Generating Digital Twins with Multiple Sclerosis Using Probabilistic Neural Networks

Jonathan R. Walsh,* Aaron M. Smith, Yannick Pouliot,
David Li-Bland, Anton Loukianov, and Charles K. Fisher
Unlearn.AI, Inc., San Francisco, CA

for the Multiple Sclerosis Outcome Assessments Consortium†

Abstract

Multiple Sclerosis (MS) is a neurodegenerative disorder characterized by a complex set of clinical assessments. We use an unsupervised machine learning model called a Conditional Restricted Boltzmann Machine (CRBM) to learn the relationships between covariates commonly used to characterize subjects and their disease progression in MS clinical trials. A CRBM is capable of generating digital twins, which are simulated subjects having the same baseline data as actual subjects. Digital twins allow for subject-level statistical analyses of disease progression. The CRBM is trained using data from 2395 subjects enrolled in the placebo arms of clinical trials across the three primary subtypes of MS. We discuss how CRBMs are trained and show that digital twins generated by the model are statistically indistinguishable from their actual subject counterparts along a number of measures.

* jon@unlearn.ai

† Data used in the preparation of this article were obtained from the Multiple Sclerosis Outcome Assessments Consortium (MSOAC). As such, the investigators within MSOAC contributed to the design and implementation of the MSOAC Placebo database and/or provided placebo data, but did not participate in the analysis of the data or the writing of this report.

I. INTRODUCTION

Statistical models that characterize and predict disease progression have the potential to become important tools for research and management of complex, chronic indications like Multiple Sclerosis (MS). In the context of clinical research, for example, statistical models of disease progression could be used to simulate different study designs [1], to identify patients who are likely to progress more rapidly than typical patients [2], or as external comparators in early stage clinical trials [3]. Despite the availability of Disease Modifying Therapeutics (DMTs) for MS, many clinical trials still compare experimental treatments to placebo controls [4–7], which can deter patients from enrolling in those trials. At the same time, the rich history of clinical trials in MS provides a high quality source of data about disease progression under placebo or non-DMT control conditions, creating an opportunity for statistical models to further the clinical understanding of MS and better inform the design and analysis of trials [8].

Ideally, a statistical model of disease progression would describe all of the clinical characteristics that are relevant for a patient with MS, how these characteristics change through time, their inter-relationships, and their variation. In more technical terms, it would describe a joint probability distribution of the relevant clinical characteristics over time. Throughout, we refer to a sample drawn from this model distribution as a *digital subject*. By definition, therefore, a digital subject is a computationally generated clinical trajectory with the same statistical properties as clinical trajectories from actual patients. The primary advantages of digital subjects, in comparison to data from actual patients, are that they present no risk of revealing private health information and make it possible to quickly simulate patient cohorts of any size and characteristics.

Although the ability to create digital subjects enables one to simulate cohorts of patients, there are many applications for which one would like to make predictions about a particular patient. A statistical model that is able to generate digital subjects can also be used for individual patients by generating digital subjects that have the same clinical characteristics as the patient of interest at a given date (such as the start of a clinical study). We refer to a digital subject that has been generated in order to match a particular patient as a *digital twin* of that patient, in analogy with the usage of the term "digital twin" in engineering applications [9]. For example, if the statistical model represents disease progression on a

placebo, then a digital twin provides a potential outcome – i.e., what would likely happen to this patient if she/he were given a placebo in a clinical trial? In more technical terms, a digital twin is a sample from a joint probability distribution of the relevant clinical characteristics at future times conditioned on the values of those characteristics at a previous time. Note that it is possible to generate multiple digital twins for any patient.

Evaluating the quality of a model that generates digital subjects or digital twins is more complex than the evaluation of traditional machine learning models. For example, it is necessary, but not sufficient, that the means, standard deviations, correlations, and autocorrelations of all covariates computed from cohorts of digital subjects agree with those computed from cohorts of actual subjects. Here, we introduce a concept called *statistical indistinguishability*. Digital subjects, or digital twins, are statistically indistinguishable from actual subjects if a statistical procedure designed to classify a given clinical record as a digital subject or an actual subject performs consistently with random guessing.

Creating a statistical model that can generate digital subjects and digital twins that are statistically indistinguishable from actual patients is no small feat, but recent advances in generative neural networks make it possible to train generative models for complex probability distributions [10–12], including time-dependent clinical data [2]. Here, we train a probabilistic generative neural network to create digital subjects with MS using a database of placebo arms aggregated across 8 historical clinical trials [8] covering many clinically relevant covariates including demographic information, relapses, the Expanded Disability Status Scale (EDSS) [13], and functional tests. To illustrate the concept of a digital subject for MS, Figure 1 shows two digital subjects generated by the model with the same starting characteristics (i.e., the two subjects are digital twins).

As a complex neurodegenerative disorder with a course that is "highly varied and unpredictable" [14], MS is both a good test-case for the application of machine learning methods as well as a disease area that would benefit from improved characterization and prediction of disease progression. Previous models for disease progression in MS include survival models [15–17], Markov models [18–20], and other methods [21–25]. Most of these studies aim to predict the progression of either the total EDSS score or relapses as they are common ways to characterize disease severity. To our knowledge, our approach is the first that models comprehensive subject-level clinical trajectories that go beyond the total EDSS score. In this case we model 20 covariates relevant to capturing MS progression (see Appendix A).

Multiple Sclerosis Digital Twins Example

Age (years): 44
Sex: Female
Race: White
Region: Europe

MS type: RRMS
Relapses in past year: 1
Relapses in past 2 years: 2

Digital Twin A

Time (months)	Baseline	3	6	9	12	15	18	21	24	27	30	33	36
Had Relapse	N	N	N	N	N	N	N	N	N	N	N	N	N
KFSS Pyramidal	2	2	3	2	2	2	3	3	3	3	2	2	2
KFSS Sensory	1	2	2	1	2	2	2	0	2	1	3	1	1
KFSS Bowel Bladder	1	1	2	1	2	2	1	2	1	1	2	2	1
KFSS Visual	1	0	1	1	1	0	1	1	1	1	3	3	3
KFSS Cerebellar	2	1	1	1	1	2	2	2	2	2	1	1	1
KFSS Brain Stem	1	0	1	1	1	2	2	2	2	1	1	1	1
KFSS Mental	0	0	0	0	0	0	0	0	1	0	1	2	2
Ambulation Score	0	0	0	0	0	0	0	0	0	0	0	1	0
EDSS Total Score	2.5	2.5	3.5	2	3	3.5	4	4	4	3.5	4	5	3.5
Timed 25-Foot Walk (s)	10.1	6.0	7.2	7.2	5.4	5.4	4.5	3.4	6.5	5.3	5.7	7.5	5.7
9-Hole Peg Test (Dominant Hand, s)	30.9	28.5	38.9	28.3	26.6	28.5	16.4	18.9	17.5	18.2	24.6	23.9	17.4
9-Hole Peg Test (Non-Dominant Hand, s)	43.3	30.7	39.1	26.5	22.7	22.7	26.8	28.5	32.0	19.7	18.3	23.4	21.7
PASAT, 3-second	50	58	59	59	55	59	51	57	57	59	60	59	59

Digital Twin B

Time (months)	Baseline	3	6	9	12	15	18	21	24	27	30	33	36
Had Relapse	N	**Y**	N	N	**Y**	N	N	N	N	N	N	N	N
KFSS Pyramidal	2	2	2	2	3	3	2	3	4	4	4	4	4
KFSS Sensory	1	1	2	1	2	1	2	1	0	1	0	0	0
KFSS Bowel Bladder	1	1	1	0	1	0	1	0	0	0	0	0	0
KFSS Visual	1	1	1	1	2	1	0	0	0	0	0	0	0
KFSS Cerebellar	2	2	2	2	3	2	4	3	2	3	3	3	3
KFSS Brain Stem	1	2	1	2	3	2	2	2	1	1	1	0	1
KFSS Mental	0	2	2	2	1	1	0	0	0	0	0	0	0
Ambulation Score	0	0	0	0	2	3	1	2	1	3	4	3	3
EDSS Total Score	2.5	3	3	3	5.5	6	5	5.5	5	6	6.5	6	6
Timed 25-Foot Walk (s)	10.1	5.8	13.6	4.2	6.9	4.4	10.7	20.8	23.0	25.6	6.9	15.8	29.5
9-Hole Peg Test (Dominant Hand, s)	30.9	30.2	18.8	32.2	46.4	32.4	35.9	28.8	27.3	27.5	28.8	23.6	33.3
9-Hole Peg Test (Non-Dominant Hand, s)	43.3	41.9	32.8	26.2	36.9	29.7	24.7	20.3	31.4	22.3	24.9	27.0	41.4
PASAT, 3-second	50	46	32	55	48	57	50	59	47	52	37	24	40

FIG. 1. **Example digital subjects for MS.** Two digital subjects were sampled from the joint probability distribution learned by our model so that the two subjects share the same baseline characteristics (i.e., they are digital twins). Because the model is probabilisitic, the two digital twins have different outcomes even though they share the same starting characteristics.

II. METHODS

A. Generating Digital Subjects with a Conditional Restricted Boltzmann Machine

Clinical data have many properties that make statistical modeling challenging. Data can come in varied modalities such as binary, ordinal, categorical, and continuous. Furthermore, observations may be occasionally or frequently missing. Conditional Restricted Boltzmann Machines (CRBMs) are well suited to address these challenges, and in past work we have found that CRBMs are able to effectively model disease progression in Alzheimer's Disease [2].

A CRBM[1] is a class of probabilistic neural network that learns a joint probability distribution over time, and is closely related to Restricted Boltzmann Machines [26–30]. A CRBM learns the relationship between covariates at k visits expressed as a parametric probability distribution. For example, for $k = 3$ with a 3-month spacing between visits the distribution takes the form

$$p(\mathbf{x}(t+3), \mathbf{x}(t), \mathbf{x}(t-3)) = Z^{-1} \int d\mathbf{h}\, e^{-U(\mathbf{x}(t+3), \mathbf{x}(t), \mathbf{x}(t-3), \mathbf{h})}\,, \tag{1}$$

in which $\mathbf{x}(t)$ is the vector of covariates at time t (in months), $\mathbf{h}$ is a vector of hidden variables, $U(\cdot)$ is called the energy function, and Z is a normalization constant. More details are provided in Appendix B.

The CRBM allows for approximate sampling from the joint probability distribution over all covariates and across k visits. Thinking of disease progression as a Markov process, the model can be used to probabilistically generate data for a subject at a visit given data from the previous $k-1$ visits. For MS, we model data in 3-month intervals and find that modeling $k = 3$ simultaneous visits in the CRBM is sufficient to accurately generate long trajectories. In other words, we model MS clinical trajectories as a lag-2 Markov process.

The CRBM is used in an iterative fashion to generate clinical trajectories for digital subjects or digital twins. To generate a digital subject, the model is used to generate data at time zero (i.e., at baseline) by sampling from the marginal distribution $p(\mathbf{x}(t = 0))$ using Markov Chain Monte Carlo (MCMC) methods. To generate a digital twin, the covariates at time zero (baseline) are set equal to the observed covariates of a particular patient. Starting

[1] We use a version of CRBMs that have purely undirected connections, for which only a composite likelihood may be used for training, that is more appropriate for this application; see Appendix B for discussion.

from these baseline data, the CRBM is used to generate data for the 3- and 6-month visits for that subject by sampling from $p(\mathbf{x}(t=6), \mathbf{x}(t=3)|\mathbf{x}(t=0))$. Then these sampled 3- and 6-month visit data may be used to generate data for the 9-month visit by sampling from $p(\mathbf{x}(t=9)|\mathbf{x}(t=6), \mathbf{x}(t=3))$. Similarly the sampled 6- and 9-month visit data may be used to generate data for the 12-month visit by sampling from $p(\mathbf{x}(t=12)|\mathbf{x}(t=9), \mathbf{x}(t=6))$. This continues as needed to create clinical trajectories of the desired length.

Under this framework, we can now define digital subjects and digital twins more formally. A digital subject is a clinical trajectory of length τ sampled from the joint distribution across all τ visits $\{\mathbf{x}(t)\}_{t=0}^{t=\tau} \sim p(\mathbf{x}(t=\tau), \ldots, \mathbf{x}(t=0))$. A digital twin is a clinical trajectory of length τ in which the visits after $t=0$ are sampled from the conditional distribution $\{\mathbf{x}(t)\}_{t=3}^{t=\tau} \sim p(\mathbf{x}(t=\tau), \ldots, \mathbf{x}(t=3)|\mathbf{x}(t=0))$.

One important concept is that digital subjects and digital twins are stochastic. Therefore, one can create many digital twins for a given patient. Each of these digital twins will have the same covariates at baseline, but their trajectories will differ after $t=0$. Taken together, these digital twins map out the distribution of possible clinical trajectories for that patient. Note that subjects in clinical trials do not receive different treatments at the baseline visit, so that digital twins may be created for subjects in any arm. If the CRBM models the progression of the disease under the control condition, then the model may be used to generate counterfactuals that can be used to estimate treatment effects. In this work, our training and validation data were collected entirely from placebo control arms of previously completed clinical trials so that digital subjects generated from the model are representative of placebo controls.

A schematic overview of our process for building the CRBM and generating digital twins is given in Figure 2.

B. Data

One legacy of the long history of clinical development for MS is a rich set of historical clinical trials with placebo arms. The Multiple Sclerosis Outcome Assessments Consortium (MSOAC) [8] has compiled data from 16 clinical trials that span three subtypes of MS: relapsing remitting (RRMS), secondary progressive (SPMS), and primary progressive (PPMS). A database of placebo arms from 8 of these trials comprising 2465 subjects has been made

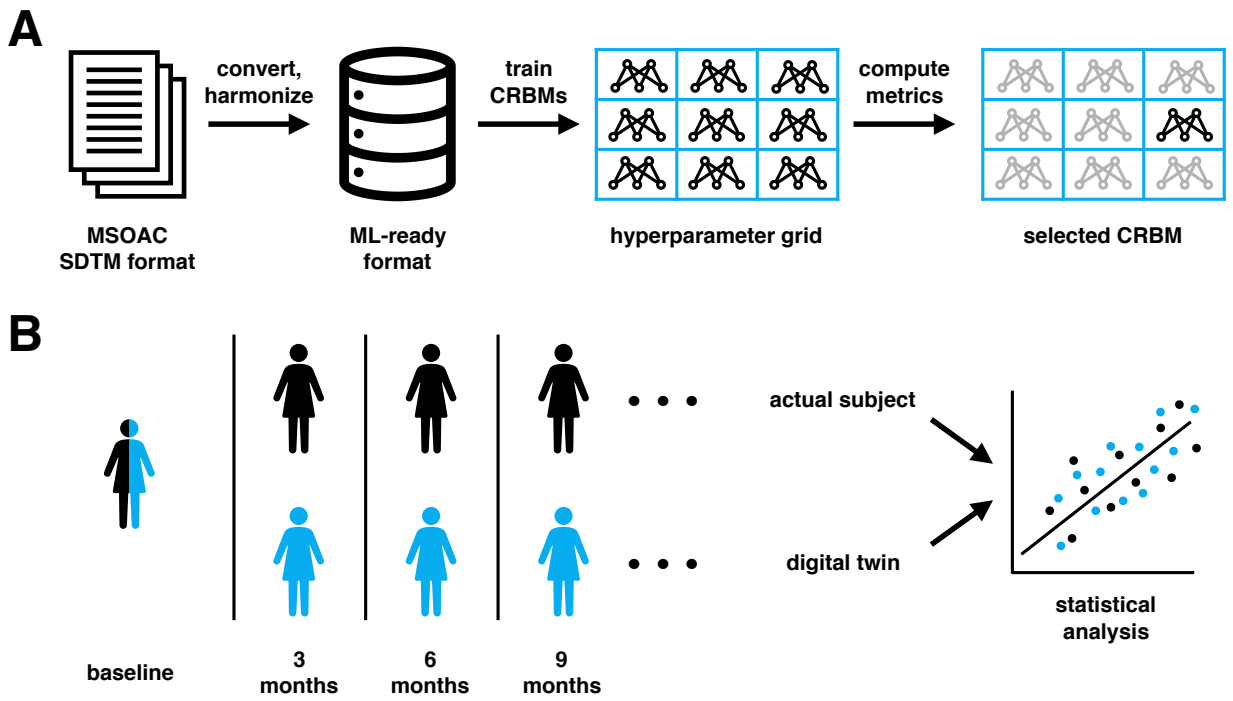


FIG. 2. **Overview.** A graphical summary of the process used to build the CRBM studied in this paper (A), and how digital twins are analyzed (B). To build the model, MSOAC data in SDTM format are converted to a numerical form suitable for machine learning. A set of models are trained that sweep over a grid of hyperparameters, variations in the way the model is trained. The final model is selected by computing metrics on validation data and choosing the best ranked model. An important feature of the CRBM is that it can create digital subjects and digital twins. Digital subjects are synthetic clinical records generated from the model. Digital twins are digital subjects whose baseline data is that of a given subject, allowing for subject-level predictions of outcomes. The CRBM is probabilistic, meaning many digital twins may be created for an actual subject and used to build distributions of predicted outcomes. In this work we use digital twins to analyze the quality of the CRBM as a disease progression model.

available to qualified researchers. This database contains measurements spanning a number of domains including background information, questionnaires and functional assessments, medical history, and concomitant medications. No imaging or biomarker data is available.

MSOAC data are encoded according to the Study Data Tabulation Model (SDTM) format, a highly structured format commonly used to submit the results of clinical trials to regulatory authorities [31, 32]. Starting from the MSOAC placebo dataset, we standardized and converted SDTM-formatted measurements into a tidy format [33] more suitable for

Name	Category	Type	Longitudinal	Statistics	Missing [%]
Baseline Age	Background	Continuous	No	42 (10) [years]	1
Sex	Background	Binary	No	67% female	0
Race	Background	Binary	No	91% white	33
Region	Background	Categorical	No	64% Europe	58
MS type	Clinical	Categorical	No	65% RRMS	0
# of relapses, 1 year before baseline	Clinical	Ordinal	No	1.4 (0.8)	26
# of relapses, 2 years before baseline	Clinical	Ordinal	No	2.2 (1.3)	68
Relapse events	Clinical	Binary	Yes	95% no-relapse	0
KFSS bowel and bladder system	Clinical	Ordinal	Yes	0.83 (0.98)	13
KFSS brain stem system	Clinical	Ordinal	Yes	0.64 (0.87)	13
KFSS cerebellar system	Clinical	Ordinal	Yes	1.27 (1.17)	14
KFSS mental system	Clinical	Ordinal	Yes	0.58 (0.84)	13
KFSS pyramidal system	Clinical	Ordinal	Yes	1.86 (1.16)	13
KFSS sensory system	Clinical	Ordinal	Yes	1.19 (1.11)	13
KFSS visual system	Clinical	Ordinal	Yes	0.76 (1.00)	13
Ambulation EDSS component	Clinical	Ordinal	Yes	0.72 (1.36)	1
Timed 25-Foot Walk	Functional	Continuous	Yes	9.5 (14.6) [s]	1
Nine-Hole Peg, dominant hand	Functional	Continuous	Yes	24.8 (15.2) [s]	1
Nine-hole Peg, non-dominant hand	Functional	Continuous	Yes	27.7 (22.7) [s]	1
Paced Auditory Serial Addition Test (3s)	Functional	Continuous	Yes	47.4 (11.8)	1

TABLE I. Covariates used in the model are grouped into three categories: background, clinical, and functional. Background covariates are demographics that characterize the subject population. Functional covariates assess various aspects of MS-induced disability such as ambulation, motor skills, and cognitive function. Clinical covariates describe a subject's clinical state such as the type of MS, degree of disability (through the Kurtzke Functional Systems Score, or KFSS, components), and relapse information. The Statistics column gives the mean and standard deviation of each covariate (or the dominant category for binary or categorical variables) at baseline in the combined training, validation, and test sets. The Missing column gives the percentage of missing data for each covariate at baseline.

statistical analysis and machine learning as described in Appendix A.

Twenty different covariates, described in Table I, were selected for inclusion in the model due to their clinical relevance and presence (non-missingness) in the dataset. They principally span background, clinical, and functional domains. Other clinically useful variables,

such as questionnaires with data recorded for only a small fraction of subjects, were excluded. The components of EDSS and relapse events, the primary non-imaging endpoints in clinical trials, were included in the covariates modeled. Additionally, we added a simple longitudinal covariate that is modeled as binary, which is 1 at baseline and 0 otherwise. This allows the model to treat baseline as a special time.

Each selected covariate was classified as either longitudinal or static and as one of binary, ordinal, categorical, or continuous; these classifications affect how the covariate is modeled by the CRBM. The resulting dataset contains clinical trajectories for all subjects in the database in 3-month intervals for up to 48 months and an average duration of approximately 24 months.

Most subjects have a 3-month visit interval, but a 6-month interval is also present for approximately 800 subjects. Visit days were standardized to have a baseline visit day of 0, and longitudinal measurements were grouped into windows centered on each 90-day (3-month) visit and averaged. Any visit windows without a measurement for a particular covariate were recorded as missing.

We model each of the constituent components of the EDSS score, a combination of the 7-component Kurtzke Functional Systems Score (KFSS) assessment of function in a variety of body systems [13, 34, 35] and the ambulatory function of the subject. We discuss the advantages and methods around different approaches for representing these data more in Appendix A.

Before training, we removed 70 subjects that had essentially no data beyond baseline, resulting in a dataset with 2395 subjects. This dataset was divided into mutually exclusive training (50% of the data, or 1198 subjects), validation (20% of the data, or 479 subjects), and test (30% of the data, or 718 subjects) datasets. Due to the computational cost of associated with our training procedure, we had to rely on a single train-validation-test split, as opposed to a method like k-fold cross validation.

C. Training

A well-trained CRBM is able to generate digital subjects with the same statistical properties as actual subjects in the dataset. To accomplish this goal, we trained the CRBM via stochastic gradient ascent to maximize a convex combination of a composite likelihood and

an adversarial objective as previously described [2, 10]. The relative weight given to the likelihood and adversarial components of the objective function was treated as a hyperparameter.

Hyperparameters refer to various parameters of the model or training process that must be specified, but cannot be learned by applying gradient ascent to the objective function as with the main model parameters. These hyperparameters included the weighting of the likelihood and adversarial terms of the objective function, number of epochs, batch size, initial learning rate, number of hidden units, magnitude of parameter regularization, and sampling parameters (i.e., the number of Monte Carlo steps and the magnitude of temperature-driven sampling [10]). We performed a grid search over each of these dimensions in parallel, training a total of 1296 CRBMs with various hyperparameter choices to maximize an objective function on the training dataset.

After training the CRBMs in the grid search, we evaluated them using a variety of metrics that assess a model's statistical performance as well as clinically important outcomes. All metrics were computed on the validation dataset. The statistical metrics are coefficients of determination (R^2 values) between the actual subjects from the test set and their digital twins the for equal-time and lagged autocorrelations. Three clinical metrics were used, one measuring the ability to predict individual relapse events, another measuring the agreement between average EDSS progression values for the data and model, and the third measuring the agreement between chronic disease worsening (CDW) fractions for the data and model.

We adopted a minimax approach to choose a best performing model across the grid search. That is, we aimed to choose a CRBM that performed well across all metrics, even if it was not the best performing model on any single metric. To accomplish this, we ranked all N (N=1296) models from best (rank 1) to worst (rank N) for each metric and determined the worst rank across metrics for each model. We used this worst rank to choose the top 25% of models. This selected models that performed well on all metrics. To choose a best model from this set, we repeated the process on the clinical metrics alone, ranking models on the clinical metrics and determining the worst rank of each model. We selected the model with the lowest (best) worst rank and designated this as the "optimal" model. Narrowing the focus to the clinical metrics emphasized their importance in model selection.

The hyperparameters of this optimal model were then used to train a new, "final" model on the combined training and validation datasets. All results that follow in the main text

focus on the performance of this final model computed using the test dataset. More details on the hyperparameters, training procedures, and selection process for the optimal model are described in Appendix B and Appendix C.

III. RESULTS

A. Model Performance Across the Hyperparameter Sweep

The hyperparameter sweep trained 1296 models over a grid of hyperparameters on the training dataset. 83 of these models failed to finish training due to poor performance (e.g., due to too high of a learning rate), and were excluded from further evaluation because they were ranked lowest in model selection. We computed the model selection metrics using the validation dataset on the remaining 1213 models and determined the hyperparameters of the optimal model. These hyperparameters were used to train a new model on the combination of the training and validation datasets, which we refer to as "the final model". For comparison, the test metrics were computed using both the optimal model determined from the validation set and the final model trained on combining training and validation sets.

The distributions of the validation metrics used for the hyperparameter sweep are shown in Figure 3, along with metric values for the optimal and final models. As expected, the optimal model performed well on all metrics without performing poorly on any individual metric. Furthermore, the performance of the optimal model was consistent across the validation and test datasets, with the largest CDW fraction showing the largest difference. The performance of the optimal and final models on the test dataset were also quite similar, aside from the relapse prediction metric. This suggests that overall model performance is fairly stable with respect to changes to the dataset used for model training or evaluation.

B. Assessing Statistical Indistinguishability of Digital Twins

We say that digital twins and actual subjects are statistically indistinguishable if statistical analysis methods cannot differentiate the two groups better than random chance. Due to the complexity of clinical data, there are a number of ways to assess statistical indistinguishability. We focus on three different methods to quantitatively assess the degree of

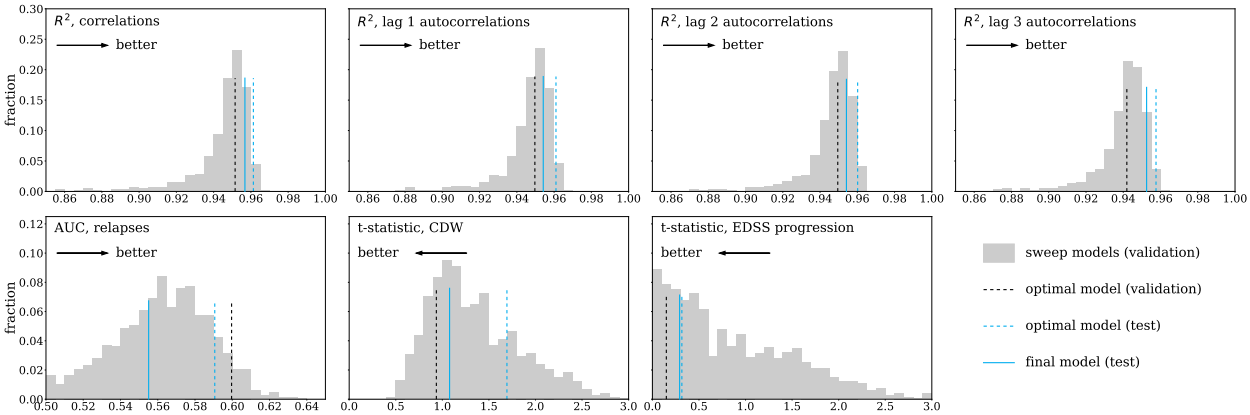


FIG. 3. **Metrics used to select hyperparameters for the best performing model.** For each metric used, the distribution of values across the hyperparameter sweep determined using the validation dataset is shown (gray histogram). The best performing model that was selected from this set is shown as a dashed black line at its value for each metric. The dashed blue line shows the value of the metric for this best performing model, computed on the test dataset. The solid blue line shows the value of the metric for the final model, computed on the test dataset. For each metric, an arrow indicates whether smaller or larger values are better.

indistinguishability that span from subject- and covariate-level statistics to population-level statistics. The first compares the means, standard deviations, correlations, and autocorrelations of all covariates computed from the digital twins to those computed from the actual subjects, quantifying agreement using linear regression and coefficients of determination where appropriate. The second aims to quantify the agreement between the observed clinical trajectory of a particular patient and the distribution of potential clinical trajectories defined by his/her digital twins. The third trains logistic regression models to distinguish between actual subjects and their digital twins, quantifying performance with the area under the receiver operating characteristic curve (AUC).

To evaluate the statistical indistinguishability of digital twins, we created 1000 digital twins for each of the 718 actual subjects in the test dataset. The statistical properties of these twins were compared to the actual subjects in a number of ways to quantify the goodness-of-fit of the CRBM.

First, in Figure 4 we compare the means, standard deviations, correlations, and autocorrelations between covariates computed from the actual subjects and a cohort consisting of a single digital twin for each subject. The duration of the clinical trajectory of each digital

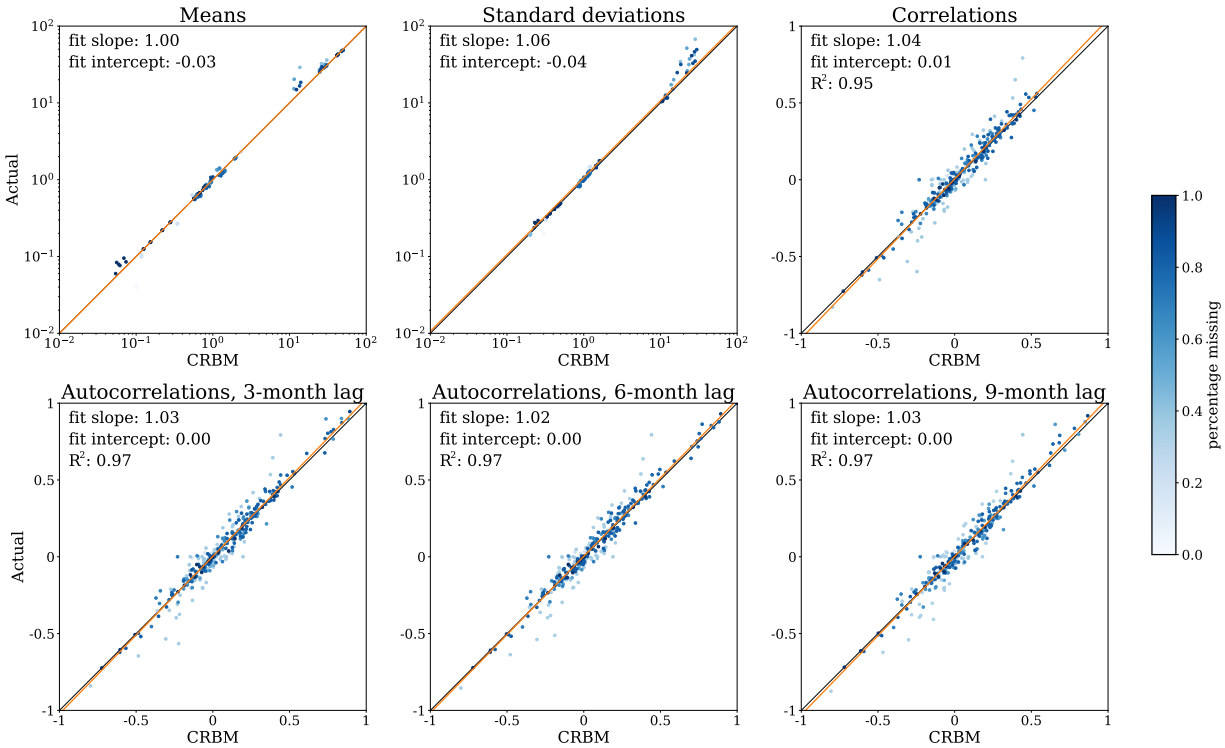


FIG. 4. **The model captures the leading statistical moments of the data.** Moments for the actual subject data are plotted against moments for digital twins for means, standard deviations, equal-time correlations, and lagged autocorrelations. Each digital twin contributes data for the same study duration as its actual subject counterpart up to 36 months. The means and standard deviations are compared at each visit, with a logarithmic scale used to accommodate differing scales of the covariates. For the equal-time and lagged autocorrelations the correlation coefficient is computed for each pair of covariates across all visits. In all cases the slope and intercept fit coefficients shown come from regressions weighted by the fraction of data present for each covariate. Points on the plot are darker if more data is present (see the color bar). These regressions estimate the relationship between actual subjects and digital twins. Theil-Sen regression is used for the means and standard deviations, which is outlier robust and appropriate for the widely varying scales present [36, 37]. For the equal-time and lagged autocorrelations, an ordinary least squares regression is used, which allows the R^2 values shown to be computed. The best fit line is shown for each comparison.

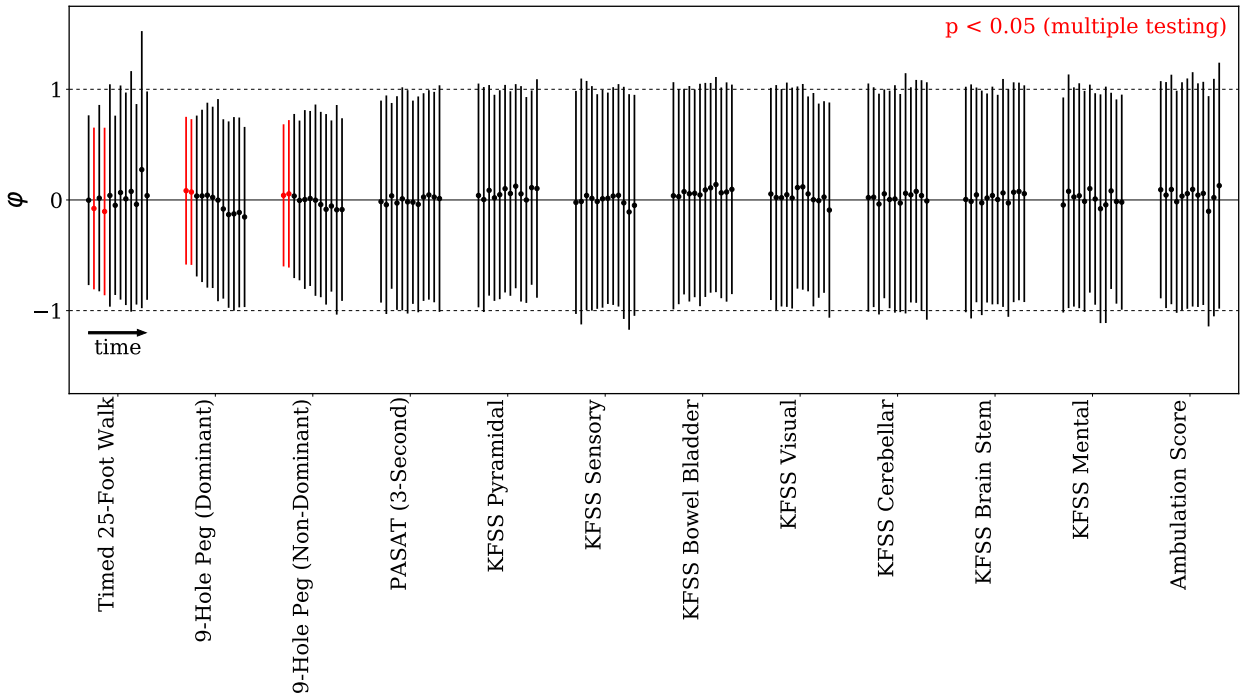
twin was the same as the matched patient. We ignored the baseline time point when computing the statistics because the actual subject and his/her digital twin have the same values of all covariates at the initial time point, by definition. All of the statistics computed from

the cohort of digital twins agree with those computed from the cohort of actual subjects, with R^2 values for all comparisons greater than 0.95 and best fit coefficients close to their ideal values (i.e., 0 for the intercept, 1 for the slope).

Figure 4 compares a cohort of actual subjects to a cohort of matched digital twins based on population-level statistics. It is also important to quantify performance using statistics that are sensitive to subject-level agreement between an actual subject and his/her digital twins as in Figure 5. We generated 1000 digital twins for each actual subject. For each subject, each covariate, and each visit, the samples from the digital twins form a distribution to which the observed data for the actual subject may be compared. We used the distribution defined by the digital twins to compute a tail area probability (i.e., p-value) for each observation, then used the inverse cumulative distribution function of the standard normal distribution to convert the p-value to a statistic, $\varphi = \Phi^{-1}(p)$, that has mean 0 and standard deviation 1 if the digital twins and subject data are drawn from identical distributions. This statistic controls for highly non-normal distributions of covariates better than a traditional z-score. We applied the Kolmogorov-Smirnov test to assess the if the distribution of these statistics is not $\mathcal{N}(0, 1)$ and mark any cases with statistically significant disagreement at $\alpha = 0.05$ after a Bonferroni correction.

Only early visits for the timed 25-foot walk and 9-hole peg test show statistically significant differences between actual subjects their digital twins, with the difference primarily due to the standard deviation of the φ distribution being too small. This indicates that the standard deviations of the distributions defined for these covariates by the digital twins are too large. These two functional tests are scored as time to completion of a task. Subjects can have large single-visit outliers in this time due to short-term ambulatory or motor difficulties, which is particularly challenging to learn. Although the model generally agrees with the data well, these results suggest that the CRBM generates large times for these tests too frequently.

Finally, in Figure 6, we demonstrate that a logistic regression model cannot distinguish between an actual subject and one of his/her digital twins better than random chance, which is a direct evaluation of statistical indistinguishability. For each time point (except baseline, when the actual subjects and their twins are the same by definition), we trained a logistic regression model to distinguish between each subject and his/her digital twins. In addition, we trained logistic regression models to distinguish between each subject and



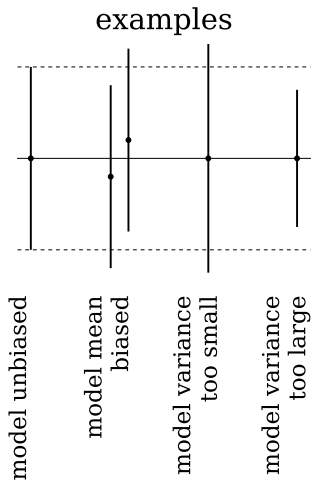


FIG. 5. **The model accurately simulates individual subject trajectories.** The mean and standard deviation of a statistic measuring subject-level agreement of the data with the CRBM, for each longitudinal covariate. Each visit beyond baseline is displayed using a point for the mean and an error bar for the standard deviation. The visits are ordered for each covariate, with data shown up to 36 months. For each subject, a p-value for each covariate at each visit is computed by comparing the data value to the distribution of values predicted by the CRBM under repeated simulations of digital twins for that subject. The inverse normal CDF is applied to this p-value to define a statistic, $\varphi \equiv \Phi^{-1}(p)$, and the mean and standard deviation of the distribution of φ over subjects are computed and plotted as a point and error bar. As the diagram on the right explains, if the CRBM is in statistical agreement with the data (unbiased) then the mean should be 0 and the standard deviation should be 1. Different types of bias are shown, and we label statistically significant cases ($p < 0.05$, with a confidence level adjusted to 0.05/144 after applying a Bonferroni correction that factors in the 144 comparisons shown in the figure) in red.

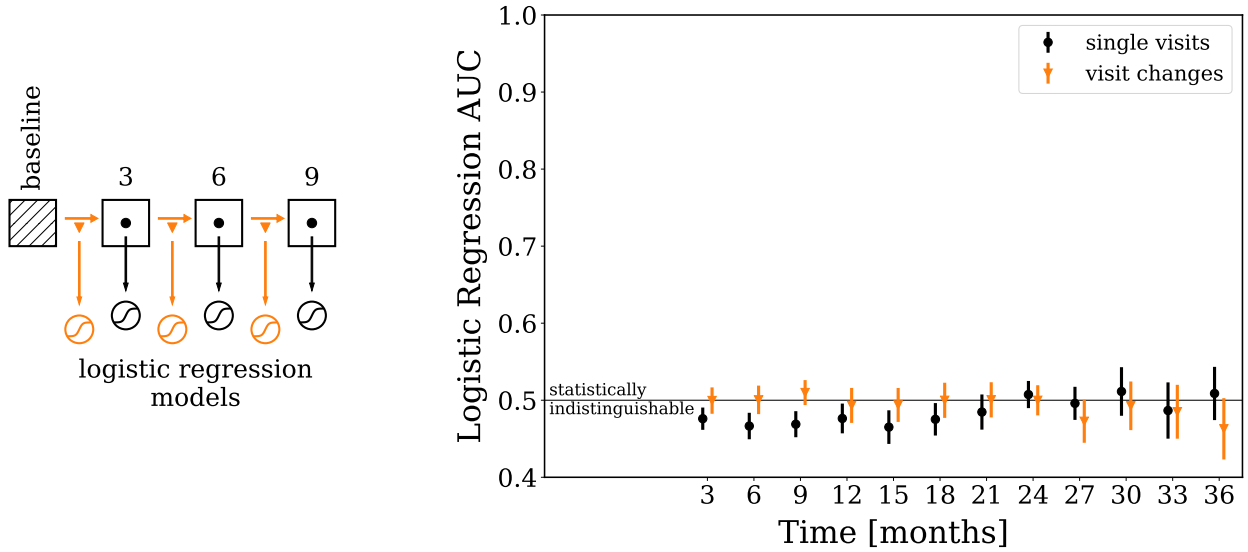


FIG. 6. **Digital twins are challenging to distinguish from actual subjects.** The area under the receiver operating characteristic curve (AUC) of logistic regression models trained to distinguish between actual data and their digital twins are shown. An AUC of 0.5 indicates the logistic regression model cannot differentiate between actual data and their digital twins better than random chance, while an AUC of 1 indicates the model can perfectly differentiate between the two groups. As the diagram at left shows, at each visit beyond baseline we evaluate the ability to distinguish digital twins from data either on data from a single visit (black circles) or the change in data from one visit to the next (orange triangles). In each case the point shown is the mean AUC is taken over 100 different simulations with different digital twins for each simulation; the error bars on the points show the standard deviation over simulations. The AUC for each simulation is estimated using 5-fold cross validation. Missing data from actual subjects is mean imputed; the corresponding values for the digital twins are assigned the same mean values to avoid biasing the logistic regression models.

his/her digital twins using differences between time points. Over 36 months, nearly every logistic regression model's performance is consistent with random guessing, meaning they are unable to distinguish between actual subjects and their digital twins better than random chance.

IV. DISCUSSION

This work introduced three terms: *digital subjects*, *digital twins*, and the concept that digital subjects may be *statistically indistinguishable* from actual subjects. Using a dataset of subjects enrolled in the placebo arms of MS clinical trials, we trained a Conditional Restricted Boltzmann Machine to generate digital subjects. This dataset included demographic information and disease history, functional assessments, and components of the EDSS score. The model learned the relationship between covariates across multiple visits, treating disease progression as a Markov process; this means that given data for a subject at some number of prior visits, the model is capable of generating clinical trajectories representing the distribution of potential outcomes at future visits for that subject. Using a variety of statistical tests, we demonstrated digital subjects generated by the CRBM are statistically indistinguishable from actual subjects enrolled in the placebo arms of clinical trials for MS.

Modeling MS disease progression is challenging due to the complexity of endpoint measures and the varied course of the disease. Relapses are difficult to predict, and the EDSS score has multiple domains and depends on many components. It is likely that significant improvements could be made to the model by incorporating additional data on imaging-related endpoints. Incorporating these additional covariates would enable generation of digital subjects with most of the covariates that are typically measured in MS clinical trials. In addition, incorporating data from subjects treated with interferons or other commonly-used DMTs would make it possible to generate cohorts of digital subjects representing broader control populations.

The ability to generate digital twins that are statistically indistinguishable from actual subjects enrolled in control arms of clinical trials is a powerful tool to understand disease progression and model clinical trials. Each digital twin represents a potential outcome of their matched patient; that is, what would likely happen to this patient if she/he were to receive a placebo in the context of a clinical trial? Models of potential outcomes are important tools for estimating treatment effects [38–40], and ability to generate digital twins representing per-subject controls opens the door to a variety of novel clinical trial analyses aimed at assessing responses to treatments for individual patients.

Taken in-context with previous work by the authors applying CRBMs to generate digital subjects for Alzheimer's Disease [2], this work suggests that approaches based on probabilis-

tic neural networks like CRBMs will be broadly applicable to generating digital subjects across multiple diseases.

V. ACKNOWLEDGEMENTS

We would like to thank Pankaj Mehta for helpful feedback on the manuscript.

---

[1] N. Holford, S. Ma, and B. Ploeger, Clinical Pharmacology & Therapeutics **88**, 166 (2010).

[2] C. K. Fisher, A. M. Smith, and J. R. Walsh, Scientific Reports **9**, 1 (2019).

[3] G. Carrigan, S. Whipple, W. B. Capra, M. D. Taylor, J. S. Brown, M. Lu, B. Arnieri, R. Copping, and K. J. Rothman, Clinical Pharmacology & Therapeutics **107**, 369 (2020).

[4] F. D. Lublin, S. C. Reingold, and National Multiple Sclerosis Society (USA) Task Force on Placebo-Controlled Clinical Trials in MS, Annals of neurology **49**, 677 (2001).

[5] C. H. Polman, S. Reingold, F. Barkhof, P. Calabresi, M. Clanet, J. Cohen, G. Cutter, M. Freedman, L. Kappos, F. Lublin, *et al.*, Neurology **70**, 1134 (2008).

[6] B. M. Uitdehaag, F. Barkhof, P. K. Coyle, J. D. Gardner, D. R. Jeffery, and D. D. Mikol, Current medical research and opinion **27**, 1529 (2011).

[7] Y. Zhang, A. Salter, E. Wallström, G. Cutter, and O. Stüve, Therapeutic Advances in Neurological Disorders **12** (2019).

[8] N. G. LaRocca, L. D. Hudson, R. Rudick, D. Amtmann, L. Balcer, R. Benedict, R. Bermel, I. Chang, N. D. Chiaravalloti, P. Chin, J. A. Cohen, G. R. Cutter, M. D. Davis, J. DeLuca, P. Feys, G. Francis, M. D. Goldman, E. Hartley, R. Kapoor, F. Lublin, G. Lundstrom, P. M. Matthews, N. Mayo, R. Meibach, D. M. Miller, R. W. Motl, E. M. Mowry, R. Naismith, J. Neville, J. Panagoulias, M. Panzara, G. Phillips, A. Robbins, M. F. Sidovar, K. E. Smith, B. Sperling, B. M. Uitdehaag, J. Weaver, and Multiple Sclerosis Outcome Assessments Consortium (MSOAC), Multiple Sclerosis (Houndmills, Basingstoke, England) **24**, 1469 (2018).

[9] M. Grieves and J. Vickers, in *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches*, edited by F.-J. Kahlen, S. Flumerfelt, and A. Alves (Springer, 2016).

[10] C. K. Fisher, A. M. Smith, and J. R. Walsh, arXiv:1804.08682 [cs, stat] (2018).

[11] G. W. Taylor and G. E. Hinton, in *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09 (ACM, New York, NY, USA, 2009) pp. 1025–1032, event-place: Montreal, Quebec, Canada.

[12] G. W. Taylor, G. E. Hinton, and S. T. Roweis, in *Advances in Neural Information Processing Systems 19*, edited by B. Schölkopf, J. C. Platt, and T. Hoffman (MIT Press, 2007) pp. 1345–1352.

[13] J. F. Kurtzke, Neurology **33**, 1444 (1983).

[14] M. M. Goldenberg, Pharmacy and Therapeutics **37**, 175 (2012).

[15] A. Scalfari, A. Neuhaus, M. Daumer, G. C. Ebers, and P. A. Muraro, Neurology **77**, 1246 (2011).

[16] H. Tremlett, Y. Zhao, P. Rieckmann, and M. Hutchinson, Neurology **74**, 2004 (2010).

[17] S. Vukusic and C. Confavreux, Current Opinion in Neurology **20**, 269 (2007).

[18] S. A. Gauthier, M. Mandel, C. R. G. Guttmann, B. I. Glanz, S. J. Khoury, R. A. Betensky, and H. L. Weiner, Neurology **68**, 2059 (2007).

[19] J. Palace, T. Bregenzer, H. Tremlett, J. Oger, F. Zhu, F. Zhu, M. Boggild, M. Duddy, and C. Dobson, BMJ open **4**, e004073 (2014).

[20] Y. Hou, Y. Jia, and J. Hou, Scientific Reports **8** (2018).

[21] F. Pellegrini, M. Copetti, M. P. Sormani, F. Bovis, C. de Moor, T. P. Debray, and B. C. Kieseier, Multiple Sclerosis (Houndmills, Basingstoke, England) (2019).

[22] M. T. Law, A. L. Traboulsee, D. K. Li, R. L. Carruthers, M. S. Freedman, S. H. Kolind, and R. Tam, Multiple Sclerosis Journal - Experimental, Translational and Clinical **5** (2019).

[23] M. Lawton, K. Tilling, N. Robertson, H. Tremlett, F. Zhu, K. Harding, J. Oger, and Y. Ben-Shlomo, Journal of Clinical Epidemiology **68**, 1355 (2015).

[24] A. Achiron, Y. Barak, and Z. Rotstein, Multiple Sclerosis (Houndmills, Basingstoke, England) **9**, 486 (2003).

[25] A. Achiron, Journal of Neurology **251 Suppl 5**, v65 (2004).

[26] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, Cognitive science **9**, 147 (1985).

[27] G. E. Hinton and R. R. Salakhutdinov, Science **313**, 504 (2006).

[28] R. Salakhutdinov and G. Hinton, in *Artificial Intelligence and Statistics* (2009) pp. 448–455.

[29] G. Hinton, Momentum **9**, 926 (2010).

[30] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, in *Advances in neural information processing systems* (2014) pp. 2672–2680.

[31] W. R. Kubick, S. Ruberg, and E. Helton, Drug Information Journal (2016).

[32] L. D. Hudson, R. D. Kush, E. Navarro Almario, N. Seigneuret, T. Jackson, B. Jauregui, D. Jordan, R. Fitzmartin, F. L. Zhou, J. K. Malone, J. Galvez, and L. B. Becnel, Clinical and Translational Science **11**, 342 (2018).

[33] H. Wickham *et al.*, Journal of Statistical Software **59**, 1 (2014).

[34] J. F. Kurtzke, Neurology **11**, 686 (1961).

[35] J. F. Kurtzke, Neurology **15**, 654 (1965).

[36] H. Theil, Nederl. Akad. Wetensch., Proc. **53**, 386 (1950).

[37] P. K. Sen, Journal of the American Statistical Association **63**, 1379 (1968).

[38] D. B. Rubin, Scandinavian Journal of Statistics **31**, 161 (2004).

[39] G. W. Imbens and D. B. Rubin, in *Microeconometrics* (Springer, 2010) pp. 229–241.

[40] S. R. Künzel, J. S. Sekhon, P. J. Bickel, and B. Yu, Proceedings of the national academy of sciences **116**, 4156 (2019).

[41] V. Mnih, H. Larochelle, and G. E. Hinton, arXiv preprint arXiv:1202.3748 (2012).

[42] J. Tubiana and R. Monasson, Physical Review Letters **118**, 138301 (2017).

[43] C. Varin, N. Reid, and D. Firth, Statistica Sinica , 5 (2011).

[44] D. P. Kingma and J. Ba, arXiv:1412.6980 [cs] (2017), arXiv: 1412.6980.

[45] C. Gouriéroux and J. Jasiak, Journal of Forecasting **25**, 129.

[46] G. Desjardins, A. Courville, Y. Bengio, P. Vincent, and O. Delalleau, in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (MIT Press Cambridge, MA, 2010) pp. 145–152.

[47] G. Desjardins, A. Courville, Y. Bengio, P. Vincent, and O. Delalleau, in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (2010) pp. 145–152.

Appendix A: Data Processing and Datasets

Data from the MSOAC database is stored in Study Data Tabulation Model (SDTM) format [31]. The transformation of data from SDTM to a format useful for statistical analysis and machine learning requires many steps. The MSOAC database, which aggregates data from multiple clinical trials in SDTM format, is a clean and high-quality data source. However, because source data in the MSOAC database are from different clinical trials, particular attention must be paid to harmonization, to ensure that data for the same concept are represented the same way. This can easily happen, for example, if different trials use different conventions to encode measurements.

In addition to harmonization, data must be converted from SDTM format to a data format that is suitable for analysis. SDTM represents observations in terms of variables encoding broad contextual data about the observation, while statistical analyses use tabular formats in which variables have implicit definitions and values for variables are plain data types. This tabular format is well described by the tidy data concept [33].

To carry out data processing, we have built a software library in `python` that makes it easier to harmonize data, convert between formats, and document and test the steps taken. This toolkit is described in [2], and has proven effective here.

1. Variables Used in Training

We selected 20 variables from the MSOAC database for further analysis (see Table II). These particular variables were chosen because they are frequently used in clinical trials, are relevant for predicting disease progression, and did not have a significant amount of missing data in MSOAC. The MSOAC database includes data on other questionnaires (e.g., 43 variables from SF-12, BDI-II, and RAND-36) and characterization of disability in medical history, but these data were rarely reported. In addition, MSOAC includes data on medications (16 variables) but these data lack dosing time information. Due to these limitations, we focused only on the variables described in Table II.

Although subjects were part of placebo control arms, they may have been taking medications that are part of the standard of care for treating MS or other conditions. Some of these medications, such as those considered to be disease modifying, are known to influence

the evolution or presentation of MS. Disease modifying therapies were rarely used (5% of subjects received them) and were likely provided as rescue therapies from relapses or adverse events. However, as dosing time information was not provided, these variables were not included in further analyses.

The Expanded Disability Status Scale (EDSS) is commonly used to assess disease progression in MS [13], but the score has a non-linear nature that makes it difficult to model. EDSS scores below 5.0 represents patients who are able to walk without aid. In this regime, the EDSS score is determined from the components of the Kurtzke Functional Systems Score (KFSS) components, a 7-component test assessing function in variety of bodily systems [13, 34, 35]. EDSS scores of 5.0 or greater, by contrast, are principally defined by the subject's ability to walk. Due to the nature of scoring, some values are much more likely than others, giving a very multi-modal marginal distribution of EDSS scores. We chose to model the individual KFSS components and ambulatory impairment of subjects separately rather than model the total score directly, allowing the model to learn the simpler constitutive covariates and the relationships between them.

Unfortunately, the MSOAC database does not record the ambulatory impairment component of EDSS directly. Instead, we infer this *ambulation score* from the EDSS score and encode it as an ordinal variable. The ambulation score was determined using

$$\text{ambulation score} = \begin{cases} 0, & \text{if EDSS} \leq 4.5 \\ 2 * (\text{EDSS} - 4.5), & \text{else} \end{cases}$$

Note that we are assuming that subjects with high EDSS scores have impaired ambulatory function. KFSS scoring rubrics do not provide consistent guidelines for assigning scores above the ambulation cutoff, indicating this assumption is valid. As a cross-check of this assumption, we took the subjects whose reported EDSS score was less than or equal to 4.5, and recomputed the score from the KFSS components. Although we found disagreements between the reported and recomputed EDSS scores in approximately 10% of cases, in a majority of these cases we were able to clearly identify an apparent mis-scoring of EDSS based on the reported KFSS component scores. As a result, we used the EDSS scores that we recomputed directly from the KFSS components in further analyses.

As with many statistical models, the performance of a CRBM is often improved by scaling the variables so that their values are of order 1. For binary and categorical covariates no

Name	Encoding	Domain	Normalizing Transformation
Baseline Age	Continuous	$18 - 72$	Standardization
Sex	Binary	$\{0, 1\}$	None
Race	Binary	$\{0, 1\}$	None
Region	1-hot, 3 labels	$\{0, 1\}$	None
Baseline	Binary	$\{0, 1\}$	None
MS type	1-hot, 3 labels	$\{0, 1\}$	None
# of relapses, 1 year before baseline	Ordinal	$\{0, 1, \ldots, 7\}$	Scaled by 1/7
# of relapses, 2 years before baseline	Ordinal	$\{0, 1, \ldots, 14\}$	Scaled by 1/14
Relapse events	Binary	$\{0, 1\}$	None
KFSS bowel and bladder system	Ordinal	$\{0, 1, \ldots, 6\}$	Scaled by 1/6
KFSS brain stem system	Ordinal	$\{0, 1, \ldots, 5\}$	Scaled by 1/5
KFSS cerebellar system	Ordinal	$\{0, 1, \ldots, 5\}$	Scaled by 1/5
KFSS mental system	Ordinal	$\{0, 1, \ldots, 5\}$	Scaled by 1/5
KFSS pyramidal system	Ordinal	$\{0, 1, \ldots, 6\}$	Scaled by 1/6
KFSS sensory system	Ordinal	$\{0, 1, \ldots, 6\}$	Scaled by 1/6
KFSS visual system	Ordinal	$\{0, 1, \ldots, 6\}$	Scaled by 1/6
Ambulation component of EDSS score	Ordinal	$\{0, 1, \ldots, 11\}$	Scaled by 1/11
Timed 25-Foot Walk	Continuous	$2 - 300$	Logit scaled to range
Nine-Hole Peg, dominant hand	Continuous	$10 - 260$	Logit scaled to range
Nine-hole Peg, non-dominant hand	Continuous	$10 - 260$	Logit scaled to range
Paced Auditory Serial Addition Test	Continuous	$0 - 60$	Logit scaled to range

TABLE II. Variables used in the model, their domain, encoding, and the normalizing transformation applied to them. Categorical variables (which are 1-hot encoded) and binary variables have no normalizing transformation, while ordinal variables are scaled to have a maximum value of 1. Continuous variables are normalized in different ways, but each results in a range where values are not much larger than 1.

normalization is applied, while for ordinal covariates we scale the covariate by its maximum so that the values range between 0 and 1. For continuous covariates we apply different normalizing transformations depending on the features of the covariate. Details of these transformations are provided in Table II.

For age, the "standardization" transform means we subtract the mean and divide by the standard deviation of the training set. The functional assessment variables (timed 25-

foot walk, 9-hole peg test, and paced auditory serial addition test (PASAT)) all use a logit transformation. The functional form of these transformations is

$$\tilde{x} = \log\left(\frac{x - (x_- + \delta)}{(x_+ + \delta) - x}\right), \tag{A1}$$

where $x_\pm$ are upper and lower limits of the variable (the range values given in the table) and δ is a buffer to prevent the argument from being too small or large (we use $\delta = 0.5$). These transformations, like the others used, are invertible, meaning we can transform from the representation of the data generated by the CRBM back to the natural representation of the data. For the PASAT test, we choose to model the variable as continuous (rounding the value when using the model output) rather than ordinal due to the large number of possible values.

Appendix B: CRBMs

1. Summary of CRBMs

Restricted Boltzmann machines (RBMs) [26, 29] are a well-known class of probabilistic neural networks capable of representing the complex interrelationships between variables in large datasets. They provide several features critical to modeling clinical data (see the supplementary material in [2]).

We use a particular type of Conditional Restricted Boltzmann Machine (CRBM) that provides a way to model time series data by leveraging the natural capabilities of Boltzmann machines. In fact, a CRBM is an RBM in which the organization of the visible units is particular to the time-dependence of the data. That is, without particular temporal labels on the visible units of the CRBM, the model is simply an RBM. These models can be trained the same way as RBMs, except that the temporal nature of the data potentially requires a reorganization of the training data.

In the original development [12, 41], a CRBM is defined as a particular kind of graphical model encompassing an RBM whose probability distribution depends conditionally on a number (k) of previous time points. That is,

$$p(\mathbf{x}_{t+k}|\mathbf{x}_{t+k-1}, \ldots, \mathbf{x}_t) = Z^{-1} \int d\mathbf{h}\, e^{-U(\mathbf{x}_{t+k}, \mathbf{h}|\mathbf{x}_{t+k-1} \ldots, \mathbf{x}_{t+1}, \mathbf{x}_t)}, \tag{B1}$$

in which $\mathbf{x}_t$ is the vector of variables at time index t, $\mathbf{h}$ is a vector of hidden variables, Z is a normalization constant, and $U(\cdot|\cdot)$ is the conditional energy function. K is the number of prior time points on which the Here, the conditional energy function is a family of functions parametrized by linear functions of k previous time points. Graphically, this is a model in which there are directed arrows extending from the previous time points to the hidden units (and, potentially, the current visible units) of an RBM (Figure 7).

The CRBM architecture described in [12, 41] is one directional; the future is conditioned on the past. However, there are a number of reasons why we may desire a model that is bidirectional so that we could predict the future given the past or predict the past given the future. For example, the ability to predict the past given the future is especially useful for imputing missing observations – e.g., knowledge that a subject had an EDSS score of 3.5 at their second visit provides some information on what their EDSS score likely was at the first visit, and we could use that information to impute the first visit value if it was not measured.

Our CRBM architecture differs from that in [12, 41] because it is bidirectional and it allows for some of the visible variables (such as sex or race) to be time-independent (denoted $\mathbf{x}_{\text{static}}$). To make the model bidirectional, we remove all connections between visible units and make all of the visible-hidden unit connections undirected (Figure 7). This leads to a joint probability distribution of $k+1$ time-adjacent vectors given by

$$p(\mathbf{x}_{t+k}, \ldots, \mathbf{x}_{t+1}, \mathbf{x}_t, \mathbf{x}_{static}) = Z^{-1} \int d\mathbf{h}\, e^{-U(\mathbf{x}_{t+k}, \ldots, \mathbf{x}_{t+1}, \mathbf{x}_t, \mathbf{x}_{\text{static}}, \mathbf{h})}\,, \tag{B2}$$

in which the energy function $U(\cdot)$ takes the form,

$$U(\mathbf{x}_{t+k}, \ldots, \mathbf{x}_{t+1}, \mathbf{x}_t, \mathbf{x}_{\text{static}}, \mathbf{h}) = \sum_{i=0,1,\ldots,k,\text{static}} \left[\sum_j a_{i,j}(\mathbf{x}_{t+i,j}) + \sum_{j\mu} W_{i,j\mu} \frac{\mathbf{x}_{t+i,j}}{\sigma_{i,j}^2} \frac{h_\mu}{\epsilon_\mu^2} \right] + \sum_\mu b_\mu(h_\mu)\,. \tag{B3}$$

Each unit of the visible and hidden layers has bias parameters determined by the choice of functions $a_{ij}(\cdot)$ and $b_\mu(\cdot)$, as well as scale parameters σ_{ij} and ϵ_μ. The connection between the layers is parameterized by the weight matrices $W_{i,j\nu}$. Understood as a single RBM, our CRBM contains the visible units for multiple time points coupled to a standard hidden layer. The visible units are organized as:

$$\mathbf{x}_{\text{CRBM}} = \mathbf{x}_{t+k} \oplus \mathbf{x}_{t+k-1} \oplus \cdots \oplus \mathbf{x}_t \oplus \mathbf{x}_{\text{static}}\,, \tag{B4}$$

where k is the time lag of the model and $\oplus$ signifies the direct sum. The static units are only used once over all time points, as they are constant over all times.

A bidirectional CRBM learns the complete joint probability distribution between all $k+1$ adjacent time points simultaneously, $p(\mathbf{x}_{t+k}, \ldots, \mathbf{x}_t, \mathbf{x}_{\text{static}})$. That means that *any* conditional sampling of the data may be performed, such as predicting the data for a time point given the previous k time points. In fact, the conditional distribution $p(\mathbf{x}_{t+k}|\mathbf{x}_{t+k-1} \ldots, \mathbf{x}_{t+1}, \mathbf{x}_t, \mathbf{x}_{static})$ has the same functional form as that in [12, 41]. Trajectories can be sampled by iteratively drawing from appropriate conditional distributions. For example, the trajectory $(\mathbf{x}_{t=\tau}, \ldots, \mathbf{x}_{t=1}) \,|\, (\mathbf{x}_{t=0}, \mathbf{x}_{static})$ is created by sampling from

$$
\begin{aligned}
\mathbf{x}_{t=1} &\sim p(\mathbf{x}_{t=1}|\mathbf{x}_{t=0}, \mathbf{x}_{static}) \\
\mathbf{x}_{t=2} &\sim p(\mathbf{x}_{t=2}|\mathbf{x}_{t=1}, \mathbf{x}_{t=0}, \mathbf{x}_{static}) \\
&\vdots \\
\mathbf{x}_{t=\tau} &\sim p(\mathbf{x}_{t=\tau}|\mathbf{x}_{t=\tau-1}, \ldots, \mathbf{x}_{t=\tau-k}, \mathbf{x}_{static})
\end{aligned}
$$

in that sequence. Note that we can sample also from a marginal distribution like $p(\mathbf{x}_{t=0}, \mathbf{x}_{static})$ simply by sampling from $p(\mathbf{x}_k, \ldots, \mathbf{x}_0, \mathbf{x}_{\text{static}})$ and discarding $(\mathbf{x}_k, \ldots, \mathbf{x}_1)$; therefore, it is possible to use the model both to simulate a cohort of subjects at baseline and to simulate the time-evolution of a cohort of subjects given their baseline measurements. In Figure 7, we show a schematic diagram highlighting the differences between an RBM, the bidirectional CRBM we describe here, and the directed CRBM originally described in [12, 41].

2. Training methods

As discussed in the main text, to train the CRBM we divided the 2395 subjects in the dataset up into 3 parts: training (50% of subjects), validation (20% of subjects), and test (30% of subjects). Training was done in two stages. First we train a large number (1296) models – each of which has a single hidden layer of ReLU units [42] – over a grid of hyperparameters and measure the performance of models according to a number of metrics. The hyperparameters of the best-performing model were used to train a model on the combination of the training and validation data.

We use a bidirectional CRBM with lag $k = 2$. As a result, the CRBM is trained on the data from adjacent triples of time points. If $\mathbf{x}_t$ is the vector of time-dependent variables

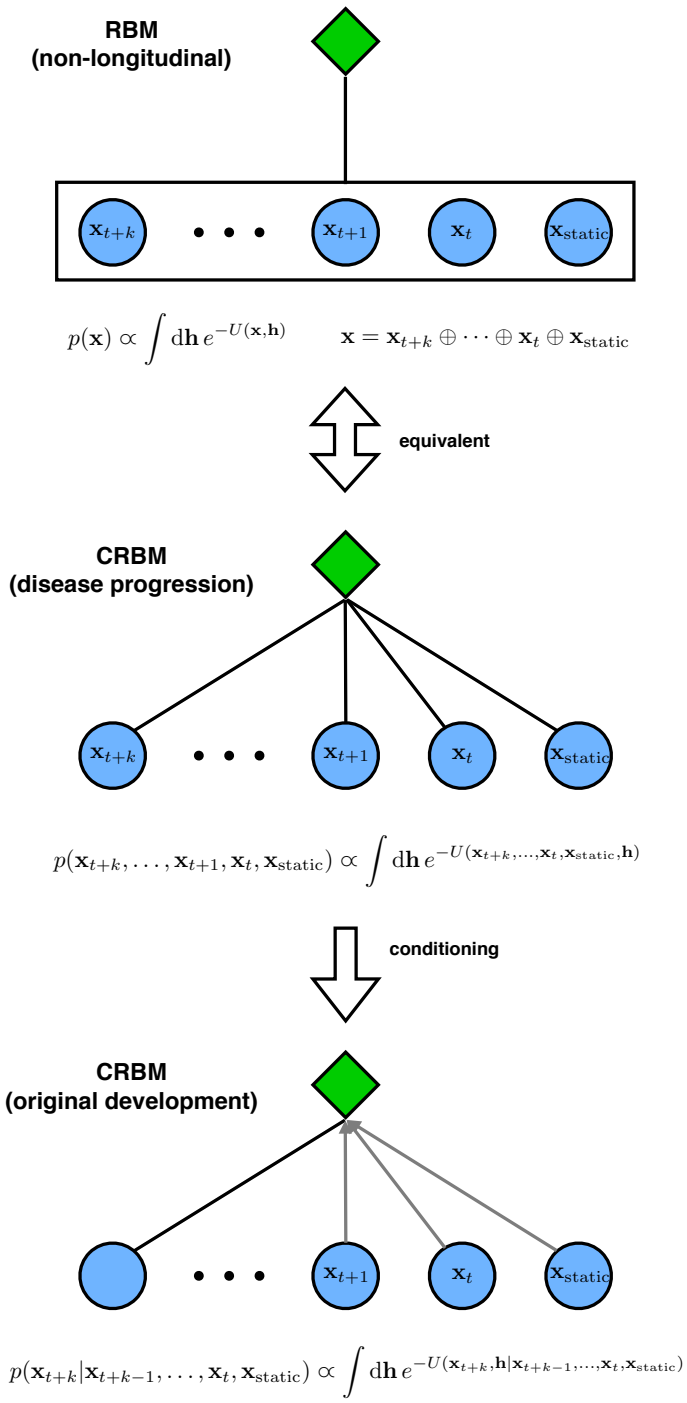


FIG. 7. **Architectures of RBMs and CRBMs.** A schematic architecture of the CRBM used in this work is shown in the middle, compared to an RBM, which is an equivalent architecture, and the original CRBMs, which are described in the text. Both the RBM and CRBM used here have completely undirected connections, while the original CRBM has directed connections except for the last time point. Directed connections are shown as gray arrows.

for a patient at time t (in months) and $\mathbf{x}_{\text{static}}$ is the vector of static variables for the same patient, then the visible units used to train the CRBM are $\mathbf{v} = \{\mathbf{x}_{t+6}, \mathbf{x}_{t+3}, \mathbf{x}_t, \mathbf{x}_{\text{static}}\}$ – a concatenation of the data from the adjacent time points t, and $t+3$ months, and $t+6$ months with the static variables represented only once. Transforming the data from a subject into this triplet format involves 3 steps:

1. For each subject we create vectors of data from all possible sets of 3 consecutive visits. For example if a subject has data through 12 months we create vectors for 0, 3, and 6 months, 3, 6, and 9 months, and 6, 9, and 12 months. Repeated occurrences of the static (non-longitudinal) covariates are dropped so that they only appear once in the vector.

2. Any vector that has no longitudinal data for the last visit is removed in order to remove cases in which the subject has missed visits.

3. The data are standardized according to the transformations discussed in Appendix A.

Each subject typically has trajectories much longer than the 6-month time window (trajectories up to 57 months are used). As a result, there are multiple triplets per subject, and the number of samples in the combined training and validation datasets (11129) is much larger than the number of subjects. Before training, samples are randomly shuffled so that minibatches contain a mixture of subjects and times.

The CRBM is trained using stochastic gradient descent to minimize an objective function that simultaneously aims to maximize the composite likelihood of the data and to minimize the ability of a classifier to discriminate between samples drawn from the model and samples taken from the training dataset [10]. Let $\mathcal{V}$ denote the set of all triplets of visible variables in the training set. The logarithm of the composite likelihood of the data is

$$\mathcal{L}(\theta) := \sum_{\mathbf{v} \in \mathcal{V}} \log p(\mathbf{v}; \theta)\,, \tag{B5}$$

in which θ denotes the parameters of the RBM. Note that two neighboring triplets from the same subject will have repeated entries and, therefore, are not independently distributed. However, we are making an approximation in which we are treating these neighboring triplets as independently distributed. Therefore, it is more appropriate to refer to $\mathcal{L}(\theta)$ as the logarithm of a composite likelihood [43] than a true likelihood.

In [10], we showed that likelihood-based approaches to training RBMs have a tendency to oversmooth the estimated distribution. To prevent this oversmoothing, we train a discriminator $q(\text{data}|\mathbf{h})$ to classify a vector of hidden unit activities as corresponding to a visible vector drawn from the data or drawn from the model. In this case, the classifier is a random forest trained on the hidden unit activities of the previous minibatch. Then, we aim to minimize an adversarial objective

$$\mathcal{A}(\theta) := -\int d\mathbf{h} p(\mathbf{h};\theta)\left(2\,q(\text{data}|\mathbf{h}) - 1\right)\,, \tag{B6}$$

which means that we aim to train the RBM so that it generates samples that the discriminator classifies as coming from the data distribution. This concept is similar to approaches used to train Generative Adversarial Networks in machine learning [30].

The final objective function $\mathcal{C}(\theta)$ is a linear combination of log-composite likelihood $\mathcal{L}(\theta)$ and adversarial $\mathcal{A}(\theta)$ objectives,

$$\mathcal{C}(\theta) = -\gamma\mathcal{L}(\theta) + (1-\gamma)\mathcal{A}(\theta)\,, \tag{B7}$$

in which γ is a parameter – selected as part of the hyperparameter grid search – weighing the relative importance of the two objectives. We minimize this objective function with stochastic gradient descent using the ADAM optimizer [44].

3. Temperature driven sampling

Computing the gradients of the objective function (Equation B7) requires computing various moments of the model distribution [10]. Unfortunately, the moments of a distribution defined by an RBM are typically intractable and can only be approximated using Monte Carlo methods. Specifically, it is possible to sample from an RBM using block Gibbs sampling by iteratively drawing $\mathbf{h} \sim p(\mathbf{h}|\mathbf{v})$ and $\mathbf{v} \sim p(\mathbf{v}|\mathbf{h})$. This creates a Markov chain that converges to the stationary distribution, but the rate of convergence may be very slow. To address this, we employ a simple approach to improve the mixing of the CRBM when generating digital subjects or digital twins.

The underlying RBM that the CRBM is built upon can integrate the concept of temperature, which enters the joint probability distribution as $p_\beta(\mathbf{v},\mathbf{h}) = Z_\beta^{-1} e^{-\beta E(\mathbf{v},\mathbf{h})}$, where β is the inverse temperature and $\mathbf{v}$ and $\mathbf{h}$ are the visible and hidden units of the model.

By drawing the inverse temperature from an autoregressive gamma process [45] with mean 1, a small standard deviation, and a non-zero autocorrelation, we can improve the rate of mixing [10]. This allows the Markov chain to mix more quickly when $\beta \ll 1$, but keeps the stationary distribution close to the case in which $\beta = 1$. Unlike parallel tempering [46, 47], this algorithm for *driven sampling* does not sample from the exact distribution, and instead samples from a *similar* distribution to the RBM that has fatter tails. However, this approach adds only a small computation cost and appears to improve training outcomes (in this and other applications); see [10] for details.

Because varying the temperature during sampling creates some bias in the model, we use different approaches during training and testing. During training, we let the temperature vary as described above and set the autocorrelation to 0.9 to ensure that the inverse temperature evolves slowly and the system can remain close to equilibrium. During testing, we set the autocorrelation to zero and linearly anneal the standard deviation of β to 0 to ensure that the end of the Markov chain samples from the distribution with $\beta = 1$.

Appendix C: Hyperparameter Sweep and Model Selection

1. Hyperparameter Grid Search

In order to train a CRBM, we need to define multiple hyperparameters related to the architecture (such as the number of hidden units) and the training protocol (such as the size of the gradient steps, or learning rate). It's not clear how to choose these hyperparamaters *a priori*. As a result, we performed a grid search over various choices for the hyperparameters (see Table III), training a total of 1296 models with different hyperparameters. Then, we selected one of these models for further analysis based on its performance on the validation dataset.

2. Selecting the Best Performing Model

A CRBM is a generative model that describes the probability distribution of many variables over time. Therefore, there are many potential metrics that one could use to define model performance. We addressed this by selecting a model from the hyperparameter grid search that performed well, on average, across a number of metrics.

Let M_{ij} denote the rank of model i on metric j. That is, $M_{ij} = 1$ if model i is the best performing model on metric j, and $M_{ij} = N$ if model i is the worst performing model on metric j. We used a two-step minimax procedure to select the best performing model on these metrics. In step one, we sorted the models based on their worst performance on any metric, $\text{MaxRank}_i = \max_j (M_{ij})$, and discarded all models outside of the lowest quartile. In the second step, we focused on a subset J of the metrics and sorted the models based on their worst performance in this subset $\text{MaxSubsetRank}_i = \max_{j \in J} (M_{ij})$. The model with the lowest MaxSubsetRank_i was selected as the best performing model overall.

This two step procedure ensured that selected model (1) did not perform poorly on any metric and (2) performed especially well on a subset of metrics. The hyperparameters of the best performing model are shown in bold in Table III. These hyperparameters are used to train a model on the combination of the training and validation datasets.

Equal and time-lagged correlations between variables computed from a model that per-

Hyperparameter	**Values**
number of hidden units	**27**, 53
number of epochs	1250, **2500**, 5000, 10000
batch size	**557**, 1113, 2226
initial learning rate	**1/212**, 1/106, 1/53
driven sampling σ_β	0, **0.15**, 0.3
ℓ_2 weight penalty	$\mathbf{10^{-4}}$, 10^{-3}
adversary weight	0, **0.3**, 0.7
Monte Carlo steps (sampling)	**100**
optimizer	**ADAM**

TABLE III. The hyperparameter grid used to train the CRBM. The grid was defined as the Cartesian product of the values listed across hyperparameters. The value of each hyperparameter for the selected model is shown in bold. The grid of batch sizes was chosen so that there were 5, 10, or 20 batches per epoch in the combined training and validation datasets. The grid of hidden units was chosen as 1/2 or 1 times the the number of visible units rounded up to the nearest integer. The grid of learning rates was chosen as 1/4, 1/2, or 1 times the inverse of the number of visible units.

forms well should be the same as those computed from actual data. Therefore, we defined some metrics to evaluate the performance of a model based on the agreement between the lag-l correlations computed from the model and those computed from the data for $l = 0, 1, 2, 3$. Computing these correlations is complicated by the fact that some observations are missing. Therefore, let $x^s_{a,t,i}$ denote the value of variable a at time t in subject i from source $s \in \{data, twin\}$; that is, $x^{twin}_{a,t=0,i} = x^{data}_{a,t=0,i}$. In addition, let $I_{a,t,i} = 1$ if $x^{data}_{a,t,i}$ was observed, and $I_{a,t,i} = 0$ if $x^{data}_{a,t,i}$ was missing. Then, we computed the lag-ℓ autocovariance between variables a and b from source s using

$$C^s_{a,b,\ell} = \frac{\sum_i \sum_t \left(x^s_{a,t,i} - \frac{\sum_i \sum_t x^s_{a,t,i} I_{a,t,i} I_{b,t+\ell,i}}{\sum_i \sum_t I_{a,t,i} I_{b,t+\ell,i}} \right) \left(x^s_{b,t+\ell,i} - \frac{\sum_i \sum_t x^s_{b,t+\ell,i} I_{a,t,i} I_{b,t+\ell,i}}{\sum_i \sum_t I_{a,t,i} I_{b,t+\ell,i}} \right)}{\sum_i \sum_t I_{a,t,i} I_{b,t+\ell,i}} . \tag{C1}$$

As a metric, we defined the squared correlation coefficient between the lag-ℓ autocovariances computed from the data and the twins as

$$R^2_\ell = 1 - \frac{\sum_{a \le b} (C^{data}_{a,b,\ell} - C^{twin}_{a,b,\ell})^2}{\sum_{a \le b} (C^{twin}_{a,b,\ell} - \frac{1}{p(p-1)} \sum_{a \le b} C^{twin}_{a,b,\ell})^2} , \tag{C2}$$

in which p is the number of variables. These correlation coefficients are computed for $\ell = 0, 1, 2, 3$ ($\ell = 0$ are the equal-time and $\ell > 0$ are the lagged autocorrelations).

The metrics computed from the equal and time-lagged autocovariances capture a general notion of goodness-of-fit for the CRBM, but we would also like to ensure that the model performs well on outcome measures that are frequently used as endpoints in clinical trials: the probability of a relapse at each visit, the change in the Expanded Disability Status Scale (EDSS) over 18 months, and 6-month, 1-year, and 2-year Confirmed Disability Worsening (CDW) scores. For relapses, which is one of the time-dependent variables in the model, we defined metrics in terms of the area under the receiver operating characteristic curve (AUC) that measures the model's ability to predict relapses at each visit from 3 to 18 months.

The total EDSS score is not one of the variables directly included in the model, but it can be computed because all of its components are included in the model. Let s^{data}_i denote the change in the EDSS score for subject i over 18 months, and let s^{twin}_{ik} denote the change in the EDSS score for the kth digital twin of subject i. We defined an EDSS score metric as

$$t_{\text{EDSS}} = \frac{\left| \frac{1}{n} \sum_i (s^{data}_i - \frac{1}{K} \sum_k s^{twin}_{ik}) \right|}{\sqrt{\frac{1}{n} \sum_i (s^{data}_i - \frac{1}{n} \sum_j s^{data}_j)^2}} \sqrt{n} , \tag{C3}$$

in which the sums only include subjects for which the 18-month change in EDSS was observed and n is the number of those subjects. We used $K = 10$ digital twins for each subject.

Similar to EDSS, CDW is not one of the variables directly included in the model but it can be computed from the model variables. A patient experiences a period of CDW if he/she has an increase of at least 1 point in EDSS (0.5 points if the initial EDSS score is greater than 6) that is sustained for (a) a continuous period of 6-months during the next year, (b) a continuous period of 3-months during the next 2 years, or (c) a continuous period of 6-months during the next 2 years. That is, there are three different definitions of CDW that we denote as CDW-a, CDW-b, and CDW-c. Let a_i^{data} denote CDW-a for subject i, and let a_{ik}^{twin} denote CDW-a for the kth digital twin of subject i. We defined a metric for CDW-a as

$$t_{\text{CDW-a}} = \frac{\left|\frac{1}{n}\sum_i (a_i^{data} - \frac{1}{K}\sum_k a_{ik}^{twin})\right|}{\sqrt{\frac{1}{n}\sum_i (a_i^{data} - \frac{1}{n}\sum_j a_j^{data})^2}}\sqrt{n}\,, \tag{C4}$$

where the sums only include subjects with primary-progressive multiple sclerosis for whom CDW-a could be computed, and n is the number of those subjects. Metrics for CDW-b and CDW-c were defined similarly. We used $K = 10$ digital twins for each subject.

All of the metrics – those related to the autocovariances, relapses, the 18-month change in EDSS, and the CDW scores – were used in step (1) of the model selection procedure. Only those metrics related to the relapses, the 18-month change in EDSS, and the CDW scores were used in step (2) of the model selection procedure.

Appendix D: Model Analysis

Figures 4, 5, and 6 present a number of analyses to assess the goodness-of-fit of the CRBM by comparing statistics computed from the subjects in the test dataset to statistics computed from their digital twins. This section provides more details on how those analyses were performed.

Let $x_{a,t,i}^{data}$ denote the value of variable a at time t that was observed for subject i and let $x_{a,t,i}^{twin}$ denote the corresponding value for a digital twin sampled from the model. By definition, $x_{a,t=0,i}^{data} = x_{a,t=0,i}^{twin}$. In addition, let $I_{a,t,i} = 1$ if $x_{a,t,i}^{data}$ was observed, and $I_{a,t,i} = 0$ if $x_{a,t,i}^{data}$ was missing.

Figures 4 compares the means, standard deviations, and autocovariances computed from the data to those computed from the digital twins. The means were computed as,

$$\mu_{a,t}^s = \frac{\sum_i x_{a,t,i}^s I_{a,t,i}}{\sum_i I_{a,t,i}}\,, \tag{D1}$$

the standard deviations were computed as,

$$\sigma^s_{a,t} = \sqrt{\frac{\sum_i (x^s_{a,t,i} - \mu^s_{a,t,i})^2 I_{a,t,i}}{\sum_i I_{a,t,i}}}, \tag{D2}$$

and the lag-ℓ autocovariances were computed using Equation C1 for sources $s \in \{data, twin\}$. Each of the sums only includes subjects from the test data set.

We assessed the agreement between the means and standard deviations computed from the data and those computed from their digital twins using a Theil-Sen regression. In principle, a slope of a regression line can be estimated from any pair of data points. However, this estimate for the slope is very noisy. Therefore, a Theil-Sen estimate computes many estimates for the slope from multiple pairs of data points (up to all pairs) and then uses the median slope. This procedure is robust to outliers, which is helpful here because the means and standard deviations for different variables can have different scales.

The lag-ℓ autocovariances can be set to the same scale by converting them to autocorrelations; i.e., by normalizing relative to the appropriate variances

$$\rho^s(a, b, \ell) := \frac{C^s(a, b, \ell)}{\sqrt{C^s(a, a, \ell) C^s(b, b, \ell)}} \tag{D3}$$

for source $s \in \{data, twin\}$. As a result, the problems that we encountered with scale differences in the means and standard deviations doesn't apply to analyzing the agreement between the autocorrelations computed from the data and those computed from their digital twins. However, each value of $\rho^s_{a,b,\ell}$ was only computed using pairs of data for which both variables a and b were observed. As a result, some autocorrelations were estimated with more precision than others. Therefore, we estimated the relationship by minimizing the weighted least-squares cost function

$$\mathrm{L}S_\ell := \sum_{a,b} f_{a,b,\ell} (C^{data}_{a,b,\ell} - \alpha - \beta C^{twin}_{a,b,\ell})^2 \tag{D4}$$

in which $f_{a,b,\ell} = \sum_t \sum_i I_{a,t,i} I_{b,t+\ell.i}$. We report the estimated coefficients and the coefficient of determination in the main text.

Although the CRBM provides a model for the probability distribution of $x_{a,t}$, this distribution is generally intractable. However, we can estimate properties of this distribution using samples drawn with MCMC algorithms. Let $x^{twin}_{a,t,i,k}$ be a set of $k = 1, \ldots, K$ digital twins matched to subject i sampled from the CRBM such that $x^{twin}_{a,t=0,i,k} = x^{data}_{a,t=0,i}$ for all k.

We can use the digital twins to compute an approximate a p-value,

$$p_{a,t,i} = 1 - \frac{1}{K} \sum_k \delta(x^{data}_{a,t,i} \leq x^{twin}_{a,t,i,k}) \tag{D5}$$

that describes the likelihood of observing a value more extreme than $x^{data}_{a,t,i}$ under the model distribution. Here, $\delta(\cdot)$ equals 1 if the argument is true and 0 otherwise. If the CRBM fits the data well, then $p_{a,t,i}$ will be uniformly distributed on $(0, 1)$ or, alternatively,

$$\varphi_{a,t,i} = \Phi^{-1}(p_{a,t,i})\,. \tag{D6}$$

will be normally distributed with mean 0 and standard deviation 1. Therefore, we compute the mean and variance by aggregating across subjects

$$\mathrm{E}[\varphi_{a,t}] = \frac{1}{N} \sum_i \varphi_{a,t,i}$$
$$\mathrm{Var}[\varphi_{a,t}] = \frac{1}{N} \sum_i (\varphi_{a,t,i} - \mathrm{E}[\varphi_{a,t}])^2$$

which we plot in Figure 5.

If the mean is above (below) 0, then the mean of the model is lower (higher) than the mean of the data. If the variance is smaller (larger) than 1, then the variance of predictions of the model is larger (smaller) than the data. Note that the relationship between the moments and the bias of the model is inverted due to the fact that the statistic is comparing the data value to the model distribution. To test significance of deviations from a standard normal distribution, we compute a Kolmogorov-Smirnov statistic between $\mathcal{N}(\mathrm{E}[\varphi_{a,t}], \mathrm{Var}[\varphi_{a,t}])$ and $\mathcal{N}(0, 1)$. We adjusted the significance level of the Kolmogorov-Smirnov test using a Bonferroni correction on the number of tests being performed, which is equal to the number of covariates evaluated (12) multiplied by the number of time points (12), for a scaling factor of 144 (meaning a comparison is significant if the p-value is less than 0.05/144).

Finally, we introduced a concept of statistical indistinguishability in the main text by saying that digital subjects, or digital twins, are statistically indistinguishable from actual subjects if a statistical procedure designed to classify a given clinical record as a digital subject or an actual subject performs consistently with random guessing. Therefore, in order to test if the data generated by the CRBM are statistically indistinguishable from the data in the test set, we should train a classifier to distinguish actual subjects from their digital twins and then assess how well it performs.

Figure 6 shows the area under the receiver operating characteristic curve (AUC) for logistic regression models trained to distinguish actual subjects from their digital twins. An AUC score of 0.5 indicates that the logistic regression cannot do better than random chance at determining whether data comes from actual subjects or their digital twins, while an AUC of 1.0 indicates that logistic regression can perfectly identify the source of a clinical record.

Different subjects in the dataset were observed for different durations, so we trained separate logistic regression models on each visit or on the difference between consecutive visits. In each case, we computed the mean AUC taken over 100 different simulations with different digital twins for each simulation; the error bars on the points show the standard deviation over simulations. The AUC for each simulation was estimated using 5-fold cross validation. Missing data from actual subjects is mean imputed; the corresponding values for the digital twins are assigned the same mean values to avoid biasing the logistic regression models.