN	0
CYCLOPHOSPHAMIDE	0
DARBEPOETIN	0
DEXAMETHASONE	0
DEXTROSE	0
DIPHENHYDRAMINE	0
DOCETAXEL	0
DOXORUBICIN	0
EPOETIN	0
ETOPOSIDE	0
EVACUATED	0
FAMOTIDINE	0
FILGRASTIM	0
FLUOROURACIL	0
FOSAPREPITANT	0
GEMCITABINE	0
IRINOTECAN	0
IV	0
LEUCOVORIN	0
LORAZEPAM	0
MAGNESIUM	0
MANNITOL	0
MEPERIDINE	0

Table 6: Percentage of missing data in each covariate in the original DFCI dataset.

Feature	% Missing
MESNA	0
METHYLPREDNISOLONE	0
METOCLOPRAMIDE	0
NS	0
ONDANSETRON	0
OXALIPLATIN	0
OXYCODONE	0
PACLITAXEL	0
PALONOSETRON	0
PEGFILGRASTIM	0
PEMETREXED	0
POTASSIUM	0
PROCHLORPERAZINE	0
RANITIDINE	0
SECONDARY	0
TRASTUZUMAB	0
VINORELBINE	0
ZOLEDRONIC	0

Table 6: Percentage of missing data in each covariate in the original DFCI dataset.

Feature	% Missing
MDS-UPDRS Total Score	30.19
Hoehn and Yahr Stage (On Stage)	17.45
Hoehn and Yahr Stage (Off Stage)	17.45
TD/PIGD Classification - Original categories	17.45
TD/PIGD Classification - New categories	17.45
TD/PIGD Classification Indeterminate	17.45
TD/PIGD Classification PIGD	17.45
TD/PIGD Classification TD	17.45
MDS-UPDRS Part III Score	17.45
Total Rigidity Score	17.39
Tremor Score	17.39
MDS-UPDRS Part III Score	17.39
APOE Genotype - number of e4 alleles	9.7
Change in Diagnosis	5.04
Primary Diagnosis: Corticobasal Degeneration	5.04
Primary Diagnosis: Dementia with Lewy bodies	5.04
Primary Diagnosis: Idiopathic Parkinson's Disease	5.04
Primary Diagnosis: Multiple System Atrophy	5.04
Primary Diagnosis: No Parkinson's Disease Nor Other Neurological Disorder	5.04
Serum Uric Acid	4.33
SCOPA-AUT Total Score	1.42
SCOPA-AUT Gastrointestinal (GI) Score	0.97
Benton Judgement of Line Orientation Score	0.84
HVLT Delayed Recognition	0.84
HVLT False Alarms	0.84
HLVT Discrimination	0.84
Right caudate	0.78
Left caudate	0.78
Right putamen	0.78
Left putamen	0.78
REM Sleep Behavior Disorder Questionnaire Score	0.65
Categorical REM Sleep Behavior Disorder	0.65
Symbol Digit Modalities Score	0.65

Table 7: Percentage of missing data in each covariate in the original PPMI dataset.

Feature	% Missing
HVLT Delayed Recall	0.58
HLVT Retention	0.58
Letter Number Sequencing Score	0.58
SCOPA-AUT Sexual Dysfunction Score	0.58
Semantic Fluency Score - Animal subscore	0.58
Semantic Fluency Score - Vegetable subscore	0.58
Semantic Fluency Score - Fruit subscore	0.58
Semantic Fluency Total Score	0.58
HVLT Immediate/Total Recall	0.52
STAI Trait Sub-score	0.52
STAI Total Score	0.52
Epworth Sleepiness Scale Score	0.45
Categorical Epworth Sleepiness Scale Score	0.45
STAI State Sub-score	0.45
QUIP disorder - Hobbies	0.39
QUIP disorder - Punding	0.39
QUIP disorder - Walking or Driving	0.39
Total Rigidity Score	0.39
Use of Dopamine Agonist	0.32
Use of Dopamine Agonist and Other PD Medication	0.32
Use of Levodopa	0.32
Use of Levodopa and Dopamine Agonist	0.32
Use of Levodopa and Dopamine Agonist and Other PD Medication	0.32
Use of Levodopa and Other PD Medication	0.32
Use of Other PD Medication	0.32
Unmedicated for PD	0.32
MDS-UPDRS Part I Score	0.32
MDS-UPDRS Part I Fatigue	0.32
MDS-UPDRS Part II Score	0.32
Geriatric Depression Scale Score	0.32
Categorical Geriatric Depression Scale Score	0.32
QUIP disorder - Gambling	0.32
QUIP disorder - Sex	0.32
QUIP disorder - Buying	0.32
QUIP disorder - Eating	0.32
General QUIP Score	0.32
Any QUIP Disorder	0.32
SCOPA-AUT Cardiovascular Score	0.32
SCOPA-AUT Thermoregulatory Score	0.32
MDS-UPDRS Part I Cognitive Impairment	0.26
MDS-UPDRS Part I Hallucinations and Psychosis	0.26
MDS-UPDRS Part I Depressed Mood	0.26
MDS-UPDRS Part I Anxious Mood	0.26
MDS-UPDRS Part I Apathy Mood	0.26
MDS-UPDRS Part I Features of Dopamine Dysregulation Syndrome	0.26
SCOPA-AUT Urinary Score	0.26
SCOPA-AUT Pupillomotor Score	0.26
Family History of Parkinson's Disease	0.26
Initial symptom (at diagnosis) - Postural Instability	0.26
Initial symptom (at diagnosis) - Other	0.26
Age	0
Gender - Female	0
Gender - Male	0
Years of Education	0
Race - Hispanic/Latino	0
Race - Non Hispanic/Non Latino	0

Table 7: Percentage of missing data in each covariate in the original PPMI dataset.

Feature	% Missing
Race - Asian	0
Race - Black	0
Race - Other	0
Race - White	0
Duration of Parkinson's Disease	0
Age Onset of Parkinson's Disease	0
Age at Diagnosis	0
Brain Side Most Affected at Parkinson's Disease Onset	0
Initial symptom (at diagnosis) - Resting Tremor	0
Initial symptom (at diagnosis) - Rigidity	0
Initial symptom (at diagnosis) - Bradykinesia	0
Missing initial symptoms	0
SNCA rs356181 Genotype - C.C	0
SNCA rs356181 Genotype - C.T	0
SNCA rs356181 Genotype - T.T	0
SNCA rs3910105 Genotype - C.C	0
SNCA rs3910105 Genotype - C.T	0
SNCA rs3910105 Genotype - T.T	0
MAPT Genotype - H1H1	0
MAPT Genotype - H1H2	0
MAPT Genotype - H2H2	0
Ipsliateral caudate	0
Ipsilateral striatum	0
Left striatum	0

Table 7: Percentage of missing data in each covariate in the original PPMI dataset.

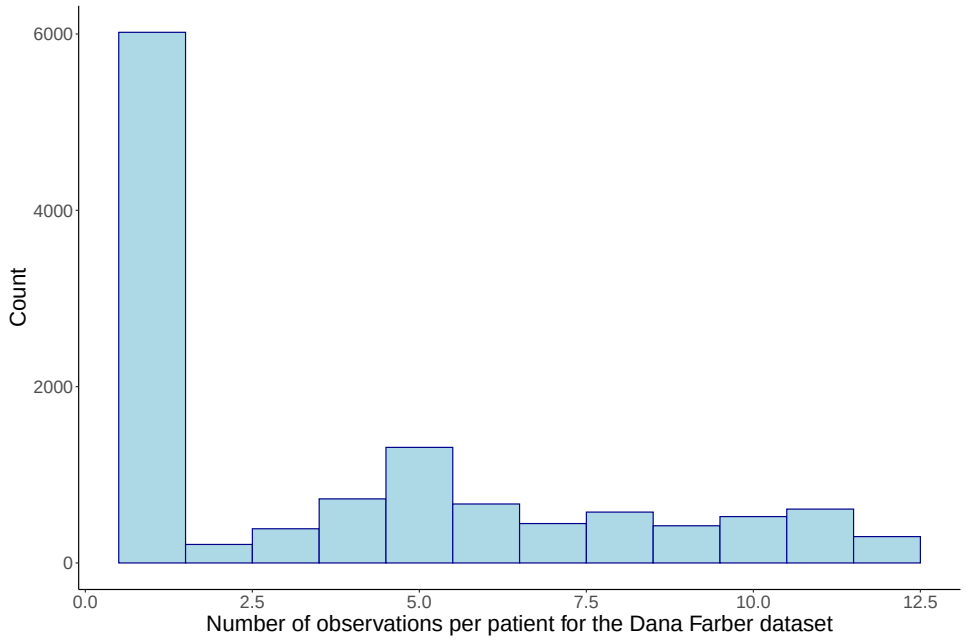

Fig. 13: Histogram of the number of observations per patient in the DFCI dataset.

$d_i = t_i - t_{i+1}$	**Mean**	**Standard Deviation**	**Median**	**Min**	**Max**	**Range**	**Skew**
$d_1 = t_1 - t_2$	19.81	6.92	21.00	1.00	30.00	29.00	-0.61
$d_2 = t_2 - t_3$	24.66	7.10	22.00	7.00	42.00	35.00	0.16
$d_3 = t_3 - t_4$	29.75	10.47	28.00	14.00	42.00	28.00	-0.12
$d_4 = t_4 - t_5$	68.02	22.88	70.00	13.50	117.00	103.50	-0.49
$d_5 = t_5 - t_6$	89.76	26.38	85.00	17.50	163.00	145.50	0.21
$d_6 = t_6 - t_7$	98.79	35.76	98.00	21.00	175.00	154.00	0.04
$d_7 = t_7 - t_8$	130.85	51.70	126.00	27.00	266.00	239.00	0.45
$d_8 = t_8 - t_9$	177.60	70.40	177.50	30.50	329.00	298.50	0.20
$d_9 = t_9 - t_{10}$	230.44	101.10	226.00	14.00	518.00	504.00	0.25
$d_{10} = t_{10} - t_{11}$	549.97	268.29	525.00	88.00	1344.00	1256.00	0.90
$d_{11} = t_{11} - t_{12}$	1236.95	626.50	1157.63	70.00	3022.00	2952.00	0.82

Table 8: Summary statistics of inter-visit time intervals (in days) for the DFCI dataset, where t_i is the time of visit i and t_1 corresponds to the most recent visit for each patient. We consider 11 intervals because the maximum number of visits per patient is 12 in the DFCI dataset.

7.2 Supplemental Experimental Results

This section provides detailed results from the computational experiments on the real-world datasets. Each table refers to either the imputation accuracy or downstream predictive performance for all FHS, DFCI, and PPMI. Tables 9 and 10 refer to the MCAR experiments when we vary the percentage of missing data from 10% to 50%. Tables 11 and 12 show the performance of all methods when we vary the number of observations per patient. Tables 13 and 14 focus on the experiments where we vary the ratio of the missing data mechanism from $\gamma = 0$ (30% MCAR, 0% MNAR) to $\gamma = 1$ (0% MCAR, 30% MNAR).

FHS

	MAE (Standard Error)						
Missing %	`med.knn`	`mice`	`moving.avg`	`amelia`	`bpca`	`mean`	`opt.knn`
10	0.251 (0.011)	0.58 (0.017)	0.578 (0.013)	0.656 (0.017)	0.554 (0.009)	0.543 (0.013)	0.505 (0.009)
20	0.247 (0.006)	0.581 (0.015)	0.569 (0.008)	0.646 (0.015)	0.549 (0.009)	0.532 (0.009)	0.502 (0.010)
30	0.262 (0.007)	0.593 (0.014)	0.576 (0.008)	0.657 (0.013)	0.555 (0.012)	0.537 (0.008)	0.514 (0.011)
40	0.273 (0.006)	0.605 (0.012)	0.578 (0.006)	0.661 (0.016)	0.544 (0.010)	0.537 (0.006)	0.508 (0.009)
50	0.289 (0.005)	0.616 (0.010)	0.582 (0.004)	0.669 (0.011)	0.535 (0.005)	0.535 (0.005)	0.503 (0.005)

DFCI

	MAE (Standard Error)						
Missing %	`med.knn`	`mice`	`amelia`	`moving.avg`	`bpca`	`mean`	`opt.knn`
10	3.076 (0.583)	4.716 (1.032)	9.627 (2.385)	4.994 (0.614)	10 (4.089)	4.248 (0.603)	4.887 (1.038)
20	3.263 (0.513)	4.847 (0.787)	10.148 (2.383)	5.125 (0.506)	6.388 (3.654)	4.34 (0.478)	4.954 (0.799)
30	3.331 (0.339)	5.006 (0.563)	10.421 (2.144)	5.167 (0.313)	4.351 (0.272)	4.351 (0.272)	4.875 (0.420)
40	3.42 (0.214)	4.866 (0.316)	10.122 (2.617)	5.222 (0.193)	4.668 (1.581)	4.351 (0.143)	5.062 (0.447)
50	3.568 (0.156)	5.139 (0.407)	10.126 (1.955)	5.342 (0.107)	4.679 (1.570)	4.367 (0.049)	5.044 (0.466)

PPMI

	MAE (Standard Error)						
Missing %	`med.knn`	`mice`	`amelia`	`moving.avg`	`bpca`	`mean`	`opt.knn`
10	1.046 (0.087)	1.368 (0.097)	2.592 (0.140)	2.143 (0.079)	2.901 (0.250)	2.256 (0.088)	2.049 (0.206)
20	1.101 (0.032)	1.58 (0.083)	2.538 (0.105)	2.214 (0.041)	2.713 (0.143)	2.236 (0.073)	1.884 (0.159)
30	1.111 (0.030)	1.784 (0.086)	2.522 (0.070)	2.201 (0.027)	2.604 (0.134)	2.214 (0.063)	1.827 (0.061)
40	1.189 (0.035)	1.934 (0.050)	2.555 (0.085)	2.278 (0.033)	2.432 (0.123)	2.207 (0.050)	1.856 (0.048)
50	1.286 (0.025)	2.188 (0.025)	2.565 (0.610)	2.361 (0.025)	2.422 (0.046)	2.228 (0.034)	1.99 (0.035)

Table 9: Imputation errors for each method using the MAE metric on the FHS, DFCI, and PPMI datasets, varying the percentage of missing data from 10% to 50%. The missing data mechanism is fixed to MCAR. For an illustration, see Figure 1.

FHS

	AUC (Standard Error)						
Missing %	med.knn	mice	moving.avg	amelia	bpca	mean	opt.knn
10	0.897 (0.018)	0.868 (0.023)	0.867 (0.021)	0.86 (0.027)	0.878 (0.017)	0.869 (0.021)	0.874 (0.019)
20	0.895 (0.018)	0.853 (0.025)	0.857 (0.020)	0.831 (0.031)	0.861 (0.023)	0.849 (0.024)	0.857 (0.023)
30	0.884 (0.021)	0.824 (0.028)	0.843 (0.021)	0.793 (0.034)	0.827 (0.026)	0.818 (0.029)	0.819 (0.028)
40	0.877 (0.017)	0.805 (0.026)	0.835 (0.020)	0.78 (0.025)	0.805 (0.030)	0.801 (0.027)	0.786 (0.036)
50	0.864 (0.016)	0.772 (0.024)	0.826 (0.018)	0.734 (0.042)	0.771 (0.022)	0.771 (0.022)	0.75 (0.031)

DFCI

	AUC (Standard Error)						
Missing %	med.knn	mice	amelia	moving.avg	bpca	mean	opt.knn
10	0.915 (0.012)	0.914 (0.013)	0.91 (0.014)	0.895 (0.013)	0.913 (0.013)	0.914 (0.013)	0.913 (0.013)
20	0.909 (0.014)	0.907 (0.015)	0.901 (0.014)	0.89 (0.014)	0.907 (0.015)	0.907 (0.015)	0.905 (0.015)
30	0.902 (0.014)	0.899 (0.017)	0.889 (0.017)	0.885 (0.015)	0.899 (0.016)	0.899 (0.016)	0.897 (0.016)
40	0.895 (0.015)	0.89 (0.018)	0.89 (0.018)	0.879 (0.015)	0.893 (0.016)	0.893 (0.016)	0.888 (0.016)
50	0.889 (0.017)	0.879 (0.019)	0.881 (0.017)	0.872 (0.016)	0.884 (0.018)	0.884 (0.018)	0.879 (0.018)

PPMI

	MAE (Standard Error)						
Missing %	med.knn	mice	amelia	moving.avg	bpca	mean	opt.knn
10	1.208 (0.176)	1.325 (0.132)	1.657 (0.170)	1.475 (0.157)	1.608 (0.176)	1.838 (0.726)	1.555 (0.179)
20	1.624 (0.170)	1.788 (0.111)	1.861 (0.116)	1.898 (0.170)	1.802 (0.106)	1.876 (0.113)	1.802 (0.099)
30	1.785 (0.108)	1.923 (0.084)	1.939 (0.079)	2.054 (0.134)	1.938 (0.075)	1.955 (0.087)	1.909 (0.080)
40	1.822 (0.119)	2.027 (0.070)	2.025 (0.085)	2.158 (0.157)	1.982 (0.108)	2.01 (0.094)	1.968 (0.095)
50	1.916 (0.067)	2.13 (0.045)	2.114 (0.087)	2.217 (0.075)	2.097 (0.077)	2.122 (0.057)	2.092 (0.072)

Table 10: Downstream accuracy results for each method on the FHS, DFCI, and PPMI datasets, varying the percentage of missing data from 10% to 50% according to the MCAR mechanism. For an illustration, see Figure 8.

FHS

	MAE (Standard Error)						
OPP	med.knn	mice	amelia	moving.avg	bpca	mean	opt.knn
1	0.489 (0.006)	0.593 (0.010)	0.667 (0.017)	0.598 (0.004)	0.52 (0.014)	0.498 (0.004)	0.472 (0.013)
2	0.371 (0.006)	0.592 (0.008)	0.661 (0.014)	0.477 (0.006)	0.499 (0.006)	0.498 (0.006)	0.46 (0.007)
3	0.318 (0.006)	0.585 (0.008)	0.665 (0.018)	0.431 (0.006)	0.496 (0.004)	0.497 (0.004)	0.459 (0.004)
4	0.299 (0.005)	0.591 (0.006)	0.665 (0.015)	0.418 (0.006)	0.501 (0.005)	0.499 (0.004)	0.463 (0.004)
5	0.291 (0.005)	0.59 (0.010)	0.656 (0.018)	0.418 (0.004)	0.504 (0.003)	0.502 (0.005)	0.466 (0.006)
6	0.286 (0.005)	0.593 (0.008)	0.66 (0.016)	0.427 (0.004)	0.504 (0.005)	0.504 (0.005)	0.469 (0.005)
7	0.286 (0.005)	0.599 (0.009)	0.661 (0.014)	0.442 (0.005)	0.508 (0.006)	0.511 (0.005)	0.478 (0.006)
8	0.289 (0.004)	0.607 (0.010)	0.665 (0.012)	0.455 (0.004)	0.519 (0.002)	0.519 (0.004)	0.486 (0.005)
9	0.295 (0.003)	0.611 (0.009)	0.666 (0.011)	0.47 (0.005)	0.53 (0.006)	0.528 (0.006)	0.495 (0.007)
10	0.289 (0.004)	0.615 (0.009)	0.668 (0.011)	0.482 (0.004)	0.534 (0.005)	0.534 (0.005)	0.501 (0.006)

DFCI

	MAE (Standard Error)						
OPP	med.knn	mice	amelia	moving.avg	bpca	mean	opt.knn
1	5.092 (0.190)	4.866 (0.579)	7.048 (2.543)	6.063 (0.065)	4.75 (0.806)	4.479 (0.066)	4.981 (0.666)
2	4.859 (0.290)	5.055 (0.853)	7.261 (2.569)	5.961 (0.087)	4.842 (0.906)	4.528 (0.075)	4.972 (0.650)
3	4.581 (0.154)	4.954 (0.673)	7.252 (2.003)	5.822 (0.117)	4.825 (0.917)	4.516 (0.083)	4.926 (0.648)
4	4.195 (0.170)	4.887 (0.655)	7.339 (2.008)	5.566 (0.098)	4.648 (0.874)	4.341 (0.057)	4.637 (0.482)
5	3.735 (0.183)	4.715 (0.474)	6.812 (1.664)	5.333 (0.083)	4.49 (1.001)	4.142 (0.050)	4.594 (0.495)
6	3.598 (0.194)	5.006 (0.594)	8.188 (2.299)	5.282 (0.112)	4.7 (1.311)	4.241 (0.051)	4.683 (0.457)
7	3.601 (0.205)	4.952 (0.523)	8.178 (2.212)	5.312 (0.108)	4.72 (1.314)	4.267 (0.048)	4.731 (0.415)
8	3.749 (0.099)	5 (0.456)	9.01 (2.149)	5.34 (0.104)	4.911 (1.635)	4.345 (0.047)	4.977 (0.475)
9	3.9 (0.131)	5.292 (0.391)	9.613 (2.236)	5.572 (0.127)	5.28 (2.080)	4.562 (0.063)	5.117 (0.237)
10	3.717 (0.129)	5.198 (0.420)	10.579 (2.651)	5.444 (0.104)	5.225 (2.241)	4.484 (0.058)	5.027 (0.455)
11	3.574 (0.140)	5.086 (0.335)	9.84 (2.205)	5.373 (0.106)	5.271 (2.444)	4.423 (0.055)	5.013 (0.369)
12	3.568 (0.156)	5.139 (0.407)	10.126 (1.955)	5.342 (0.107)	4.679 (1.570)	4.367 (0.049)	5.044 (0.466)

PPMI

	MAE (Standard Error)						
OPP	med.knn	mice	amelia	moving.avg	bpca	mean	opt.knn
1	2.106 (0.035)	2.336 (0.053)	3.23 (0.964)	3.222 (0.019)	2.556 (0.058)	2.202 (0.019)	2.317 (0.197)
2	1.658 (0.032)	2.236 (0.051)	2.876 (1.429)	2.717 (0.034)	2.308 (0.040)	2.198 (0.019)	2.016 (0.087)
3	1.394 (0.024)	2.171 (0.082)	2.623 (1.243)	2.458 (0.023)	2.359 (0.063)	2.194 (0.060)	2.079 (0.173)
4	1.284 (0.025)	2.181 (0.021)	2.605 (0.660)	2.358 (0.026)	2.42 (0.038)	2.234 (0.031)	1.996 (0.028)

Table 11: Imputation errors for each method using the MAE metric on the FHS, DFCI, and PPMI datasets, varying the time horizon which determines the number of observations per patient. The missing data mechanism is fixed to MCAR, and the total percentage of missing data is fixed to 50%. For an illustration, see Figure 3.

FHS

	AUC (Standard Error)						
OPP	med.knn	mice	amelia	moving.avg	bpca	mean	opt.knn
1	0.778 (0.029)	0.789 (0.028)	0.742 (0.034)	0.786 (0.026)	0.819 (0.023)	0.806 (0.026)	0.798 (0.020)
2	0.835 (0.024)	0.777 (0.027)	0.74 (0.030)	0.815 (0.024)	0.8 (0.035)	0.79 (0.026)	0.781 (0.030)
3	0.868 (0.023)	0.79 (0.032)	0.738 (0.037)	0.842 (0.019)	0.806 (0.040)	0.804 (0.032)	0.797 (0.034)
4	0.864 (0.026)	0.772 (0.025)	0.736 (0.040)	0.837 (0.022)	0.786 (0.021)	0.781 (0.031)	0.764 (0.033)
5	0.862 (0.020)	0.782 (0.036)	0.741 (0.030)	0.839 (0.022)	0.799 (0.017)	0.788 (0.033)	0.778 (0.035)
6	0.865 (0.014)	0.778 (0.024)	0.745 (0.030)	0.84 (0.017)	0.783 (0.032)	0.783 (0.024)	0.77 (0.027)
7	0.86 (0.028)	0.775 (0.032)	0.732 (0.034)	0.838 (0.025)	0.773 (0.027)	0.781 (0.031)	0.764 (0.031)
8	0.882 (0.015)	0.769 (0.036)	0.733 (0.034)	0.835 (0.026)	0.785 (0.024)	0.779 (0.032)	0.757 (0.037)
9	0.873 (0.018)	0.772 (0.038)	0.743 (0.035)	0.827 (0.022)	0.787 (0.030)	0.773 (0.040)	0.757 (0.037)
10	0.864 (0.016)	0.769 (0.024)	0.735 (0.037)	0.827 (0.018)	0.768 (0.021)	0.768 (0.021)	0.749 (0.029)

DFCI

	AUC (Standard Error)						
OPP	med.knn	mice	amelia	moving.avg	bpca	mean	opt.knn
1	0.845 (0.017)	0.857 (0.016)	0.854 (0.012)	0.835 (0.016)	0.854 (0.017)	0.854 (0.016)	0.846 (0.017)
2	0.838 (0.023)	0.844 (0.015)	0.831 (0.015)	0.83 (0.017)	0.842 (0.018)	0.842 (0.019)	0.835 (0.019)
3	0.84 (0.021)	0.845 (0.019)	0.838 (0.021)	0.83 (0.020)	0.841 (0.020)	0.841 (0.020)	0.836 (0.021)
4	0.847 (0.017)	0.845 (0.017)	0.845 (0.020)	0.836 (0.018)	0.843 (0.020)	0.844 (0.018)	0.84 (0.020)
5	0.862 (0.015)	0.85 (0.017)	0.855 (0.015)	0.85 (0.017)	0.852 (0.017)	0.852 (0.017)	0.844 (0.019)
6	0.872 (0.014)	0.858 (0.015)	0.859 (0.020)	0.858 (0.015)	0.865 (0.017)	0.865 (0.017)	0.853 (0.018)
7	0.878 (0.014)	0.862 (0.015)	0.865 (0.018)	0.864 (0.014)	0.867 (0.016)	0.867 (0.015)	0.858 (0.016)
8	0.878 (0.019)	0.866 (0.018)	0.873 (0.017)	0.866 (0.016)	0.871 (0.017)	0.871 (0.017)	0.861 (0.019)
9	0.882 (0.012)	0.872 (0.015)	0.865 (0.012)	0.867 (0.013)	0.873 (0.016)	0.873 (0.016)	0.867 (0.017)
10	0.889 (0.015)	0.878 (0.018)	0.9 (0.014)	0.872 (0.016)	0.881 (0.019)	0.881 (0.019)	0.873 (0.020)
11	0.89 (0.014)	0.881 (0.017)	0.906 (0.026)	0.874 (0.015)	0.886 (0.016)	0.886 (0.016)	0.881 (0.017)
12	0.889 (0.017)	0.879 (0.019)	0.881 (0.017)	0.872 (0.016)	0.884 (0.018)	0.884 (0.018)	0.879 (0.018)

PPMI

	MAE (Standard Error)						
OPP	med.knn	mice	amelia	moving.avg	bpca	mean	opt.knn
1	2.116 (0.076)	2.183 (0.058)	2.12 (0.079)	2.249 (0.094)	2.08 (0.063)	2.104 (0.098)	2.087 (0.107)
2	1.982 (0.102)	2.141 (0.047)	2.094 (0.115)	2.167 (0.088)	2.101 (0.075)	2.1 (0.099)	2.089 (0.081)
3	1.899 (0.089)	2.09 (0.062)	2.075 (0.090)	2.071 (0.081)	2.03 (0.121)	2.029 (0.120)	2.009 (0.082)
4	1.923 (0.040)	2.138 (0.032)	2.126 (0.078)	2.076 (0.063)	2.101 (0.058)	2.126 (0.034)	2.097 (0.050)

Table 12: Downstream accuracy results for each method on the FHS, DFCI, and PPMI datasets, varying the time horizon which determines the number of observations per patient. The missing data mechanism is fixed to MCAR, and the total percentage of missing data is fixed to 50%. For an illustration, see Figure 9.

FHS

	MAE (Standard Error)						
Gamma	med.knn	mice	moving.avg	amelia	bpca	mean	opt.knn
0.1	0.261 (0.005)	0.602 (0.031)	0.473 (0.005)	0.672 (0.027)	0.562 (0.018)	0.547 (0.021)	0.519 (0.020)
0.2	0.26 (0.004)	0.612 (0.036)	0.474 (0.004)	0.681 (0.027)	0.571 (0.031)	0.554 (0.026)	0.527 (0.027)
0.3	0.259 (0.005)	0.621 (0.043)	0.471 (0.008)	0.681 (0.035)	0.565 (0.026)	0.555 (0.029)	0.523 (0.028)
0.4	0.262 (0.003)	0.603 (0.040)	0.474 (0.004)	0.671 (0.030)	0.558 (0.026)	0.548 (0.026)	0.515 (0.028)
0.5	0.261 (0.006)	0.623 (0.043)	0.478 (0.007)	0.695 (0.046)	0.573 (0.032)	0.563 (0.037)	0.532 (0.032)
0.6	0.261 (0.004)	0.627 (0.067)	0.472 (0.005)	0.689 (0.061)	0.569 (0.039)	0.559 (0.042)	0.527 (0.038)
0.7	0.262 (0.006)	0.614 (0.051)	0.473 (0.005)	0.677 (0.055)	0.559 (0.042)	0.549 (0.043)	0.517 (0.041)
0.8	0.262 (0.006)	0.59 (0.012)	0.473 (0.005)	0.662 (0.028)	0.558 (0.011)	0.538 (0.013)	0.516 (0.013)
0.9	0.262 (0.006)	0.59 (0.012)	0.473 (0.005)	0.654 (0.014)	0.558 (0.011)	0.538 (0.013)	0.516 (0.013)
1	0.262 (0.006)	0.59 (0.012)	0.473 (0.005)	0.656 (0.012)	0.558 (0.011)	0.538 (0.013)	0.516 (0.013)

DFCI

	MAE (Standard Error)						
Gamma	med.knn	mice	amelia	moving.avg	bpca	mean	opt.knn
0.1	5.11 (0.082)	5.862 (0.104)	7.043 (0.278)	6.353 (0.073)	7.16 (0.107)	5.978 (0.056)	6.214 (0.200)
0.2	5.41 (0.087)	6.182 (0.110)	7.105 (0.449)	6.598 (0.085)	7.429 (0.169)	6.243 (0.054)	6.464 (0.181)
0.3	5.608 (0.091)	6.376 (0.098)	7.246 (0.290)	6.754 (0.086)	7.593 (0.200)	6.421 (0.058)	6.615 (0.133)
0.4	5.781 (0.097)	6.545 (0.085)	7.291 (0.257)	6.908 (0.093)	7.643 (0.245)	6.567 (0.058)	6.777 (0.120)
0.5	5.922 (0.100)	6.685 (0.085)	7.462 (0.197)	7.026 (0.097)	7.69 (0.280)	6.685 (0.056)	6.878 (0.112)
0.6	6.045 (0.095)	6.801 (0.079)	7.532 (0.173)	7.138 (0.090)	7.726 (0.298)	6.785 (0.056)	6.968 (0.104)
0.7	6.144 (0.096)	6.892 (0.078)	7.581 (0.211)	7.225 (0.092)	7.758 (0.326)	6.872 (0.057)	7.06 (0.117)
0.8	6.255 (0.101)	7.002 (0.084)	7.601 (0.179)	7.328 (0.093)	7.798 (0.366)	6.958 (0.062)	7.158 (0.098)
0.9	6.347 (0.103)	7.1 (0.080)	7.671 (0.192)	7.406 (0.096)	7.814 (0.372)	7.03 (0.060)	7.251 (0.087)
1	6.438 (0.104)	7.204 (0.075)	7.709 (0.200)	7.486 (0.097)	7.903 (0.354)	7.097 (0.058)	7.294 (0.092)

PPMI

	MAE (Standard Error)						
Gamma	med.knn	mice	amelia	moving.avg	bpca	mean	opt.knn
0.1	1.302 (0.050)	2.11 (0.065)	2.774 (0.091)	2.359 (0.051)	2.815 (0.105)	2.485 (0.061)	2.179 (0.114)
0.2	1.459 (0.038)	2.269 (0.088)	2.87 (0.102)	2.502 (0.029)	2.973 (0.115)	2.659 (0.072)	2.373 (0.183)
0.3	1.636 (0.043)	2.41 (0.058)	3.01 (0.116)	2.667 (0.055)	3.104 (0.072)	2.835 (0.028)	2.507 (0.119)
0.4	1.78 (0.039)	2.571 (0.112)	3.051 (0.130)	2.801 (0.044)	3.203 (0.100)	2.977 (0.073)	2.679 (0.159)
0.5	1.91 (0.051)	2.641 (0.092)	3.161 (0.120)	2.935 (0.047)	3.308 (0.122)	3.08 (0.054)	2.761 (0.069)
0.6	2.006 (0.057)	2.778 (0.077)	3.308 (0.144)	3.013 (0.057)	3.343 (0.143)	3.19 (0.074)	2.904 (0.087)
0.7	2.117 (0.029)	2.931 (0.100)	3.424 (0.099)	3.126 (0.027)	3.489 (0.094)	3.311 (0.049)	3.052 (0.049)
0.8	2.246 (0.034)	3.076 (0.078)	3.538 (0.112)	3.258 (0.036)	3.655 (0.121)	3.459 (0.055)	3.199 (0.065)
0.9	2.374 (0.043)	3.225 (0.071)	3.736 (0.480)	3.385 (0.037)	3.752 (0.158)	3.557 (0.057)	3.339 (0.069)
1	2.487 (0.047)	3.375 (0.095)	3.883 (0.535)	3.496 (0.041)	3.856 (0.202)	3.664 (0.070)	3.564 (0.220)

Table 13: Imputation errors for each method on the FHS, DFCI, and PPMI datasets, varying the ratio of the missing data mechanism from $\gamma = 0$ (30% MCAR, 0% MNAR) to $\gamma = 1$ (0% MCAR, 30% MNAR). For an illustration, see Figure 6.

FHS

	AUC (Standard Error)						
Gamma	med.knn	mice	moving.avg	amelia	bpca	mean	opt.knn
0.1	0.89 (0.015)	0.841 (0.024)	0.861 (0.021)	0.808 (0.028)	0.846 (0.028)	0.836 (0.023)	0.838 (0.027)
0.2	0.897 (0.021)	0.843 (0.036)	0.866 (0.025)	0.816 (0.038)	0.847 (0.036)	0.841 (0.034)	0.841 (0.037)
0.3	0.89 (0.014)	0.857 (0.018)	0.859 (0.024)	0.824 (0.023)	0.859 (0.022)	0.851 (0.020)	0.85 (0.018)
0.4	0.892 (0.019)	0.854 (0.025)	0.86 (0.023)	0.831 (0.032)	0.857 (0.025)	0.849 (0.023)	0.848 (0.028)
0.5	0.891 (0.021)	0.858 (0.018)	0.857 (0.025)	0.84 (0.024)	0.862 (0.015)	0.855 (0.015)	0.854 (0.015)
0.6	0.889 (0.020)	0.865 (0.025)	0.86 (0.029)	0.834 (0.028)	0.867 (0.018)	0.86 (0.021)	0.859 (0.023)
0.7	0.894 (0.019)	0.865 (0.022)	0.868 (0.020)	0.845 (0.031)	0.868 (0.020)	0.861 (0.023)	0.862 (0.022)
0.8	0.894 (0.019)	0.848 (0.024)	0.868 (0.020)	0.835 (0.039)	0.854 (0.015)	0.847 (0.020)	0.848 (0.022)
0.9	0.894 (0.019)	0.848 (0.024)	0.868 (0.020)	0.812 (0.019)	0.854 (0.015)	0.847 (0.020)	0.848 (0.022)
1	0.894 (0.019)	0.848 (0.024)	0.868 (0.020)	0.825 (0.018)	0.854 (0.015)	0.847 (0.020)	0.848 (0.022)

DFCI

	AUC (Standard Error)						
Gamma	med.knn	mice	amelia	moving.avg	bpca	mean	opt.knn
0.1	0.916 (0.013)	0.915 (0.014)	0.902 (0.015)	0.909 (0.012)	0.912 (0.014)	0.91 (0.014)	0.907 (0.014)
0.2	0.916 (0.012)	0.915 (0.014)	0.901 (0.016)	0.909 (0.012)	0.91 (0.014)	0.909 (0.014)	0.906 (0.014)
0.3	0.915 (0.012)	0.912 (0.014)	0.902 (0.015)	0.908 (0.012)	0.909 (0.013)	0.909 (0.014)	0.905 (0.013)
0.4	0.914 (0.012)	0.91 (0.014)	0.899 (0.014)	0.908 (0.011)	0.909 (0.013)	0.909 (0.013)	0.905 (0.014)
0.5	0.914 (0.013)	0.909 (0.015)	0.898 (0.016)	0.907 (0.012)	0.908 (0.014)	0.908 (0.014)	0.903 (0.015)
0.6	0.913 (0.012)	0.908 (0.014)	0.895 (0.016)	0.908 (0.011)	0.907 (0.014)	0.908 (0.014)	0.902 (0.015)
0.7	0.912 (0.012)	0.907 (0.015)	0.897 (0.016)	0.909 (0.011)	0.906 (0.013)	0.909 (0.012)	0.902 (0.013)
0.8	0.913 (0.012)	0.906 (0.014)	0.897 (0.014)	0.906 (0.011)	0.905 (0.013)	0.905 (0.013)	0.902 (0.014)
0.9	0.912 (0.012)	0.905 (0.014)	0.897 (0.014)	0.903 (0.012)	0.903 (0.014)	0.901 (0.013)	0.902 (0.014)
1	0.912 (0.012)	0.904 (0.015)	0.897 (0.013)	0.903 (0.011)	0.904 (0.014)	0.902 (0.013)	0.902 (0.014)

PPMI

	MAE (Standard Error)						
Gamma	med.knn	mice	amelia	moving.avg	bpca	mean	opt.knn
0.1	1.851 (0.104)	2 (0.065)	2.037 (0.112)	2.061 (0.098)	2.01 (0.109)	2.065 (0.093)	2.016 (0.097)
0.2	1.905 (0.072)	2.04 (0.073)	2.084 (0.115)	2.115 (0.081)	2.066 (0.086)	2.126 (0.087)	2.094 (0.046)
0.3	1.905 (0.055)	2.039 (0.064)	2.116 (0.090)	2.106 (0.052)	2.066 (0.053)	2.12 (0.088)	2.065 (0.066)
0.4	1.968 (0.064)	2.066 (0.085)	2.163 (0.088)	2.174 (0.064)	2.085 (0.095)	2.171 (0.097)	2.089 (0.083)
0.5	2.011 (0.084)	2.152 (0.048)	2.223 (0.092)	2.212 (0.087)	2.167 (0.059)	2.246 (0.072)	2.186 (0.075)
0.6	2.004 (0.029)	2.169 (0.080)	2.222 (0.073)	2.214 (0.024)	2.2 (0.052)	2.261 (0.074)	2.2 (0.063)
0.7	2.026 (0.066)	2.17 (0.088)	2.259 (0.074)	2.237 (0.054)	2.199 (0.086)	2.274 (0.088)	2.2 (0.068)
0.8	2.031 (0.064)	2.189 (0.065)	2.259 (0.064)	2.243 (0.069)	2.179 (0.061)	2.248 (0.081)	2.214 (0.066)
0.9	2.07 (0.086)	2.235 (0.068)	2.29 (0.076)	2.289 (0.087)	2.24 (0.080)	2.295 (0.081)	2.225 (0.090)
1	2.074 (0.106)	2.245 (0.071)	2.293 (0.066)	2.309 (0.112)	2.257 (0.072)	2.311 (0.066)	2.236 (0.095)

Table 14: Downstream accuracy results for each method on the FHS, DFCI, and PPMI datasets, varying the ratio of the missing data mechanism from $\gamma = 0$ (30% MCAR, 0% MNAR) to $\gamma = 1$ (0% MCAR, 30% MNAR). For an illustration, see Figure 10.

MedImpute Hyperparameter Tuning Results for FHS Experiments

In this section, we present the MedImpute hyperparameters which were selected for the FHS experiments. These hyperparameters were tuned via the custom procedure described in Section 2.4. The covariates in the FHS dataset are:

1. **Afib**: Boolean whether or not the patient has a diagnosis of Atrial Fibrillation.
2. **Age**: Age of the patient (in years).
3. **AHT**: Boolean whether or not the patient has a diagnosis of Arterial Hypertension.
4. **BMI**: Body Mass Index of the patient.
5. **CVD**: Boolean whether or not the patient has a diagnosis of Cardiovascular Disease.
6. **Diabetes**: Boolean whether or not the patient has a diagnosis of Diabetes.
7. **Gender**: Gender of the patient.
8. **Glucose_bl**: Blood glucose level of the patient.
9. **HDL**: High-Density Lipoproteins level of the patient.
10. **LVH**: Boolean whether or not the patient has a diagnosis of Left Ventricular Hypertrophy.
11. **SBP**: Systolic Blood Pressure of the patient.
12. **Smoking**: Categorical variable describing the smoking behavior of the patient.

	Missing %				
Covariate	10	20	30	40	50
Afib	365	365	365	365	365
Age	180	180	180	180	180
AHT	365	365	1000	1000	1000
BMI	365	365	1000	1000	1000
CVD	365	365	365	365	365
diabetes	1000	1000	1000	1000	1000
Gender	1	1	1	1	1
Glucose_bl	1000	1000	1000	1000	1000
HDL	1000	1000	1000	1000	1000
Hemat	1000	1000	1000	1000	1000
LVH	1000	1000	1000	1000	1000
SBP	1000	1000	1000	1000	1000
Smoking	1000	1000	1000	1000	1000

Table 15: Median halflife parameter selected for each covariate in the FHS dataset in the MCAR Missing Percentage experiments with 10 observations per patient.

	Missing %				
Covariate	10	20	30	40	50
Afib	1	1	1	1	1
Age	1	1	1	1	0.95
AHT	1	1	1	1	1
BMI	1	1	1	1	1
CVD	1	1	1	1	1
diabetes	1	1	1	1	1
Gender	1	1	1	1	1
Glucose_bl	0.65	0.6	0.6	0.55	0.5
HDL	0.85	0.85	0.85	0.85	0.85
Hemat	0.8	0.8	0.8	0.75	0.75
LVH	1	1	1	1	1
SBP	0.85	0.85	0.8	0.75	0.75
Smoking	1	1	1	1	1

Table 16: Median alpha parameter selected for each covariate in the FHS dataset in the MCAR Missing Percentage experiments with 10 observations per patient.

	Observations Per Patient (OPP)									
Covariate	1	2	3	4	5	6	7	8	9	10
Afib	1	1	1000	1000	365	365	365	365	365	365
Age	1	1	90	90	180	180	180	180	180	180
AHT	1	1	1000	1000	1000	1000	1000	1000	1000	1000
BMI	1	1	365	365	1000	1000	1000	1000	1000	1000
CVD	1	1	180	365	365	365	365	365	365	365
diabetes	1	1	1000	1000	1000	1000	1000	1000	1000	1000
Gender	1	1	1	1	1	1	1	1	1	1
Glucose_bl	1	1	1000	1000	1000	1000	1000	1000	1000	1000
HDL	1	1	1000	1000	1000	1000	1000	1000	1000	1000
Hemat	1	1	1000	1000	1000	1000	1000	1000	1000	1000
LVH	1	1	1000	1000	1000	1000	1000	1000	1000	1000
SBP	1	1	1000	1000	1000	1000	1000	1000	1000	1000
Smoking	1	1	365	1000	1000	1000	1000	1000	1000	1000

Table 17: Median halflife parameter selected for each covariate in the FHS dataset in the MCAR OPP experiments with 50% missing data.

	Observations Per Patient (OPP)									
Covariate	1	2	3	4	5	6	7	8	9	10
Afib	0	1	1	1	1	1	1	1	1	1
Age	0	0.8	0.8	0.8	0.85	0.85	0.9	0.9	0.9	0.95
AHT	0	1	1	1	1	1	1	1	1	1
BMI	0	0.95	0.95	0.95	0.95	0.95	0.95	0.95	1	1
CVD	0	1	1	1	1	1	1	1	1	1
diabetes	0	1	1	1	1	1	1	1	1	1
Gender	0	1	1	1	1	1	1	1	1	1
Glucose_bl	0	0.6	0.6	0.5	0.55	0.5	0.55	0.55	0.5	0.5
HDL	0	0.9	0.85	0.9	0.9	0.85	0.9	0.85	0.85	0.85
Hemat	0	0.8	0.7	0.7	0.75	0.75	0.75	0.75	0.75	0.75
LVH	0	1	1	1	1	1	1	1	1	1
SBP	0	0.65	0.6	0.65	0.65	0.65	0.7	0.7	0.7	0.75
Smoking	0	1	1	1	1	1	1	1	1	1

Table 18: Median alpha parameter selected for each covariate in the FHS dataset in the MCAR OPP experiments with 50% missing data.

	Gamma										
Covariate	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Afib	365	1000	365	365	365	365	365	365	365	365	365
Age	180	180	180	180	180	180	180	180	180	180	180
AHT	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
BMI	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
CVD	365	365	365	365	365	365	365	365	365	365	365
diabetes	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
Gender	1	1	1	1	1	1	1	1	1	1	1
Glucose_bl	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
HDL	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
Hemat	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
LVH	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
SBP	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
Smoking	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000

Table 19: Median halflife parameter selected for each covariate in the FHS dataset in the MNAR experiments with 50% missing data and 10 observations per patient.

	Gamma										
Covariate	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Afib	1	1	1	1	1	1	1	1	1	1	1
Age	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
AHT	1	1	1	1	1	1	1	1	1	1	1
BMI	1	1	1	1	1	1	1	1	1	1	1
CVD	1	1	1	1	1	1	1	1	1	1	1
diabetes	1	1	1	1	1	1	1	1	1	1	1
Gender	1	1	1	1	1	1	1	1	1	1	1
Glucose_bl	0.5	0.5	0.5	0.5	0.5	0.5	0.55	0.5	0.5	0.5	0.5
HDL	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85
Hemat	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
LVH	1	1	1	1	1	1	1	1	1	1	1
SBP	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
Smoking	1	1	1	1	1	1	1	1	1	1	1

Table 20: Median alpha parameter selected for each covariate in the FHS dataset in the MNAR experiments with 50% missing data and 10 observations per patient.

7.3 Supplemental Synthetic Experiments Results

Figure 14 provides a direct comparison between the Custom Tuning and Grid Search parameter selection processes. We provide results with respect to the imputation error using the RMSE metric.

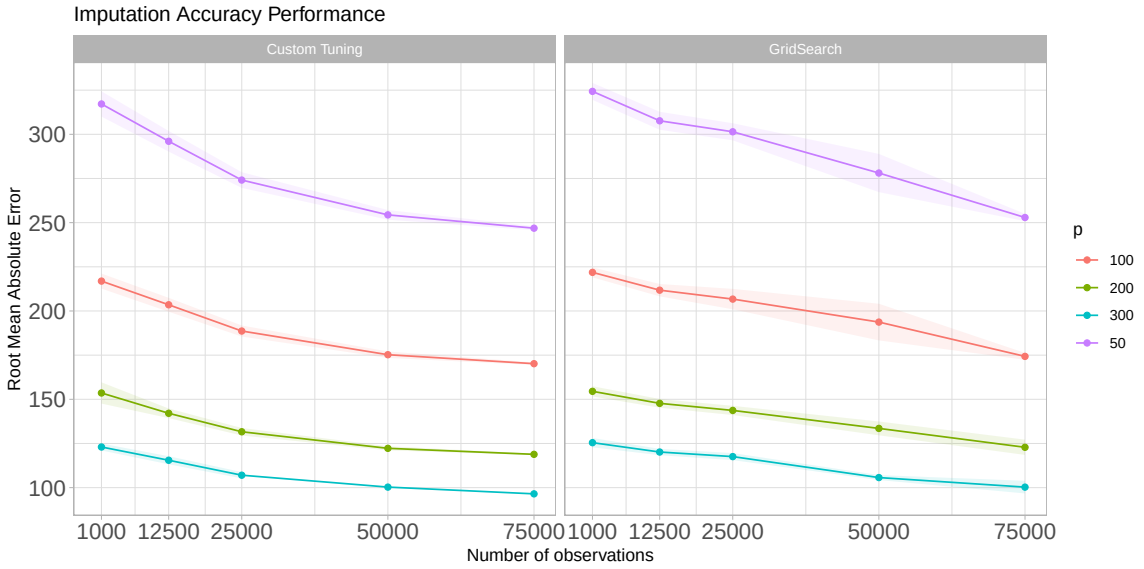

Fig. 14: Average RMSE imputation errors for MedImpute methods on the Synthea dataset using different procedures for hyperparameter tuning, with varying numbers of observations n and features p in the dataset.

7.4 Nemenyi Critical Diagrams

In this section, we show Nemenyi Critical Diagrams (Ismail Fawaz et al. 2019) for the results from the Percentage of Missing Data experiments which are presented in Section 3. These graphs highlight statistically significant differences in the overall rankings of the methods. To generate these diagrams, first the Friedman Rank Test was performed to compare the relative performance of the different imputation methods in each experiment. Second, the Wilcoxon-Holm method was performed to detect pairwise significance. In the diagram for a single experiment, each imputation method is plotted according to its average relative rank on a number line from one to seven. Methods which do not have statistically significant differences in their overall rankings are joined by a horizontal line.

In Figures 15 and 16, we show the Nemenyi Critical Diagrams comparing the methods on the imputation tasks under the MAE and RMSE metrics, respectively. For the FHS and PPMI datasets, we observe that `med.knn` is consistently the top ranked method across all of the experiments. For the DFCI dataset, `med.knn` is consistently top ranked method under the MAE metric, however `med.knn` outperforms the benchmark methods by a smaller margin under the RMSE metric.

In Figure 17, we show the Nemenyi Critical Diagrams comparing the methods on the downstream predictive tasks. Our results demonstrate the edge of the proposed algorithm over the other missing data imputation methods considered. The performance gap is wider in the longitudinal datasets (FHS, PPMI) compared to the DFCI dataset. Nevertheless, even in the latter case, we notice that `med.knn` improves upon the other best performing methods (`mean` and `bpca`).

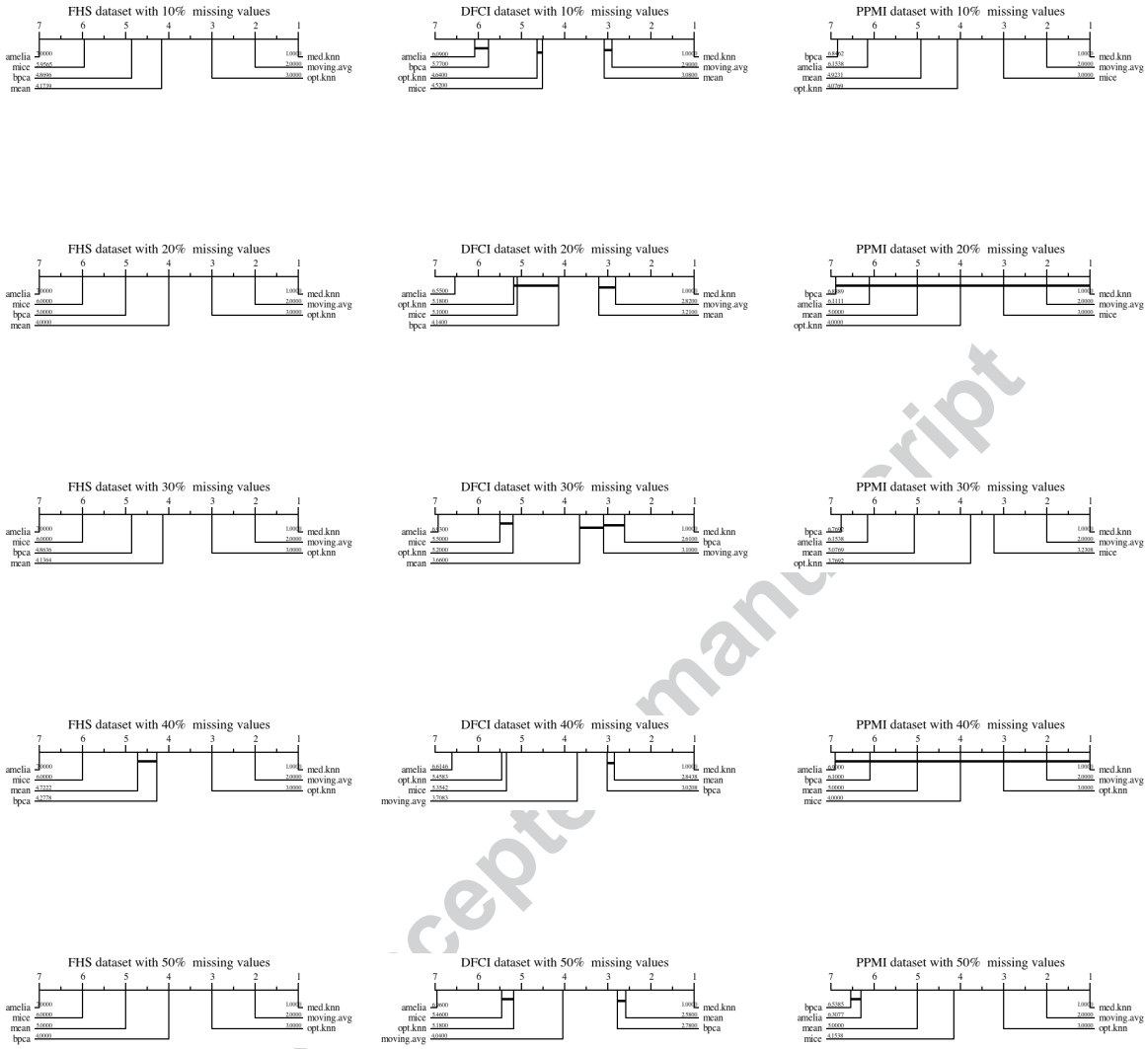

Fig. 15: Nemenyi Critical Diagrams comparing the relative ranking of all methods on the imputation tasks varying the percentage of missing data with respect to the MAE metric. Diagrams are shown for the FHS, DFCI, and PPMI datasets (right to left) and varying levels of missing data from 10% to 50% (top to bottom). In each diagram, imputation methods are plotted according to their average relative rankings on a number line from one to seven. Methods which do not have statistically significant differences in their overall rankings are joined by a horizontal line.

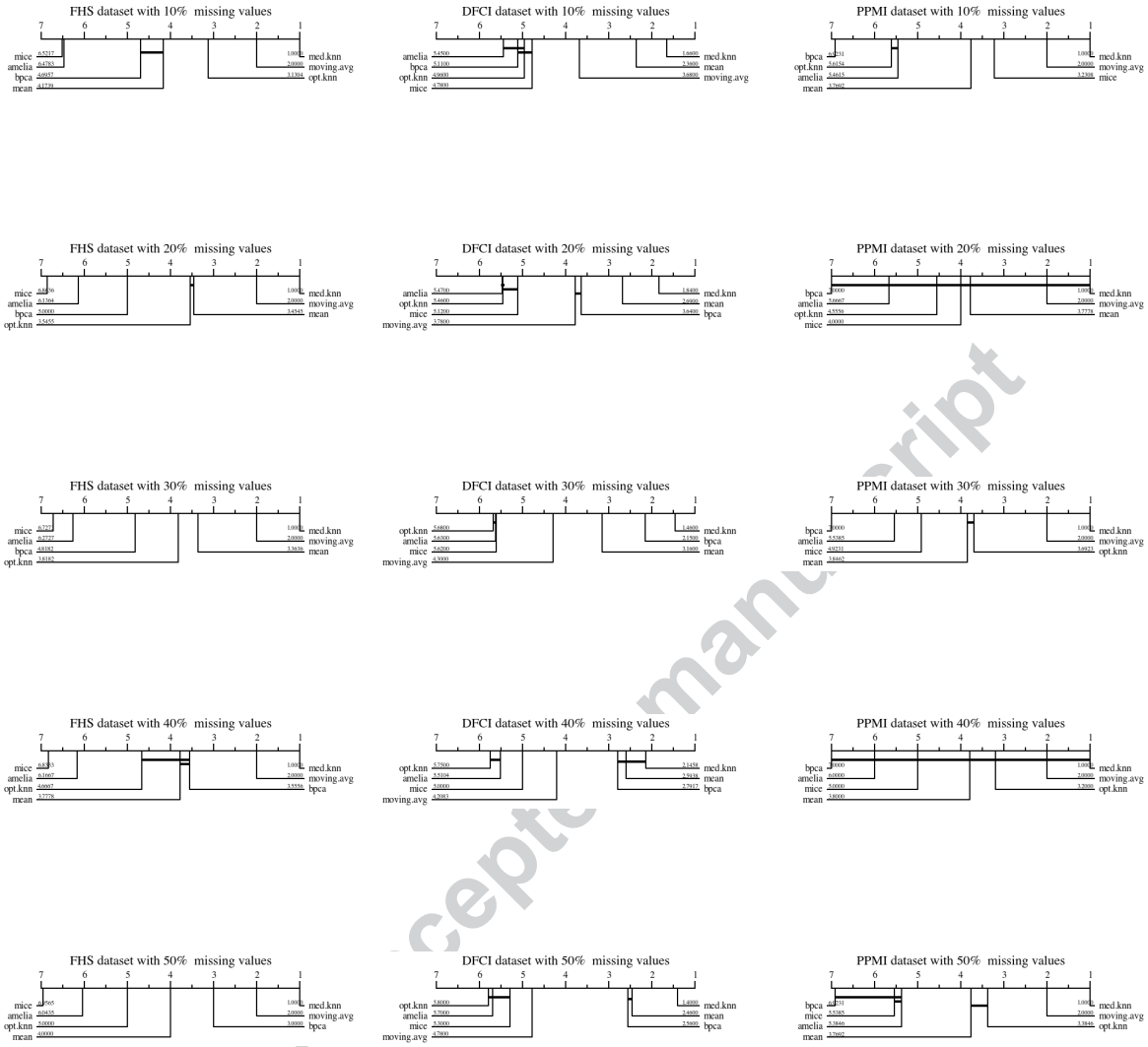

Fig. 16: Nemenyi Critical Diagrams comparing the relative ranking of all methods on the imputation tasks varying the percentage of missing data with respect to the RMSE metric. Diagrams are shown for the FHS, DFCI, and PPMI datasets (right to left) and varying levels of missing data from 10% to 50% (top to bottom). In each diagram, imputation methods are plotted according to their average relative rankings on a number line from one to seven. Methods which do not have statistically significant differences in their overall rankings are joined by a horizontal line.

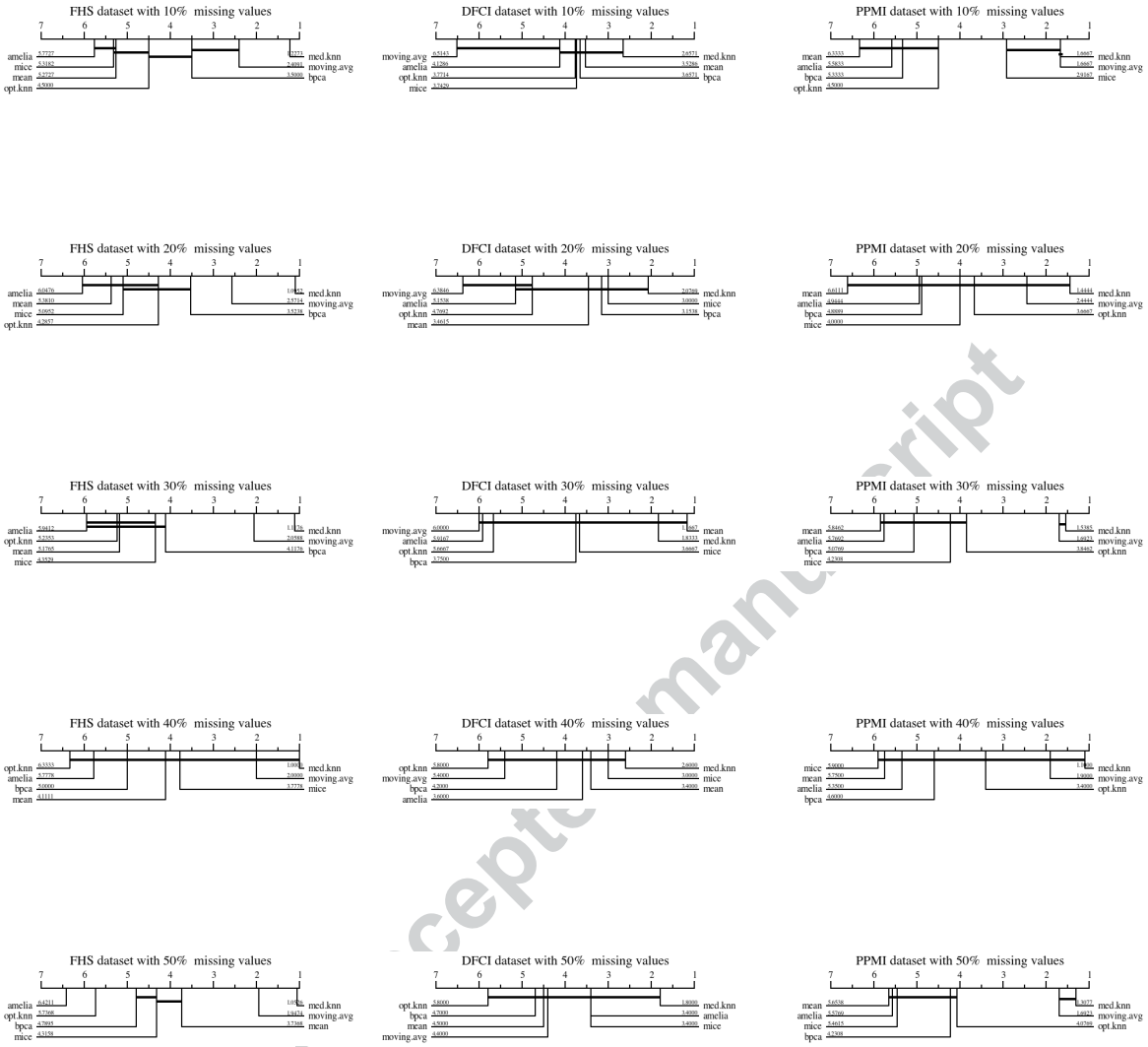

Fig. 17: Nemenyi Critical Diagrams comparing the relative ranking of all methods on the downstream predictive tasks varying the percentage of missing data. For the FHS and DFCI datasets, rankings are based upon the out-of-sample AUC metric, and for the PPMI dataset, rankings are based upon the out-of-sample MAE metric. Diagrams are shown for the FHS, DFCI, and PPMI datasets (right to left) and varying levels of missing data from 10% to 50% (top to bottom). In each diagram, imputation methods are plotted according to their average relative rankings on a number line from one to seven. Methods which do not have statistically significant differences in their overall rankings are joined by a horizontal line.

7.5 Additional Statistical Significance Tests for the Percentage of Missing Data Experiments

In this section, we present the results from statistical tests comparing the performance of the `med.knn` method at different levels of missing data. In particular, we run Welch two-sided t-tests evaluating whether `med.knn` leads to higher imputation error and lower downstream model accuracy as we increase the percentage of missing data. We run statistical tests for the following pairs of missing percentages: 10% vs. 20%, 20% vs. 30%, 30% vs. 40%, and 40% vs. 50%.

The results from the t-tests comparing the imputation errors are summarized in Table 21. We observe that in all of the experiments, the imputation error significantly increases going from 40% to 50% missing values. Similarly, the imputation error significantly increases going from 30% to 40% missing values with the exception of the DFCI dataset. For the MAE metric, most of the differences are not significant for percentage shifts below 30%. On the other hand, most of the differences for the percentage shifts below 30% are significant for the RMSE metric, with the exception of two cases where the opposite trend is observed.

Dataset	**Metric**	**10% − 20%**	**20% − 30%**	**30% − 40%**	**40% − 50%**
DFCI	MAE	1.696 (0.093)	0.79 (0.432)	1.543 (0.127)	3.914 (<0.001***)
FHS	MAE	-1.612 (0.116)	7.506 (<0.001***)	5.648 (<0.001***)	10.413 (<0.001***)
PPMI	MAE	2.146 (0.023*)	0.851 (0.403)	6.217 (<0.001***)	8.155 (<0.001***)
DFCI	RMSE	4.772 (<0.001***)	6.318 (<0.001***)	7.716 (<0.001***)	9.336 (<0.001***)
FHS	RMSE	-39.232 (0.005**)	44.142 (<0.001***)	51.106 (<0.001***)	69.95 (<0.001***)
PPMI	RMSE	11.486 (<0.001***)	-6.807 (0.028*)	28.54 (<0.001***)	35.852 (<0.001***)

Table 21: Results from Welch two-sided t-tests comparing the imputation error of the `med.knn` algorithm for varying pairs of missing percentages under the MCAR missing data mechanism. Each entry shows the t-statistic for a particular comparison with the associated p-value in parentheses. Positive t-statistics indicate that the lower missing percentage is associated with the lower imputation error.

The results from the t-tests comparing the downstream model performance are summarized in Table 22. In all but one case, we observe that the downstream model performance significantly declines as the percentage of missing data increases.

Dataset	**Metric**	**10% − 20%**	**20% − 30%**	**30% − 40%**	**40% − 50%**
DFCI	AUC	32.366 (<0.001***)	50.443 (<0.001***)	81.234 (<0.001***)	120.435 (<0.001***)
FHS	AUC	160.727 (<0.001***)	160.093 (<0.001***)	135.249 (<0.001***)	163.204 (<0.001***)
PPMI	MAE	2.154 (0.051)	10.672 (<0.001***)	18.744 (<0.001***)	16.165 (<0.001***)

Table 22: Results from Welch two-sided t-tests comparing the downstream model accuracy of the `med.knn` algorithm for varying pairs of missing percentages under the MCAR missing data mechanism. Each entry shows the t-statistic for a particular comparison with the associated p-value in parentheses. Positive t-statistics indicate that the lower missing percentage is associated with higher downstream model accuracy.