8-2021

Improving Reinforcement Learning Techniques for Medical Decision Making

Matthew Baucum
University of Tennessee, Knoxville, mbaucum1@vols.utk.edu

To the Graduate Council:

I am submitting herewith a dissertation written by Matthew Baucum entitled "Improving Reinforcement Learning Techniques for Medical Decision Making." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Industrial Engineering.

Anahita Khojandi, Major Professor

We have read this dissertation and recommend its acceptance:

Rama K Vasudevan, John E Kobza, Jim Ostrowski

Accepted for the Council:

Dixie L. Thompson

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Improving Reinforcement Learning Techniques for Medical Decision Making

A Dissertation Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Matthew Baucum

August 2021

Acknowledgments

I would like to express my sincere gratitude to my advisor, Anahita Khojandi, for her support in preparing this dissertation. This work would not have been possible without her expertise, guidance, and encouragement.

Research in Chapter 1 is partially supported by Science Alliance, The University of Tennessee, and the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the U.S. Department of Energy. The material in Chapter 1 is reprinted from [11], with permission from IEEE (©2020 IEEE): M. Baucum, A. Khojandi, and R. Vasudevan, "Impriving Deep Reinforcement Learning with Transitional Variational Autoencoders: A Healthcare Application," in IEEE Journal of Biomedical and Health Informatics, doi: 0.1109/JBHI.2020.3027443. In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of the University of Tennessee's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink. If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

The material in Chapter 2 is partially supported by the Joint Directed Research and Development program at Science Alliance, University of Tennessee. Data collection and sharing for this project was funded by the Alzheimer's Disease Neuroimaging Initiative (ADNI) (National Institutes of Health Grant U01 AG024904) and DOD ADNI (Department of Defense award number W81XWH-12-2-0012). This research was sponsored by the

Laboratory Directed Research and Development Program of Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the US Department of Energy under contract DE-AC05-00OR22725.

The material in Chapter 3 is partially supported by Science Alliance, The University of Tennessee, and the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the U.S. Department of Energy.

The material in Chapter 4 is partially supported by the Science Alliance, The University of Tennessee, and by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory managed by UT-Battelle, LLC, for the U.S. Department of Energy (DOE). The authors would also like to acknowledge Dr. Fatta Nahab for providing the PKG data used for all analyses.

Abstract

Reinforcement learning (RL) is a powerful tool for developing personalized treatment regimens from healthcare data. In RL, an agent samples experiences from an environment (such as a model of patient health) to learn a policy that maximizes long-term reward. This dissertation proposes methodological and practical developments in the application of RL to treatment planning problems.

First, we develop a novel time series model for simulating patient health states from observed clinical data. We use a generative neural network architecture that learns a direct mapping between distributions over clinical measurements at adjacent time points. We show that this model produces realistic patient trajectories and can be paired with on-policy RL to learn effective treatment policies.

Second, we develop a novel extension of hidden Markov models, which are commonly used to model and predict patient health states. Specifically, we develop a special case of recurrent neural networks with the same likelihood function as a corresponding discrete-observation hidden Markov model. We demonstrate how combining our model with other predictive neural networks improves disease forecasting and offers novel clinical interpretations compared with a standard hidden Markov model.

Third, we develop a method for selecting high-performing reinforcement learning-based treatment policies for underrepresented patient subpopulations using limited observations. Our method learns a probability distribution over treatment policies from a reference patient group, then adapts its recommendations using limited data from an underrepresented patient group. We show that our method outperforms state-of-the-art benchmarks in selecting effective treatment policies for patients with non-typical clinical characteristics, and predicting these patients' outcomes under its policies.

Finally, we use RL to optimize medication regimens for Parkinson's disease patients using high-frequency wearable sensor data. We build an environment model of how patients' symptoms respond to medication, then use RL to recommend optimal medication types, timing, and dosages for each patient. We show that these patient-specific RL-prescribed medication regimens outperform physician-prescribed regimens and provide clinically defensible treatment strategies. Our framework also enables physicians to identify patients who could could switch to lower-frequency regimens for improved adherence, and to identify patients who may be candidates for advanced therapies.

Table of Contents

List of Tables

List of Figures

Chapter 1

Improving Deep Reinforcement Learning with Transitional Variational Autoencoders: A Healthcare Application

1.1 Introduction

The reinforcement learning (RL) paradigm offers a promising way to optimize treatment regimens for acute and chronic conditions. RL generally seeks to learn a policy that maps an environmental state (e.g., a patient's health status) to an optimal action, with the goal of maximizing long-term expected reward. In recent years, researchers have applied RL and deep RL (DRL) techniques to optimize treatment strategies in a large array of diseases, e.g., for HIV medication regimens [100], sepsis treatment [111], drug therapy for myeloma patients [163], and anticoagulant medication administration [93].

Healthcare RL is complicated by the fact that treatment policies often cannot be learned from actual patients in real time, for obvious ethical reasons. There are two primary approaches for addressing this limitation. The first is to learn policies directly from existing datasets (e.g., [111]). This approach, known as off-policy reinforcement learning, allows

an agent to map patient states to optimal treatment actions using data generated from a separate policy (e.g., from the patient's original physician).

The second approach is to learn an 'environment model' from historical patient data that provide an RL agent with feedback on any state/action pair. This allows for on-policy learning (where an agent can receive feedback on its own policy), and is considered state-of-the-art and has been shown to outperform off-policy learning [83]. Below, we review existing approaches that may be used for learning environment models from patient data. We then introduce our proposed model, the *transitional variational autoencoder* (tVAE), which addresses several shortcomings of existing methods.

1.1.1 Existing approaches for RL environment modeling

One of the most common modeling approaches using historical patient data is hidden Markov models (HMMs) [12], which have been used for modeling disease progression and treatment response for HIV [100], cancer [163], and glaucoma [74]. HMMs model patient measurements as manifestations of a discrete set of latent disease states, and estimate the transitions between such states as well as the measurement distributions associated with each state. For continuous measurements, HMMs typically assume Gaussian emission distributions for tractability [100, 163]. HMMs' intuitive structure and relatively interpretable parameters make them appealing for fitting generative models to patient data.

Long short-term memory (LSTM) networks are another common patient modeling approach, and are specifically designed to capture long-term dependencies in their input sequences. Unlike HMMs, LSTMs do not require the Markov assumption, i.e., observations are allowed to depend on multiple previous disease states, not just the previous state. Because of this flexibility, LSTMs have been used to model Alzheimer's patient progression [43], blood anticoagulent therapy [93], and sepsis [73]. While LSTMs are predictive (rather than generative) models, they can easily be used as generative models by treating their output layers as probability distributions over discrete patients states, or by treating their outputs as the mean of a continuous emission distribution [152].

Another attractive approach for learning fully-responsive environment models from patient data is generative neural networks. Unlike standard neural networks, which optimize

predictive accuracy, generative neural networks attempt to learn underlying representations for their training data that can be used to generate novel, synthetic data points from a lower-dimensional 'latent' distribution. Generative neural networks have already shown promise in generating synthetic patient profiles based on real data [35, 24].

The two subclasses of generative neural networks are variational autoencoders (VAEs; [65]) and generative adversarial networks (GANs; [44]). Both models train a decoder network to transform a latent state vector into a realistic feature vector that resembles training examples. VAEs do this by learning to reproduce training examples from latent state distributions that are conditioned on those training examples. GANs learn to generate synthetic data from random latent states that a second 'discriminator' network cannot distinguish from real training data. Conditional VAEs (CVAEs; [64]) and conditional GANs (CGANs; [81]) condition their decoder networks on training labels, and learn to generate realistic data *given* a particular label. CVAEs have been used to generate handwritten digits [130] and realistic object trajectories [143]. CGANS have been used for face aging [7] and medical image analysis [115].

1.1.2 Limitations of existing approaches

Existing approaches for learning environment models from patient data suffer from several shortcomings, which this work aims to address. HMMs and LSTMs both require that the functional form of the emission distribution be specified *a priori*, and HMMs are often limited in practice to Gaussian emission distributions for tractability. In addition, HMMs are typically implemented with discrete state spaces, which may introduce unwanted discontinuities in model-generated patient measurements. LSTMs, being predictive rather than generative networks, are also limited by their inability to estimate emission distribution variances during model fitting.

In general, CVAEs and CGANs encourage a strong dependence between consecutive patient measurements (i.e., between the desired output and conditioning label). Therefore, these models may simply learn to approximate patient measurements by reproducing their previous measurements, without adding sufficient variability to their simulations. Furthermore, in particular to the CVAE architecture, latent distributions learned during

training must be replaced with standard Gaussian distributions during testing, possibly yielding inaccurate patient trajectories.

1.1.3 Objective and contributions

The purpose of this work is to introduce *transitional variational autoencoders* (tVAEs), and to examine their ability to learn realistic disease progression environment models in the context of DRL. We benchmark tVAEs against a set of best-known and most-used generative models for patient data, specifically HMMs, LSTMs, CVAEs, and CGANs.

The tVAE adapts the VAE structure to the type of longitudinal patient data frequently encountered in the healthcare domain by learning to map patient states to distributions over states at the following time point. In doing so, tVAEs address several shortcomings of existing methods for patient modeling. Unlike HMMs, tVAEs use continuous latent state spaces, and because they map their latent state spaces to model outputs using neural networks, they avoid placing distributional assumptions on the observed data. Furthermore, unlike LSTMs, tVAEs incorporate randomness at the latent state level rather than requiring the post-hoc application of emission distributions to model outputs. Unlike CGANs, tVAEs use a stochastic latent layer to separate inputs and outputs, which ensures that random variability is incorporated into the model. In contrast to CVAEs, tVAEs do not replace learned latent distributions with standard Gaussian distributions during testing, and therefore have identical training and testing architectures. Therefore, our proposed generative neural network approach, tVAE, contributes to the healthcare RL literature by 1) placing no distributional assumptions on the observed clinical data, 2) allowing for a continuous disease state space, and 3) building randomness directly into the model in a way that is specifically designed for medical time series.

We showcase the contributions of our proposed method in the context of optimal medication administration planing (dosage and timing) in the intensive care unit (ICU). Specifically, we use an existing dataset of anticoagulant medication administration records (timing/dosage) and the corresponding outcomes in a cohort of ICU patients to learn and prescribe optimal medication administration policies. We use a tVAE and four benchmark methods (HMM, LSTM, CGAN, CVAE) as environment models capable of generating

synthetic patient trajectories and training medication dosage policies through on-policy RL. To assess each model's validity, we compare their simulated patient trajectories with actual patient data. We also assume that a valid patient model should be able to facilitate learning effective treatment policies. Thus, we also use each environment model for training an optimal heparin dosage policy through DRL, and assess each policy's ability to maintain desirable blood coagulation levels for patients. In doing so, we use an 'ensemble' environment based on all five generative models (tVAE, HMM, LSTM, CGAN, CVAE) as a proxy for patients' responses to heparin.

The remainder of the chapter is organized as follows. Section 2.3 discusses the methods. Specifically, Section 1.2.1 discusses the dataset, Sections 1.2.2 and 3.2.5 formally introduce the tVAE and the benchmark models, respectively, and Section 1.2.4 describes the RL problem formulation and solution algorithm. Results are provided in Section 2.4. Section 1.3.1 provides patient descriptive statistics, and Sections 1.3.2 and 1.3.3 assess each environment model's ability to generate realistic patient data and train personalized RL policies, respectively. Section 2.5 discusses the results. Lastly, Section 1.5 concludes the chapter.

1.2 Methods

In this section, we first discuss our dataset, and introduce our proposed tVAE and benchmark models. We then present our method for using DRL to learn optimal dosage policies from each environment model.

1.2.1 Dataset

Data consists of intensive care unit data from 2,067 patients in the publicly available MIMIC dataset [57].

The dataset consists of hourly measurements of each patient's heparin dosage, aPTT values, and 13 other clinical variables - arterial carbon dioxide, heart rate, creatinine, the Glasgow Coma Score, hematocrit, hemoglobin, international normalized ratio of prothrombin, platelet count, prothrombin time, arterial oxygen saturation, temperature, urea, and white blood cell count. These measurements are included in our study because of

their general relevance to patient health or their specific relevance to blood anticoagulation [93]. Note that this dataset was also used for analysis in [93].

Because heparin dosages are weight-based (expressed as mL/kg of patient body weight [79]), each patient's hourly heparin dosage is divided by their weight and discretized into six categories [93]. Heparin dosages of zero (no administration) constituted one dosage category, with the remaining dosages split along the 20th, 40th, 60th, and 80th percentiles into five additional categories, resulting in six heparin dosage groups $a_t \in \{0, 1, 2, 3, 4, 5\}$ for all time t.

As in [93], missing heparin dosages are imputed according to sample-and-hold interpolation. Missing aPTT measurements are imputed according to a neural network, which predicted aPTT values from the remaining 13 clinical values, heparin dosage, as well as the patients' gender, age, and weight. It is worth noting that removing gender and age from the aPTT imputation model did not degrade predictive accuracy, suggesting that any confounding influences of these demographic factors on patient aPTT are accounted for by the other clinical variables. The final dataset consists of 54,906 measurements (all variables standardized), with an average of 26.6 sequential hourly measurements for each of the 2,067 patients.

1.2.2 Transitional variational autoencoders (tVAE)

1.2.2.1 tVAE overview

The proposed transitional variational autoencoder (tVAE) adapts the structure of standard VAEs to model transitions between consecutive patient measurements. A standard VAE seeks to maximize the probability of a dataset, $P(X)$, by assuming that entries in X are distributed according to some function of a latent normal variable Z, which is assigned a prior distribution $P(Z) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ [65]. That is, $X \sim g(Z)$ such that maximizing $P(X)$ is equivalent to maximizing $\mathbb{E}_{Z \sim P(Z)} P(X|Z)$. Since large regions of $P(Z)$ may yield near-zero values of $P(X|Z)$, VAEs assume an approximator distribution $Q(Z|X)$ that conditions Z on X and thus identifies values of Z that are likely to have produced X. The model learns a function h that maps X to $Q(Z|X)$, i.e., $h : X_t \mapsto Q(Z_t|X_t)$ at time t, typically assuming $Q(Z|X) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu} = h_\mu(X)$ and $\boldsymbol{\Sigma}$ is a diagonal matrix with entries

$\boldsymbol{\sigma^2} = h_\sigma(X)$. The model also learns a function g that maps $Q(Z|X)$ to $P(X|Z)$, i.e., $g : Q(Z_t|X_t) \mapsto P(X_t|Z_t)$ for time t. In other words, g learns to 're-map' a distribution of latent values to a distribution over observations.

If a suitable approximator distribution is found, then $\mathbb{E}_{Z\sim Q(Z|X)}\, P(X|Z)$ can be substituted for $\mathbb{E}_{Z\sim P(Z)}\, P(X|Z)$. This substitution yields the following relation between $P(X)$ (the quantity to be maximized) and $\mathbb{E}_{Z\sim Q(Z|X)}\, P(X|Z)$:

$$\begin{aligned}\log P(X) &= D_{KL}[Q(Z|X)||P(Z|X)] \\ &\quad + \mathbb{E}_{Z\sim Q(Z|X)}[\log P(X|Z)] - D_{KL}[Q(Z|X)||P(Z)]\end{aligned} \tag{1.1}$$

where D_{KL} is the Kullback-Leibler divergence. This equation forms the basis for the VAE objective function, and follows directly from the expression for the Kullback-Leibler divergence between $Q(Z|X)$ and $P(Z|X)$ (the true, unknown distribution for Z given X):

$$\begin{aligned}&D_{KL}[Q(Z|X)||P(Z|X)] \\ &= \mathbb{E}_{Z\sim Q(Z|X)}[\log Q(Z|X) - \log P(Z|X)] \\ &= \mathbb{E}_{Z\sim Q(Z|X)}[\log Q(Z|X) - \log(P(Z)\cdot P(X|Z)/P(X))] \\ &= \mathbb{E}_{Z\sim Q(Z|X)}[\log Q(Z|X) - \log P(Z) - \log P(X|Z)] \\ &\quad + \log P(X) \\ &= D_{KL}[Q(Z|X)||P(Z)] - \mathbb{E}_{Z\sim Q(Z|X)}[\log P(X|Z)] \\ &\quad + \log P(X)\end{aligned} \tag{1.2}$$

Note that the last line in equation (1.2) can easily be rewritten as the expression in equation (1.1).

In equation (1.1), $D_{KL}[Q(Z|X)||P(Z|X)]$ cannot be computed (as $P(Z|X)$ is unknown), but since it is always nonnegative, $\mathbb{E}_{Z\sim Q(Z|X)}[\log P(X|Z)] - D_{KL}[Q(Z|X)||P(Z)]$ is a lower bound for $\log P(X)$. VAEs are thus designed to maximize this "evidence lower bound" (ELBO) as a proxy for maximizing $P(X)$. The term $\mathbb{E}_{Z\sim Q(Z|X)}[\log P(X|Z)]$ represents the likelihood of the data given Z, *according to the conditional distribution* $Q(Z|X) = \mathcal{N}(h_\mu(X), \mathrm{diag}(h_\sigma(X)))$. The VAE thus learns functions h_μ and h_σ that produce

conditional distributions over Z, which, in turn, yield high values of $P(X|Z)$. Minimizing $D_{KL}[Q(Z|X)||P(Z)]$ ensures that the learned distribution for $Q(Z|X)$ does not stray too far from the prior distribution $P(Z) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

More intuitively, the VAE learns a neural network h that maps individual data points to mean and variance vectors for the distribution $Q(Z|X)$, in such a way that samples from $Q(Z|X) \sim \mathcal{N}(h_\mu(X), \text{diag}(h_\sigma(X)))$ can be passed to a decoder network g capable of reconstructing the original input. Maximizing the term $\mathbb{E}_{Z\sim Q(Z|X)}[\log P(X|Z)]$ is equivalent to minimizing the reconstruction loss between the data X and the output of the decoder network. Minimizing the term $D_{KL}[Q(Z|X)||P(Z)]$ regularizes the encoder network h to produce latent distributions that are relatively close to $\mathcal{N}(\mathbf{0}, \mathbf{I})$; this ensures that $Q(Z|X)$ is relatively compact and can be directly sampled from without producing erroneous decodings from g.

We modify the standard VAE with the assumption that, for sequential data of length T, data at each time point arise from identically distributed latent variables $\{Z_0, ..., Z_T\}$, and that changes between consecutive time points arise from changes in Z, i.e., for two consecutive time points t and $t+1$, $X_t \sim g(Z_t)$ and $X_{t+1} \sim g(Z_{t+1})$. While the actual function governing $Z_t \mapsto Z_{t+1}$ is unknown, we assume that it is Markovian (dependent on the current state only), and hypothesize that for any t, Z_t can be used to predict Z_{t+1}.

For any t, a standard VAE would identify regions of Z likely to have produced X_t using $Q(Z_t|X_t)$; under the assumption outlined above, we instead use $Q(Z_t|X_{t-1})$. Intuitively, we assume that we can learn some function h' which will map an observation at $t-1$ to regions of the latent space that were likely to have generated the *next* observation at t, i.e., $h' : X_{t-1} \mapsto Q(Z_t|X_{t-1})$. The decoder function g' then maps the latent distribution $Q(Z_t|X_{t-1})$ to a distribution over observations at time t, i.e., $g' : Q(Z_t|X_{t-1}) \mapsto P(X_t|Z_t)$. Note that we can easily condition the latent distribution on an action a_{t-1} taken at time $t-1$, by simply including a_{t-1} in the input to h', resulting in the latent distribution $Q(Z_t|X_{t-1}, a_{t-1})$. Thus, whereas a standard VAE encodes an input X_t into the latent distribution $Q(Z_t|X_t)$ before reconstructing X_t, our tVAE encodes a data point X_{t-1} into a latent distribution $Q(Z_t|X_{t-1})$ (or $Q(Z_t|X_{t-1}, a_{t-1})$), then attempts to construct the observation X_t. As in equation (1.2),

the expression for $D_{KL}[Q(Z_t|X_{t-1}, a_{t-1})||P(Z_t|X_t)]$ can be rewritten to show that

$$\begin{aligned}\log P(X_t) = & D_{KL}[Q(Z_t|X_{t-1}, a_{t-1})||P(Z_t|X_t)] \\ & + \mathbb{E}_{Z\sim Q(Z_t|X_{t-1},a_{t-1})}[\log P(X_t|Z_t)] \\ & - D_{KL}[Q(Z_t|X_{t-1}, a_{t-1})||P(Z_t)]\end{aligned}$$

Fig. 1.1 outlines the structure of standard VAEs versus tVAEs.

1.2.2.2 tVAE-based environment

The model takes as input the 14-length state vector for time t appended to a one-hot encoding of the action take at that time point (e.g., $[1, 0, 0, 0, 0, 0]$ for $a_t = 0$). Inputs pass through a u-unit hidden layer with sigmoid activations, then passed to a layer defining l means ($\boldsymbol{\mu}$) and l variances ($\boldsymbol{\sigma^2}$) for the latent distribution Z. An l-length vector is drawn from $\mathcal{N}(\boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma^2}))$ and passed through a u-unit decoder layer with sigmoid activations, before reaching the output layer predicting the patient state at $t + 1$. The final model has $u = 10$ hidden units (roughly 2/3 of the output feature space) and $l = 7$ latent dimensions (roughly 1/2 of output feature space), based on guidelines in [60].

The model is trained using gradient descent to minimize a weighted sum of the prediction loss (measured in mean squared error) and the KL divergence between $\mathcal{N}(\boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma^2}))$ and $\mathcal{N}(\mathbf{0}, \mathbf{I})$. In practice, equally weighting these two terms this can cause the model to under-train toward prediction loss [16]; weighting prediction loss at 80% and regularization at 20% improves training stability and model performance. We also monitor the model's loss on a 15% hold-out set to confirm that the model does not overfit.

1.2.3 Benchmark models

1.2.3.1 HMM-based environment

Our HMM assumes that for a given time $t \in \{0, ..., T\}$, each patient occupies one of m latent disease states. The vector π defines the probability distribution over state membership at $t = 0$, while the $m \times m$ transition matrix P_{a_t} defines the state transition probabilities when

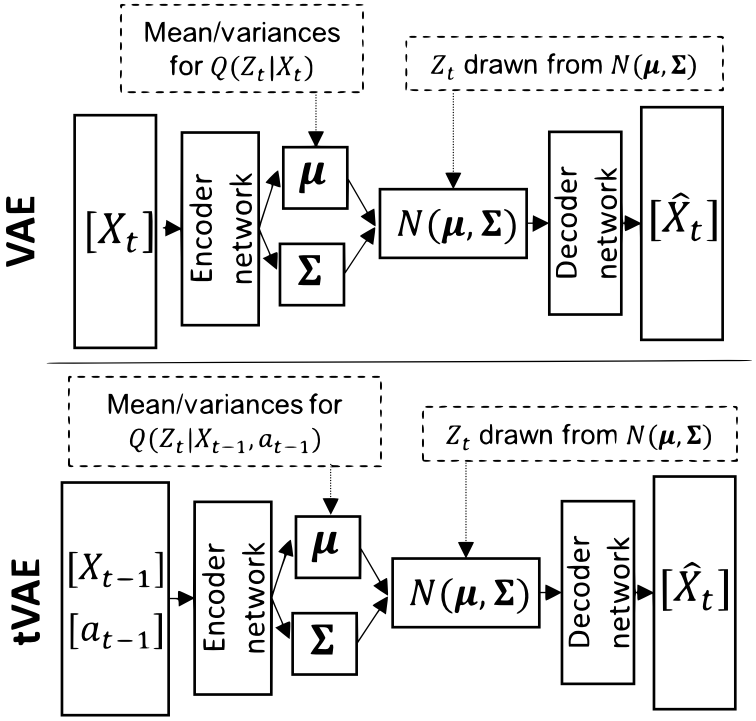

Figure 1.1: Structure of the variational autoencoder (VAE, top) and our transitional variational autoencoder (tVAE, bottom).

dosage $a_t \in \{0, 1, 2, 3, 4, 5\}$ is administered at time t. We assume a time-homogeneous HMM where the transition matrix P_{a_t} does not vary based on time t. The emission model Ψ consists of m k-dimensional Gaussian distributions $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma_i})$, where k is the dimensionality of the observation vectors and i is the index of the latent state. We assume all $\boldsymbol{\Sigma_i}$ are diagonal matrices (i.e., all feature covariances are accounted for by the latent state).

We first use the Baum-Welch algorithm [12] to estimate $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma_i}$ from patient data for $i \in \{1, ..., k\}$ with $k = 14$. We choose $m = 9$ to optimize model fit according to the Bayesian Information Criteria [126] (after removing two states with less than 0.1% visitation). We then estimate P_{a_t} for $a_t \in \{0, 1, 2, 3, 4, 5\}$ by stratifying patients' data sequences by heparin dosage groups.

1.2.3.2 LSTM-based environment

The LSTM model is a recurrent neural network that can model long-term dependencies by using 'forget gates' to store relevant sequence information. Our LSTM takes as input a full history of each patient's clinical states and heparin dosages, and for each time t outputs the patient's predicted clinical state at time $t + 1$. The LSTM's recurrent layer included $u = 10$ hidden units (roughly 2/3 times the output dimensionality [60]) with a *tanh* activation. When simulating patient sequences, a patient's state at time t is sampled according to $\mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}_t$ is the LSTM output at time t and $\boldsymbol{\Sigma}$ is a diagonal covariance matrix with entries σ_i^2 for $i \in \{1, ..., k\}$. For a clinical indicator i, σ_i^2 is defined as its conditional variance at time t given its values at time $t - 1$. As with the tVAE, we monitor the model's loss on a 15% hold-out set to confirm that the model does not overfit.

1.2.3.3 CGAN-based environment

During training, the CGAN takes as input a patient's clinical state and heparin dosage at time t, concatenated to an l-dimensional random Gaussian noise vector. A generator network with one u-unit hidden layer converts this input to a 'synthetic' clinical vector for time $t+1$. This structure incorporates random variability into the network (as with standard GANs, which take random noise vectors as input) while ensuring that generated patient states for time $t + 1$ are dependent on the state and action at time t.

The synthetic and ground-truth patient states for time $t+1$ are fed to a 'discriminator' network (with a u-unit hidden layer), which predicts the probability that the input comes from the ground truth dataset. The generator network is trained to maximize the ground truth probabilities assigned to its synthetic outputs, while the discriminator network is trained to minimize them. Both models are jointly trained until equilibrium, i.e., until the differences between their losses converges with tolerance $\epsilon = 0.001$. To make the CGAN comparable with the tVAE, we set $l = 7$ and $u = 10$, and we monitor training loss on a 15% hold-out set to confirm that the model does not overfit. When simulating patient sequences, clinical states at time t are concatenated with an l-dimensional random Gaussian noise vector and fed into the generator network to predict the patient state at time $t+1$.

1.2.3.4 CVAE-based environment

During training, the CVAE takes as input a concatenation of a patient's clinical states at times t *and* $t+1$, along with their heparin dosage at time t. A u-unit hidden layer then transforms the input to an l-dimensional latent distribution $\mathcal{N}(\boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma^2}))$. A random sample from $\mathcal{N}(\boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma^2}))$ is then concatenated with the patient state at time t, fed into a u-unit decoder layer, and used to predict the patient state at time $t+1$. The latent distributions $\mathcal{N}(\boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma^2}))$ are regularized based on their KL-divergence from a standard normal prior. As with the tVAE, we set $l = 7$ and $u = 10$, and weight prediction loss at 80% and KL-divergence loss at 20%. We monitor the model's loss on a 15% hold-out set to confirm that the model does not overfit. CVAEs simulate patient sequences in a manner identical to the CGAN: clinical states at time t are concatenated with an l-dimensional random Gaussian noise vector and fed into the decoder network to predict the patient state at time $t+1$.

1.2.4 A3C RL algorithm and model evaluation

Using the aforementioned model environments, we train RL agents to maximize rewards through a heparin dosage policy $\pi_\theta(a_t|s_t)$. The resulting stochastic optimal policy maps any patient state $s_t \in S = \mathbb{R}^{14}$ at time $t \in \{0, .., T\}$, representing the patient's 14-length clinical

feature vector, to a probability distribution over actions $a_t \in \{0, 1, 2, 3, 4, 5\}$, i.e., the six possible heparin dosages. The reward function $r(s_t, a_t)$ is taken from [93], and depends only on the aPTT value in the current state. This function is given by $r(s_t, a_t) = \frac{2}{1+e^{-(aPTT-60)}} - \frac{2}{1+e^{(aPTT-100)}} - 1$, and assigns reward of almost 1 for aPTT values within the 60–100 sec therapeutic range, and -1 for aPTT values outside this range. The RL algorithm thus seeks policy parameter vector θ that maximizes the expected discounted cumulative reward, i.e., $\arg\max_\theta \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r(s_t, \pi_\theta(s_t))\right]$, with $\gamma = 0.99$.

We trained the RL agents with the Asynchronous Advantage Actor-critic algorithm (A3C; [83]). A3C is on-policy and considered state-of-the-art, and therefore an appropriate example of a powerful reinforcement learning algorithm that can only be used when provided a fully-defined model of the patient environment. The algorithm uses neural networks to map each encountered state into an estimated reward-to-go (the "value network") and a stochastic policy over the possible actions (the "policy network", denoted $\pi_\theta(a_t|s_t)$ and parameterized by θ). In this implementation, the value and policy networks shared all layers except for the output layers. From any given state s_t, the agent evaluates a possible action a_t based on its advantage $A(s_t, a_t)$ – the difference between the action's expected reward-to-go and the state's expected reward to-go, averaged across all actions, i.e.,

$$A(s_t, a_t) = \sum_{i=0}^{k-1} \gamma^i r_{t+i} + \gamma^k V(s_{t+k}) - V(s_t) \tag{1.3}$$

where $k \leq T$ and $V(s_t)$ is the expected reward-to-go from state s_t, approximated with a neural network. The agent then updates the policy network weights (denoted with θ) based on the gradient $\nabla_\theta \log(\pi_\theta(a_t|s_t)A(s_t, a_t)) + \eta \nabla_\theta H(\pi_\theta(s_t))$, where $H(\pi_\theta(s_t))$ is the entropy of the policy distribution over all a_t from s_t. Intuitively, the policy update encourages actions that yield large advantage values, regularized with an entropy term that encourages action exploration. The algorithm is "asynchronous" because it performs these updates based on the interactions of multiple agents moving through the environment in parallel. Agents are trained for 2000 epochs, each of which simulated 500 hours of treatment (a relatively long time horizon was required to stabilize training), using environment models that are trained on the entire dataset.

We use A3C to learn one heparin dosage policy from each model (that is, using each model as the 'environment' in which the policies are trained). To assess each model's ability to train effective dosage policies, we evaluate each policy's cumulative discounted return over 5000 test runs that simulate one week of treatment (typical of ICU heparin patients [41]). Test runs are evenly split across the tVAE, HMM, LSTM, CGAN, and CVAE environments, allowing us to assess policies' robustness to different environments. For each test run, a policy's score is the difference between its cumulative return and the cumulative return in that environment under a 'no treatment' policy ($a_t = 0, \forall t$). Lastly, note that all model environments (tVAE, HMM, LSTM, CGAN, CVAE) and the A3C policy/value network are trained using tensorflow [1].

1.3 Results

In this section, we provide an overview of the patient characteristics, and compare our tVAE-based environment's simulated patient trajectories with actual patient trajectories and those of the benchmark environments. Finally, we evaluate the performance of RL policies learned from our tVAE model against those obtained under the benchmarks.

1.3.1 Patient characteristics

The sample is 42.4% female, with a median age of 70.4 years (IQR 58.3–79.8) and a median weight of 173 lbs (IQR 145–205). The median sequence length for each patient is 27 hours (IQR 19-35), and on average, patients received non-zero heparin dosages 96.0% of the time (IQR 85.7%–100%).

1.3.2 Simulated trajectory characteristics

We train each model on 85% of the patient data ($n = 1757$ patients) and use the trained models to generate 100 "alternative" trajectories for each of the remaining 15% of patients ($n = 310$). These trajectories start from the patient's first measurement vector (aPTT and additional clinical predictors), last for the same number of hours, and use the same heparin

dosage groups that the patient received. We compared the characteristics of the 310 held-out sequences with each model's simulated sequences. Figure 1.2 provides example trajectories from the tVAE and each benchmark model. Results are summarized in Table 1.1. In the following, we discuss the results in detail.

1.3.2.1 Predicted aPTT values

We examine the average aPTT value produced by each model, and compare it to patients' average aPTT of 61.12 sec. The tVAE, HMM, LSTM, CGAN, and CVAE yield average aPTT values of 61.46, 62.15, 63.99, 58.23, and 59.25 sec, respectively, with only the tVAE not significantly differing from the patient average of 61.12 sec (p-value > 0.05). We also calculate the mean absolute error (MAE) between each patient's aPTT sequence and their corresponding synthetic aPTT sequences from each model. The tVAE and CGAN and tVAE sequences most closely match the ground truth aPTT sequences, with respective MAEs of 12.15 and 11.15 (difference not significant, p-value > 0.05). The HMM, LSTM, and CVAE yield MAEs of 16.05, 16.32, and 13.40, respectively, significantly higher than the CGAN and tVAE (p-value < 0.05).

1.3.2.2 aPTT variability

For each actual and simulated sequence, we examine the absolute percentage changes in aPTT between consecutive time points as a measure of trajectory variability. Fig. 1.3 presents the distributions of absolute percentage changes in the test set and in each model's synthetic sequences. The average absolute percentage aPTT change in the test set is 11.0%. The averages for the tVAE, HMM, LSTM, CGAN, and CVAE trajectories are 10.7%, 30.9%, 30.7%, 5.0%, and 14.6%, respectively. Thus, the CGAN underestimates aPTT variability, while the HMM, LSTM, and CVAE overestimate it.

We also examine the proportion of aPTT changes that exceed 10%, since aPTT changes above 10% are considered clinically significant [114]. In the test set, 25.1% of aPTT changes exceed 10%, compared with 37.5%, 76.2%, 76.7%, 6.5%, and 53.4% for the tVAE, HMM, LSTM, CGAN, and CVAE trajectories, respectively. Thus, all models' sequences

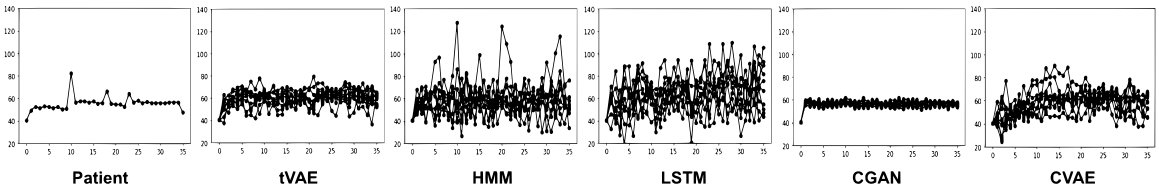

Figure 1.2: Comparison of actual and simulated aPTT trajectories for a randomly sampled patient (patient 2503). Plots show the patient's actual aPTT sequence, and ten randomly selected alternate trajectories from each environment (out of 100 total).

Table 1.1: Descriptive Statistics for real and simulated aPTT trajectories. MAE=Mean absolute error.

	Patients	tVAE	HMM	LSTM	CGAN	CVAE
Mean (sec)	61.12	61.46	62.15	63.99	58.23	59.25
MAE (sec)	-	12.15	16.05	16.32	11.15	13.40
Abs. % Change (Mean)	11.0%	10.7%	30.9%	30.7%	5.0%	14.6%
Abs. % Change (> 10%)	25.1%	37.5%	76.2%	76.7%	6.5%	53.4%
Abs. % Change (by aPTT window)	11.7% 8.3% 48.7%	13.9% 8.0% 45.4%	38.0% 22.7% 36.3%	42.1% 21.7% 27.7%	4.4% 6.2% 53.8%	17.4% 11.1% 41.6%

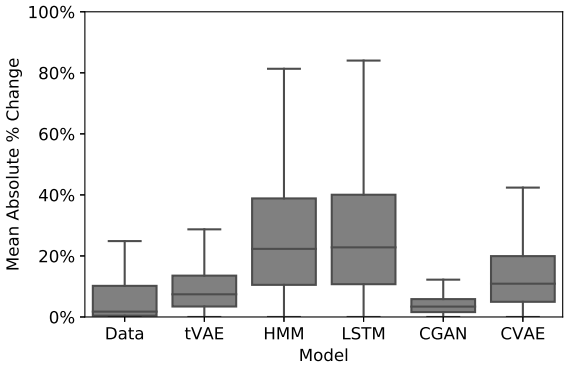

Figure 1.3: Distributions of absolute percentage changes in aPTT. Outliers not shown. Distribution means are, from left to right, 11.0%, 10.7%, 30.9%, 30.7%, 5.0%, and 14.6%. The tVAE trajectories most closely resemble patient trajectories in terms of consecutive time-step variability.

overestimate the proportion of clinically significant aPTT changes, with the tVAE doing so the least.

Lastly, we examine how aPTT variability differs based on the patient's clinical status. For the patient test data and for each model's synthetic sequences, we compute the mean absolute percentage change separately for time points in which the patient was below, within, or above the therapeutic aPTT window (60-100 sec). Results are shown in Fig. 1.4. In the test set, patients' aPTT varies more when it is above the therapeutic window. The mean absolute percentage changes for aPTT values below, within, and above the therapeutic range are 11.7%, 8.3%, and 48.7%, respectively. Trajectories from the tVAE exhibit a similar pattern, with mean percentage changes of 13.9%, 8.0%, and 45.4%, respectively. Mean percentage changes for the HMM, LSTM, CGAN, and CVAE are, respectively, 38.0%/22.7%/36.3%, 42.1%/21.7%/27.7%, 4.4%/6.2%/53.8%, and 17.4%/11.1%/41.6%.

1.3.3 Reinforcement learning performance

Using A3C, we train RL agents to learn an optimal heparin dosage policy, using each of the generative models as the 'environment' with which the RL agent interacts.

1.3.3.1 Dosage recommendations

The tVAE-, HMM-, LSTM-, CGAN-, and CVAE-trained policies each recommend a modal heparin dosage for a majority of patient states, similar to the policies learned from this dataset in [93]. The second dosage category, $a_t = 2$, is the modal action for the tVAE policy (average selection probability 94.1%), HMM policy (average selection probability 80.2%), and CVAE policy (average selection probability 97.6%). The LSTM policy's modal action is $a_t = 4$ (average selection probability 98.0%), while the CGAN's modal action is to administer no heparin ($a_t = 0$, average selection probability 96.9%).

1.3.3.2 Policy explainability

We calculate the average dosage recommendation, i.e., the expected value of actions, for each patient state in the dataset under each model's policy. For each patient state, we

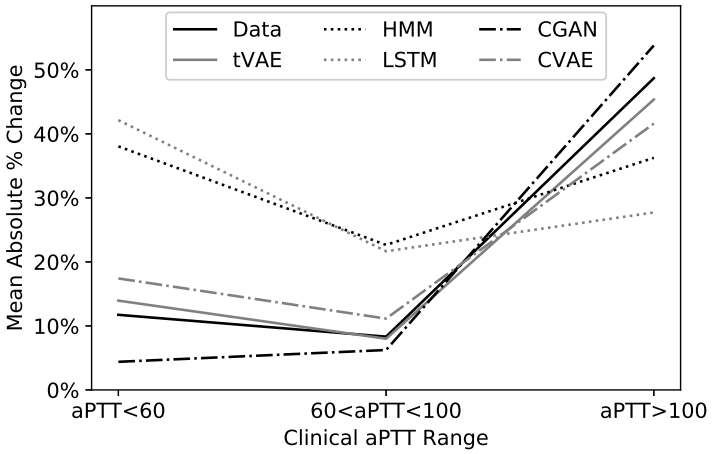

Figure 1.4: Trajectory variability by patient clinical status. Actual patient trajectories vary less when below or within the therapeutic aPTT range, and vary more when above the therapeutic range. The pattern is most accurately captured by tVAEs.

also calculate its aPTT measurement's distance from the therapeutic window of 60-100 sec; distances are negative for aPTT values below 60 sec, 0 for values between 60 and 100 sec, and positive for values above 100 sec. The correlations between average recommended heparin dosages and aPTT distances are -0.23, -0.34, -0.07, 0.35, and 0.29 for the tVAE, HMM, LSTM, CGAN, and CVAE policies, respectively. Thus, consistent with current practice [79], agents trained in a tVAE, HMM, or LSTM environments learn to prescribe higher heparin dosages for patients farther below the desired aPTT window. In contrast, agents trained in a CGAN or CVAE environment learn to prescribe higher dosages for patients closer to (or exceeding) the therapeutic window.

1.3.3.3 Policy performance

All policies were evaluated on 5000 test runs simulating one week of ICU treatment, split evenly across all five model environments. The tVAE- and CVAE-trained policies yield average scores of 11.83 and 12.79, respectively (difference not significant, p-value > 0.05). These policies significantly outperform the HMM-, LSTM-, and CGAN-trained policies, with average scores of 9.92, 7.07, and -0.59, respectively (p-value < 0.05). Thus, across all environments, the policies that tend to administer dosage category $a_t = 2$ outperform those that administer other dosages.

1.4 Discussion

Patient trajectories generated from a tVAE better reflect patients' average aPTT values and aPTT variability, compared with multiple benchmarks. The tVAE and CVAE doage policies both perform well across multiple environments, though the tVAE policy is more clinically defensible.

The HMM and LSTM overestimate patients' aPTT variability (Fig. 1.4). The HMM's number of hidden states is constrained by the need to ensure sufficient visitation to each state, requiring high emission variances to cover the observation space. In fact, the HMM's average aPTT standard deviation is 15.1 seconds, only slightly smaller than the standard deviation of the entire aPTT distribution (16.6 sec). Sampling from these high-variance distributions

yields large absolute aPTT changes between successive time points. The LSTM, being a predictive rather than a generative model, requires its emission standard deviations to be specified post-hoc. While the standard deviation of patients' aPTT changes is 0.93 (standard deviation units), 72.2% of these aPTT changes are between -0.25 and 0.25. Thus, the LSTM's emission standard deviation of 0.93 is too large for most patient trajectories - something that the LSTM is not able to correct for during training.

Our CGAN produces low-variance trajectories with no appreciable changes in patient aPTT (Fig. 1.2). The CGAN reaches training equilibrium by simply generating patient states for time $t+1$ that are similar to the patient's previous state at time t. Because of the strong dependence between consecutive clinical measurements, the discriminator network is likely unable to distinguish between these generated patient states and the true patient state at time $t+1$, thereby incentivizing the CGAN to down-weight the role of its random noise input.

Unsurprisingly, the CVAE's patients trajectories and dosage policy are somewhat similar to that of the tVAE's. Yet the tVAE's policy is more interpretable, and unlike the CVAE policy, it prescribes higher aPTT dosages for patients below the therapeutic aPTT window (60-100 sec), in line with clinical practice. The CVAE also overestimates aPTT variability. The CVAE's latent standard deviations average 0.61 during training, but, since they are conditioned on patients' previous *and* current states, must be replaced with unit variances during testing, yielding high-variance trajectories. This also explains why CVAE trajectories overestimate aPTT variability below/within the therapeutic range, and under-estimate it above this range (Fig. 1.4). While the tVAE can condition its latent variances on its inputs (and can reproduce these variances during testing), any such relationships learned during CVAE training are lost during testing when unit variances must be used.

The tVAE used here is trained on limited history, i.e., only a single time point. Preliminary tests found that tVAE performance was not improved by using longer input sequences; thus, we use the LSTM benchmark to account for the impact of long-term sequence dependence on generated trajectories. Still, further research is needed to confirm how longer input sequences might alter tVAE-generated trajectories. Further research is also needed to assess the impact of sample size on tVAE performance and the risk of

overfitting, and to more closely examine the role of patient demographics in the tVAE's synthetic trajectories. We also chose to limit our RL experiments to the A3C algorithm, which combines the strengths of multiple classes of algorithms. Still, future work will explore how results might vary based on the RL training algorithm. Lastly, we recognize that ICU patients represent a unique problem class, and additional work is needed to establish the advantages of the tVAE in other domains.

1.5 Conclusion

The novel variant of VAE, tVAE, produces realistic patient trajectories and can learn effective and clinically defensible medication dosage policies. It outperforms other state-of-the-art methods due to its relaxed distributional assumptions, continuous state space, use of a mediating latent state layer, and consistency between training and testing architectures.

Chapter 2

Hidden Markov Models as Recurrent Neural Networks: An Application to Alzheimer's Disease

2.1 Introduction

Hidden Markov models (HMMs; [12]) are commonly used for modeling disease progression, because they capture complex and noisy clinical measurements as originating from a smaller set of latent health states. Because of their intuitive parameter interpretations and flexibility, HMMs have been used to model sepsis [131], Alzheimer's disease progression [74], and patient response to blood anticoagulants [93].

Researchers may wish to use patient-level covariates to improve the fit of HMM parameter solutions [163], or to integrate HMMs directly with treatment planning algorithms [93]. Either modification requires incorporating additional parameters into the HMM, which is typically intractable with expectation-maximization algorithms. Incorporating covariates or additional treatment planning models therefore requires multiple estimation steps (e.g., [163]) changes to HMM parameter interpretation (e.g., [93]), or Bayesian estimation, which involves joint prior distributions over all parameters and can suffer from poor convergence in complex models [119].

We present neural networks as a valuable alternative for implementing and solving HMMs for disease progression modeling. Neural networks' substantial modularity allows them to easily incorporate additional input variables (e.g., patient-level covariates) or predictive models and simultaneously estimate all parameters [20]. We therefore introduce Hidden Markov Recurrent Neural Networks (HMRNNs) - neural networks that mimic the computation of hidden Markov models while allowing for substantial modularity with other predictive networks. In doing so, our primary contributions are as follows: (1) We prove how recurrent neural networks (RNNs) can be formulated to optimize the same likelihood function as HMMs, with parameters that can be interpreted as HMM parameters (section 2.3), and (2) we demonstrate the HMRNN's utility for disease progression modeling, in which combining it with other predictive neural networks improves predictive accuracy and offers unique parameter interpretations not afforded by simple HMMs (section 2.4).

2.2 Related work

A few studies in the speech recognition literature model HMMs with neural networks [151, 17]; these implementations require HMM pre-training [151] or minimize the mutual information criterion [17], and they are not commonly used outside the speech recognition domain. These works also present only theoretical justification, with no empirical comparisons with expectation-maximization algorithms.

A limited number of healthcare studies have also explored connections between neural networks and Markov models. [93] employs a recurrent neural network to approximate latent health states underlying patients' ICU measurements. [32] compares HMM and neural network effectiveness in training a robotic surgery assistant, while [11] proposes a generative neural network for modeling ICU patient health based on HMMs. These studies differ from our approach of directly formulating HMMs as neural networks, which maintains the interpretability of HMMs while allowing for joint estimation of the HMM with other predictive models.

2.3 Methods

In this section, we briefly review HMM preliminaries, formally define the HMRNN, and prove that it optimizes the same likelihood function as a corresponding HMM.

2.3.1 HMM preliminaries

Formally, an HMM models a system over a given time horizon T, where the system occupies a hidden state $x_t \in S = \{1, \ldots, k\}$ at any given time point $t \in \{0, 1, \ldots, T\}$; that is, $x_t = i$ indicates that the system is in the i-th state at time t. For any state $x_t \in S$ and any time point $t \in \{0, 1, \ldots, T\}$, the system emits an observation according to an emission distribution that is uniquely defined for each state. We consider the case of categorical emission distributions, which are commonly used in healthcare (e.g., [9, 131]). These systems emit a discrete-valued observation $y_t \in O$ at each time t, where $O = \{1, \ldots, c\}$.

Thus, an HMM is uniquely defined by a k-length initial probability vector $\boldsymbol{\pi}$, $k \times k$ transition matrix $\boldsymbol{P}$, and $k \times c$ emission matrix $\boldsymbol{\Psi}$. Entry i in the vector $\boldsymbol{\pi}$ is the probability of starting in state i, row i in the matrix $\boldsymbol{P}$ is the state transition probability distribution from state i, and row i of the matrix $\boldsymbol{\Psi}$ is the emission distribution from state i. We also define $\text{diag}(\boldsymbol{\Psi}_i)$ as a $k \times k$ diagonal matrix with the i-th column of $\boldsymbol{\Psi}$ as its entries (i.e., the probabilities of observation i from each of the k states). We define the likelihood of an observation sequence $\boldsymbol{y}$ in terms of $\alpha_t(i)$, the probability of being in state i at time t *and* having observed $\{y_0, ..., y_t\}$. We denote $\boldsymbol{\alpha}_t$ as the (row) vector of all $\alpha_t(i)$ for $i \in S$, with

$$\boldsymbol{\alpha}_t = \boldsymbol{\pi}^\top \cdot \text{diag}(\boldsymbol{\Psi}_{y_0}) \cdot (\prod_{i=1}^{t} \boldsymbol{P} \cdot \text{diag}(\boldsymbol{\Psi}_{y_i})) \tag{2.1}$$

for $t \in \{1, ..., T\}$, with $\boldsymbol{\alpha}_0 = \boldsymbol{\pi}^\top \cdot \text{diag}(\boldsymbol{\Psi}_{y_0})$. The likelihood of a sequence $\boldsymbol{y}$ is thus given by $\Pr(\boldsymbol{y}) = \boldsymbol{\alpha}_T \cdot \mathbf{1}_{k \times 1}$.

2.3.2 HMRNN definition

An HMRNN is a recurrent neural network whose parameters directly correspond to the initial state, transition, and emission probabilities of an HMM. As such, training an HMRNN optimizes the joint log-likelihood of the N T-length observation sequences given these parameters.

Definition 2.1. *An HMRNN is a recurrent neural network with parameters $\boldsymbol{\pi}$ (a k-length vector whose entries sum to 1), $\boldsymbol{P}$ (a $k \times k$ matrix whose rows sum to one), and $\boldsymbol{\Psi}$ (a $k \times c$ matrix whose rows sum to one). It receives $T+1$ input matrices of size $N \times c$, denoted by $\boldsymbol{Y}_t$ for $t \in \{0, 1, \ldots, T\}$, where the n-th row of matrix $\boldsymbol{Y}_t$ is a one-hot encoded vector of observation $y_t^{(n)}$ for sequence $n \in \{1, \ldots, N\}$. The HMRNN consists of an inner block of hidden layers that is looped $T+1$ times (for $t \in \{0, 1, \ldots, T\}$), with each loop containing hidden layers $\boldsymbol{h}_1^{(t)}$, $\boldsymbol{h}_2^{(t)}$, and $\boldsymbol{h}_3^{(t)}$, and a c-length input layer $\boldsymbol{h}_y^{(t)}$ through which the input matrix $\boldsymbol{Y}_t$ enters the model. The HMRNN has a single output unit $o^{(T)}$ whose value is the joint negative log-likelihood of the N observation sequences under an HMM with parameters $\boldsymbol{\pi}$, $\boldsymbol{P}$, and $\boldsymbol{\Psi}$; the summed value of $o^{(T)}$ across all N observation sequences is the loss function (minimized via neural network optimization, such as gradient descent).*

Layers $\boldsymbol{h}_1^{(t)}$, $\boldsymbol{h}_2^{(t)}$, $\boldsymbol{h}_3^{(t)}$, and $o^{(T)}$ are defined in the following equations. Note that the block matrix in equation (2.3) is a $c \times (kc)$ block matrix of c $\mathbf{1}_{1\times k}$ vectors, arranged diagonally, while the block matrix in equation (2.4) is a $(kc) \times k$ row-wise concatenation of c $k \times k$ identity matrices.

$$\boldsymbol{h}_1^{(t)} = \begin{cases} \boldsymbol{\pi}^\top, & t = 0, \\ \boldsymbol{h}_3^{(t-1)} \boldsymbol{P}, & t > 0. \end{cases} \tag{2.2}$$

$$\begin{aligned} \boldsymbol{h}_2^{(t)} = \mathrm{ReLu}\Big(& \boldsymbol{h}_1^{(t)} \Big[\mathrm{diag}(\boldsymbol{\Psi}_1) \ldots \mathrm{diag}(\boldsymbol{\Psi}_c) \Big] + \\ & \boldsymbol{Y}_t \begin{bmatrix} \mathbf{1}_{1\times k} & \ldots & \mathbf{0}_{1\times k} \\ \ldots & \ldots & \ldots \\ \mathbf{0}_{1\times k} & \ldots & \mathbf{1}_{1\times k} \end{bmatrix} - \mathbf{1}_{n\times(kc)} \Big) \end{aligned} \tag{2.3}$$

$$\boldsymbol{h}_3^{(t)} = \boldsymbol{h}_2^{(t)} \begin{bmatrix} \boldsymbol{I}_k & \ldots & \boldsymbol{I}_k \end{bmatrix}^\top \tag{2.4}$$

$$o^{(t)} = -\log(\boldsymbol{h}_3^{(T)} \mathbf{1}_{k\times 1}). \tag{2.5}$$

Fig. 2.1 outlines the structure of the HMRNN. Note that layer $\boldsymbol{h}_3^{(t)}$ is equivalent to $\boldsymbol{\alpha}_t$, the probability of being in each hidden state given $\{y_0, ..., y_t\}$. Also note that, for long sequences, underflow can be addressed by normalizing layer $\boldsymbol{h}_3^{(t)}$ to sum to 1 at each time point, then simply subtracting the logarithm of the normalization term (i.e., the log-sum of the activations) from the output $o^{(T)}$.

2.3.3 Proof of HMM/HMRNN equivalence

We now formally establish that the HMRNN's output unit, $o^{(T)}$, is the negative log-likelihood of an observation sequence under an HMM with parameters $\boldsymbol{\pi}$, $\boldsymbol{P}$, and $\boldsymbol{\Psi}$. We prove this for the case of $N = 1$ and drop notational dependence on n (i.e., we write $y_t^{(1)}$ as y_t), though extension to $N > 1$ is trivial since the log-likelihood of multiple independent sequences is the sum of their individual log-likelihoods. We first rely on the following lemma.

Lemma 2.2. *If all units in $\boldsymbol{h}_1^{(t)}(j)$ are between 0 and 1 (inclusive), then $\boldsymbol{h}_3^{(t)} = \boldsymbol{h}_1^{(t)} \operatorname{diag}(\boldsymbol{\Psi}_{y_t})$.*

Proof. Let $\boldsymbol{h}_1^{(t)}(j)$ and $\boldsymbol{h}_3^{(t)}(j)$ represent the jth units of layer $\boldsymbol{h}_1^{(t)}$ and $\boldsymbol{h}_3^{(t)}$, respectively, and recall that $\boldsymbol{h}_2^{(t)}$ contains $k \times c$ units, which we index with a tuple (l, m) for $l \in \{1, \ldots, c\}$ and $m \in \{1, \ldots, k\}$. According to equation (2.3), the connection between units $\boldsymbol{h}_1^{(t)}(j)$ and $\boldsymbol{h}_2^{(t)}(l, m)$ is $\boldsymbol{\Psi}_{j,l}$ when $j = m$, and 0 otherwise. Also recall that matrix $\boldsymbol{Y}_t$ enters the model through a c-length input layer that we denote $\boldsymbol{h}_y^{(t)}$. According to equation (2.4), the connection between unit $\boldsymbol{h}_y^{(t)}(j)$ and unit $\boldsymbol{h}_2^{(t)}(l, m)$ is 1 when $j = l$, and 0 otherwise. Thus, unit $\boldsymbol{h}_2^{(t)}(l, m)$ depends only on $\boldsymbol{\Psi}_{m,l}$, $\boldsymbol{h}_1^{(t)}(m)$, and $\boldsymbol{h}_y^{(t)}(l)$. Lastly, a bias of -1 is added to all units in $\boldsymbol{h}_2^{(t)}$, which is then subject to a ReLu activation, resulting in the following expression for each unit in $\boldsymbol{h}_2^{(t)}$:

$$\boldsymbol{h}_2^{(t)}(l, m) = \text{ReLu}(\boldsymbol{\Psi}_{m,l} \cdot \boldsymbol{h}_1^{(t)}(m) + \boldsymbol{h}_y^{(t)}(l) - 1). \tag{2.6}$$

Because $\boldsymbol{h}_y^{(t)}(l)$ is 1 when $y_t = l$, and equals 0 otherwise, then if all units in $\boldsymbol{h}_1^{(t)}$ are between 0 and 1, this implies $\boldsymbol{h}_2^{(t)}(l, m) = \boldsymbol{\Psi}_{m,l} \cdot \boldsymbol{h}_1^{(t)}(m)$ when $j = y_t$ and $\boldsymbol{h}_2^{(t)}(l, m) = 0$ otherwise.

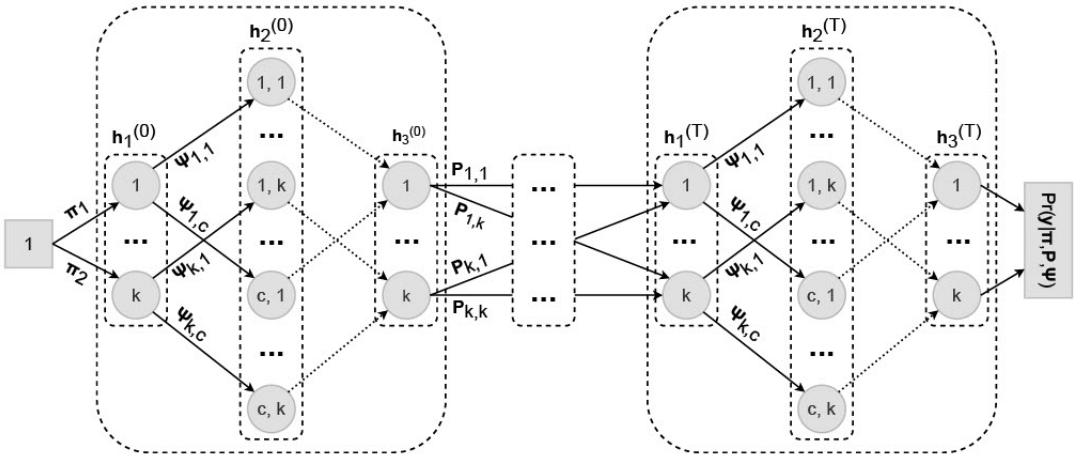

Figure 2.1: Structure of the hidden Markov recurrent neural network (HMRNN). Solid lines indicate learned weights that correspond to HMM parameters; dotted lines indicate weights fixed to 1. The inner block initializes with the initial state probabilities then mimics multiplication by $\text{diag}(\boldsymbol{\Psi}_{y_t})$; connections between blocks mimic multiplication by $\boldsymbol{P}$.

According to equation (2.5), the connection between $\boldsymbol{h}_2^{(t)}(l,m)$ and $\boldsymbol{h}_3^{(t)}(j)$ is 1 if $j=m$, and 0 otherwise. Hence,

$$\boldsymbol{h}_3^{(t)}(j) = \sum_{j=0}^{c} \boldsymbol{h}_2^{(t)}(l,j) = \boldsymbol{\Psi}_{j,y_t} \cdot \boldsymbol{h}_1^{(t)}(j). \tag{2.7}$$

Thus, $\boldsymbol{h}_3^{(t)} = \boldsymbol{h}_1^{(t)} \operatorname{diag}(\boldsymbol{\Psi}_{y_t})$. □

Theorem 2.3. *An HMRNN with parameters $\boldsymbol{\pi}$ ($1 \times k$ stochastic vector), $\boldsymbol{P}$ ($k \times k$ stochastic matrix), and $\boldsymbol{\Psi}$ ($k \times c$ stochastic matrix), and with layers defined as in equations (2.2-2.5), produces output neuron $o^{(T)}$ whose value is the negative log-likelihood of a corresponding HMM.*

Proof. Note that, based on Lemma 2.2 and equation (2.2), $\boldsymbol{h}_3^{(t)} = \boldsymbol{h}_3^{(t-1)} \cdot \boldsymbol{P} \cdot \operatorname{diag}(\boldsymbol{\Psi}_{y_t})$ for $t \in \{1, ..., T\}$, assuming that $\boldsymbol{h}_1^{(t)}(j) \in [0,1]$ for $j \in \{1, .., k\}$. Since $\boldsymbol{\alpha}_t = \boldsymbol{\alpha}_{t-1} \cdot \boldsymbol{P} \cdot \operatorname{diag}(\boldsymbol{\Psi}_{y_t})$, then if $\boldsymbol{h}_3^{(t-1)} = \boldsymbol{\alpha}_{t-1}$, then $\boldsymbol{h}_1^{(t)}(j) \in [0,1]$ for $j \in \{1, .., k\}$ and therefore $\boldsymbol{h}_3^{(t)} = \boldsymbol{\alpha}_t$. We show the initial condition that $\boldsymbol{h}_3^{(0)} = \boldsymbol{\alpha}_0$, since $\boldsymbol{h}_1^{(0)} = \boldsymbol{\pi}^\top$ implies that $\boldsymbol{h}_3^{(0)} = \boldsymbol{\pi}^\top \cdot \operatorname{diag}(\boldsymbol{\Psi}_{y_0}) = \boldsymbol{\alpha}_0$. Therefore, by induction, $\boldsymbol{h}_3^{(T)} = \boldsymbol{\alpha}_T$, and $o^{(T)} = -\log(\boldsymbol{\alpha}_T \cdot \mathbf{1}_{k\times 1})$, which is the logarithm of the HMM likelihood based on equation (2.1). □

2.4 Experiment and results

We demonstrate how combining an HMRNN with other predictive neural networks improves predictive accuracy and offers novel clinical interpretations over a standard HMM, using an Alzheimer's disease case study. We test our HMRNN on clinical data from $n = 426$ patients with mild cognitive impairment (MCI), collected over the course of three ($n = 91$), four ($n = 106$), or five ($n = 229$) consecutive annual clinical visits [6]. Given MCI patients' heightened risk of Alzheimer's, modeling their symptom progression is of considerable clinical interest. We analyze patients' overall cognitive functioning based on the Mini Mental Status Exam (MMSE; [37]).

MMSE scores range from 0 to 30, with score categories for 'no cognitive impairment' (scores of 27-30), 'borderline cognitive impairment' (24-26), and 'mild cognitive impairment' (17-23) [84]. Scores below 17 were infrequent (1.2%) and were treated as scores of 17 for

analysis. We use a 3-state latent space $S = \{0, 1, 2\}$, with $x_t = 0$ representing 'no cognitive impairment,' $x_t = 1$ representing 'borderline cognitive impairment,' and $x_t = 2$ representing 'mild cognitive impairment.' The observation space is $O = \{0, 1, 2\}$, using $y_t = 0$ for scores of $27 - 30$, $y_t = 1$ for scores of $24 - 26$, and $y_t = 2$ for scores of $17 - 23$. This HMM therefore allows for the possibility of measurement error, i.e., that patients' observed score category y_t may not correspond to their true diagnostic classification x_t.

To showcase the benefits of the HMRNN's modularity, we augment it with two predictive neural networks. First, we predict patient-specific initial state probabilities based on gender, age, degree of temporal lobe atrophy, and amyloid-beta 42 levels (Aβ42, a relevant Alzheimer's biomarker [15]), using a single-layer neural network with a softmax activation. Second, at each time point, the probability of being in the most impaired state, $\boldsymbol{h}_t^{(1)}(2)$, is used to predict concurrent scores on the Clinical Dementia Rating (CDR, [86]), a global assessment of dementia severity, allowing another relevant clinical metric to inform paramter estimation. We use a single connection and sigmoid activation to predict patients' probability of receiving a CDR score above 0.5 (corresponding to 'mild dementia'). The HMRNN is trained via gradient descent to minimize $o^{(T)}$ from equation (2.5), plus the predicted negative log-likelihoods of patients' CDR scores. Figure 2.2 visualizes the structure of this augmented HMRNN.

We compare the HMRNN to a standard HMM without these neural network augmentations, trained using Baum-Welch, an expectation-maximization algorithm [12]. We assess parameter solutions' ability to predict patients' final MMSE score categories from their initial score categories, using 10-fold cross-validation. We evaluate performance using weighted log-loss L, i.e., the average log-probability placed on each final MMSE score category. This metric accounts for class imbalance and rewards models' confidence in their predictions, an important component of medical decision support [19]. We also report $\bar{p}$, the average probability placed on patients' final MMSE scores (computed directly from L). We train all models using a relative log-likelihood tolerance of 0.001%. Runtimes for Baum-Welch and the HMRNN are 2.89 seconds and 15.24 seconds, respectively.

Model results appear in Table 2.1. Note that the HMRNN's weighted log-loss L is significantly lower than Baum-Welch's (paired t-test p-value $= 2.396 \times 10^{-6}$), implying

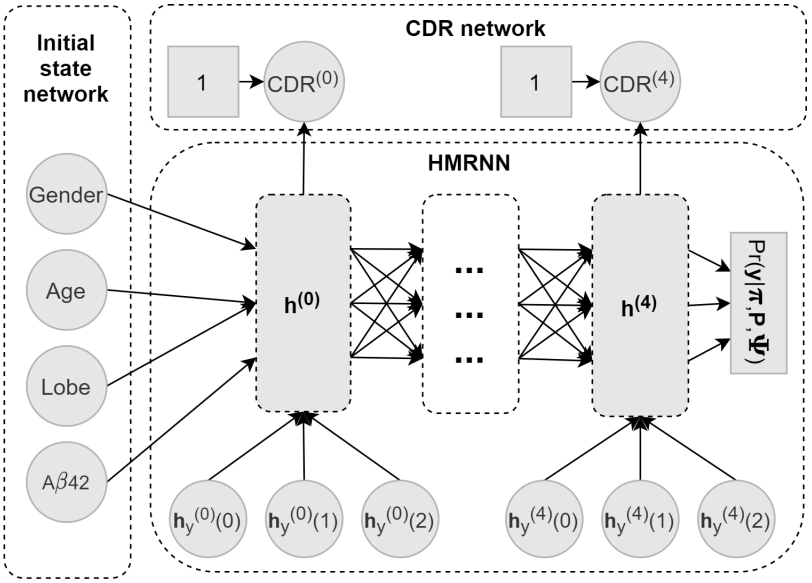

Figure 2.2: Augmented HMRNN for Alzheimer's case study. $\text{CDR}^{(t)}$ refers to predicted CDR classification (above or below 0.5) at time $t \in \{0, 1, 2, 3, 4\}$. 'Lobe' refers to measure of temporal lobe atrophy. Units $\boldsymbol{h}_y^{(t)}$ are a one-hot encoded representation of the MMSE score category at time t.

Table 2.1: Results from Alzheimer's disease case study. $\boldsymbol{\pi}$ is initial state distribution, $\boldsymbol{P}$ is state transition matrix, $\boldsymbol{\Psi}$ is emission distribution matrix, L is weighted log-loss, and $\bar{p}$ is average probability placed on ground truth score categories.

	Baum-Welch			HMRNN		
$\boldsymbol{\pi}$	0.727	0.271	0.002	0.667	0.333	0.000
$\boldsymbol{P}$	0.898	0.080	0.022	0.970	0.028	0.002
	0.059	0.630	0.311	0.006	0.667	0.327
	0.000	0.016	0.984	0.000	0.003	0.997
$\boldsymbol{\Psi}$	0.939	0.060	0.001	0.930	0.067	0.003
	0.175	0.819	0.006	0.449	0.548	0.003
	0.004	0.160	0.836	0.005	0.308	0.687
L	-0.992			-0.884		
$\bar{p}$	0.371			0.413		

greater predictive performance. The HMRNN also yields lower transition probabilities and lower estimated diagnostic accuracy for the MMSE (i.e., lower diagonal values of $\mathbf{\Psi}$) than Baum-Welch, suggesting that score changes are more likely attributable to testing error as opposed to true state changes.

2.5 Discussion

The HMRNN can be combined with other neural networks to improve predictive accuracy in disease progression applications when additional patient data is available. In our experiment, augmenting an HMRNN with two predictive networks improves forecasting performance compared with a standard HMM trained with Baum-Welch. The HMRNN also yields a clinically distinct parameter interpretation, predicting poor diagnostic accuracy for the MMSE's 'borderline' and 'mild' impairment categories. This suggests that fewer diagnostic categories might improve MMSE utility, which aligns with existing research [84] and suggests the HMRNN might be used to improve the clinical utility of HMM parameter solutions. We also make a novel theoretical contribution by formulating discrete-observation HMMs as a special case of RNNs and proving coincidence of their likelihood functions.

Future work might formally assess HMRNN time complexity. Yet since data sequences in healthcare are often shorter than in other domains that employ HMMs (e.g., speech analysis), runtimes will likely be reasonable for many healthcare datasets. Future work might explore the HMRNN in other healthcare applications besides disease progression.

Chapter 3

Adapting Reinforcement Learning Treatment Policies Using Limited Data to Personalize Critical Care

3.1 Introduction

Due to the severity of conditions treated in critical care settings (e.g., intensive care units), patients often require constant monitoring and frequent intervention to prevent rapid deterioration. This makes critical care a particularly data-intensive component of the modern healthcare system. Critical care settings must also treat patients as efficiently as possible to avoid resource strain, which can lead to increased patient mortality [153, 38]. Given these problem characteristics, data-driven methods show considerable promise for optimizing patient care in data-rich critical care settings.

One such approach is reinforcement learning (RL), which has been extensively applied to learn personalized treatment policies for critical care settings [109, 111, 145, 93, 11]. RL outputs a *treatment policy*, which is a function mapping a patient's current state to an optimal treatment action (e.g., "increase medication dosage"). Historical patient data can be used to directly learn the treatment policy [109, 111, 145], or can be used to learn a realistic model of how patient symptoms change over time (which we refer to as a *transition*

model). Transition models, in turn, can be used to learn the treatment policy [93, 11, 163]. Transition models allows for the use of 'on-policy' RL algorithms, which optimize treatment policies through repeated interactions with the transition model and have been shown to outperform other RL approaches [83].

It is crucial that RL-based treatment policies take into account the full range of patients' clinical characteristics. Applying a single treatment policy to all patients can yield poor outcomes for those whose clinical characteristics differ from the 'average' patient [67]. Indeed, recent research has begun to focus on identifying how treatment efficacy varies between different patient clusters and phenotypes [100, 3, 36].

Yet past studies generally assume that sufficient data are available from all patient groups of interest. This assumption may not be tenable for certain patient subpopulations, especially those from underrepresented demographic groups or those with rare comorbidities. Therefore, to make recent advances in RL more equitable, there is a timely need for RL techniques capable of identifying how treatment policies should be adapted for specific patient subpopulations, even if only limited data are available from those subpopulations.

In this study, we develop an approach for identifying effective treatment policies for a target patient subpopulation, even if that group is severely underrepresented in the training data. The proposed approach allows for accurate prediction of the generalizability of the identified treatment policies to this underrepresented patient population. We showcase the utility of this approach in an ICU treatment application, specifically in the administration of heparin, a blood anticoagulant. Since the potency of heparin depends on patient body weight, we define weight-based groups that differentially respond to heparin administration. We then test whether a *single patient's historical data* from the underrepresented patient subpopulation can be used to (1) identify successful treatment policies and (2) predict these policies' performance for other unseen members of this group. Specifically, we propose *noisy Bayesian policy updates (NBPU)*, which leverages variational inference in learning and updating policies. That is, NBPU first *learns a distribution* over candidate dosage policies from a 'reference' patient group (i.e., the subpopulation for which sufficient data are available). It then updates each policy's probability to generate *noisy estimates* of each policy's efficacy in an 'underrepresented' patient group (i.e., the subpopulation for which

little data are available), using a *single patient's historical data* from this 'underrepresented' patient group. Our Bayesian updating formulation also provides an opportunity to leverage this noisy estimate of a policy's efficacy to accurately predict the policy's performance when applied to other members of the underrepresented patient population.

We benchmark our approach against a wide array of state-of-the-art methods and current practice. Specifically, we use an RL policy learned from the reference patient group, transfer learning (a common technique for adapting machine learning models to new environments), a clustering approach, and a hidden parameter Markov decision process (HiP-MDP) model. In addition, because our approach involves variational policy learning *and* Bayesian updating of each policy, we also compare it to variational policy learning *without* updates to demonstrate the value of such Bayesian updating (per limited information from a single underrepresented patient's historical data). Finally, we use a standard clinical treatment protocol to benchmark our model against current practice.

3.1.1 Reinforcement Learning Applications in Treatment Planning

There is a considerable body of research on using RL to optimize treatment selection. These studies generally use patients' historical data to identify the optimal patient-specific or group-specific treatment policies. The goal is generally to maximize patient 'reward' over a given time horizon, where the reward might constitute the probability of survival, expected quality adjusted life years (QALYs), or the utility associated with certain health states.

One class of RL problems is multi-armed bandits (MABs). MABs involve an agent selecting from a set of available actions, each with an unknown distribution over rewards, so as to maximize the expected cumulative reward over some time horizon. MABs thus seek to balance exploitation of high-reward actions with exploration of actions whose reward distributions are not well-understood. In healthcare, 'actions' might refer to available therapies or medications, and the agent seeks to learn optimal mappings between patient characteristics and treatment or medication recommendations. For instance, [92] use multi-armed bandits to identify optimal treatments for multiple sclerosis patients, [163] use them

to optimize drug selection for myeloma patients, and [77] use them to recommend emotion regulation strategies for patients with social anxiety.

Another class of RL problems are Markov decision processes (MDPs). In MDPs, patients typically transition between health states according to their current health state and the treatment action taken. Consequently, treatment actions affect a patient's future health state, which may in turn impact available actions in the future. RL algorithms for solving MDPs maximize some measure of reward over a given time horizon, taking into account all of the patient's possible symptom trajectories. Some studies develop MDP models to optimize outpatient care. For instance, [148] use clinical literature to build MDPs that simulate Parkinson's symptoms and employ RL to prescribe optimal medication dosages and timing. [106] use RL to regulate neurostimulation implants for epilepsy patients, and [28] use RL to regulate insulin pumps for diabetes patients. Other studies focus on inpatient critical care settings. [109] and [158] use RL to learn optimal strategies for weaning patients off mechanical ventilators, while most other studies focus on medication and intravenous fluid administration. For instance, [111, 66] and [150] use RL to optimize intravenous fluid regimens for sepsis patients, while [93] and [11] use RL to optimize blood anticoagulant administration for ICU patients. Note that some studies utilize 'off-policy' RL algorithms, which learn treatment policies directly from historical patient data (e.g., [106, 109, 111, 93]). Others utilize 'on-policy' RL algorithms, which learn treatment policies by simulating patient symptoms under an MDP model (e.g., [148, 28, 158, 11]).

In summary, RL is an increasingly relied-upon method for deriving optimal treatment policies from historical patient data. Most studies to date leverage high-frequency patient data, obtained either through bedside monitoring [109, 158, 111, 66, 150, 93, 11] or wearable/implanted sensors [28, 106], to model patients' symptoms as MDPs. This framework yields treatment policies that take future patient health states into account, maximizing long-term outcomes over the course of the patient's treatment.

3.1.2 Adapting Treatment Policies to Specific Subpopulations

Not all patients are guaranteed to respond to treatment in the same way. Hence, treatment policies need to be learned in a way that accounts for variability across patients and maximizes patient-specific rewards. There exists a limited number of techniques that attempt to adapt treatment policies to specific patient groups with unique disease progression dynamics.

Transfer learning, which is commonly used in medical imaging and diagnosis [23, 121, 58], uses model parameters from one domain to initialize model training in another domain. While often used for classification tasks, it can be extended to RL problems as well [55, 133, 112]. RL algorithms can use policy parameters learned from well-understood environments (e.g., for 'reference' patient groups) to initialize policy training in new environments (e.g., for 'underrepresented' patient groups).

Algorithms can also organize patients into clinically homogeneous clusters, each with their own unique treatment policy. For instance, [100] use kernel regression to identify promising drug therapies for HIV patients that conform to well-defined patient clusters. Similarly, [142] and [31] use cluster-based approaches to identify optimal interventions for separate patient groups, first by identifying patient clusters and then by learning separate policies for each cluster-specific transition model.

Hidden parameter Markov decision processes (HiP-MDPs; [8, 156, 157, 159]) assume the existence of multiple transition models that are characterized by a latent parameter. This latent parameter determines the unique characteristics of each transition model. A 'master policy' conditioned on this latent parameter can then be used to transfer the policy to new environments, provided that the new environment's latent parameter is correctly estimated.

Lastly, variational policy learning is a promising technique for learning *multiple* near-optimal RL policies for the same environment. The most common variational policy learning algorithm is Stein variational policy gradient (SVPG; [75]), which learns a probability distribution over RL policy parameters rather than a single policy solution. Some of the policies learned through variational policy learning might transfer to new environments more effectively than other policies; therefore, variational policy learning might be used to learn effective policies for novel or underrepresented patient populations. However, to

date variational policy learning has primarily been used to improve RL performance within a *single* environment, and has not been extended to the medical domain. For instance, [49] use SVPG to learn optimal stochastic policies for continuous state spaces, while [26] use an adaptation of SVPG to improve state space exploration in video game tasks.

3.1.3 Limitations of Existing Methods and Proposed Method's Contributions

The existing techniques for adapting treatment policies to specific patient subpopulations are generally ill-suited to problems where the subpopulation of interest is underrepresented in the training data. For instance, transfer learning agents must still be fully trained in the environment of interest (e.g., with underrepresented patient groups), which may be unstable without sufficient data. Similarly, cluster-based approaches [142, 100] and HiP-MDPs [8, 156, 157, 159] both assume that the testing environment (e.g., specific patient subpopulation of interest) is already well-represented in training data. This assumption is not necessarily tenable in cases where only limited data are available for an underrepresented patient group. While variational policy learning can yield multiple 'candidate' policies, some of which might perform well for underrepresented patient groups, it provides no mechanism for identifying such policies.

As such, new methods need to be developed for adapting treatment policies to underrepresented patients using only a limited amount of training data. To address this, we propose *noisy Bayesian policy updates (NBPU)*, which selects treatment policies for underrepresented patient groups for whom only little data (specifically, only a single patient's historical data) are available. Our approach leverages variational policy learning [75] to first learn a probability distribution over *multiple* candidate RL policies. We augment variational policy learning with a Bayesian updating step, in which patient-specific treatment response models [142, 94] are used to update the probabilities associated with each RL policy. This updating step identifies the RL policies that are most likely to perform well for underrepresented patient groups. In our case, the patient-specific transition models are fitted using a *single underrepresented patient's historical data*, to demonstrate our approach's

ability to successfully select treatment policies for underrepresented groups based on very small sample sizes. NBPU can be used with any treatment policy structure, can leverage observational patient data (in contrast with [156]), and is specifically designed to work with limited data from underrepresented patient subpopulations (in contrast with [8, 142]). We demonstrate our method's effectiveness in the critical care setting, using a dataset of ICU patients receiving blood anticoagulation therapy.

In addition to *policy learning*, there is also a need for methods that can accurately predict *how a policy will perform* when applied to a new patient group [14, 45, 116]. We show that NBPU's Bayesian updating formulation is equivalent to estimating each policy's performance for underrepresented patients as a linear combination of (1) its performance in reference patients' transition model, and (2) its performance in a 'noisy' transition model learned from a single underrepresented patient. We then show that this approach outperforms other benchmarks in predicting its policies' performances with underrepresented patients.

Our primary contribution is thus developing the first approach for selecting critical care treatment policies for underrepresented patient subpopulations, when only a single patient's historical data from the group are available. In doing so, we offer a novel application of variational RL policy learning to the treatment planning domain, and demonstrate how extending variational RL policy learning with patient-specific treatment response models can improve performance and prediction for underrepresented patient groups. While we develop and demonstrate our approach for the medical treatment domain, this approach may also apply to other RL domains in which only limited data from a novel testing environment is available.

3.2 Methods

In this section, we provide our data, approach, and evaluation scheme. In Section 4.2.1, we introduce our dataset. Next, in Section 3.2.2, we discuss the distinct patient subpopulations in the data (which we refer to as the 'reference' and 'underrepresented' patient groups), and outline our distinct transition models for the two patient groups. Next, in Section 3.2.3, we formalize our treatment planning problem, which involves learning k near-optimal candidate

treatment policies for the reference patient group. These policies are sampled according to a 'policy probability distribution,' where the higher the probability, the better the expected performance of the corresponding policy in the reference patient group. In Section 3.2.4, we introduce our proposed approach for using a single underrepresented patient's historical data to update the policies' probability values to include information about their predicted performance in the underrepresented patient group. These updated policy probabilities are then used to select policies for the underrepresented patient group and to predict these policies' performance in practice. Next, in Section 3.2.5, we introduce our benchmarks. Finally, in Section 3.2.6, we describe our testing and evaluation procedure. Figure 3.1 provides an overview of our method.

3.2.1 Data

We base our analysis on patient data from the MIMIC dataset [57], a publicly available database of clinical records. Specifically, we use hourly clinical measurements from 2,020 ICU patients who receive intravenous heparin administration (a blood anticoagulant) and have at least six consecutive hourly measurements. Patient sequences include hourly heparin dosages (in units/mL) and activated partial thromboplastin time (aPTT) measurements; aPTT is a measure of blood coagulation that increases in response to heparin, and it is typically used to assess the efficacy of heparin treatment. The desirable therapeutic window for aPTT is defined as 60-80 seconds [136]. Data also includes 13 additional clinical variables for each patient: arterial carbon dioxide (CO_2), heart rate (HR), creatinine, Glasgow Coma Score (GCS), hematocrit, hemoglobin, international normalized ratio of prothrombin (INR), platelet count, prothrombintime, arterial oxygen saturation (SAO_2), temperature, urea, and white blood cell count (WBC).

As in [93], we use sample-and-hold interpolation to impute missing heparin doses for each patient. We also use a neural network to impute missing aPTT values, which was trained on each patient's demographic information, heparin dosage, and their 13 additional clinical variables.

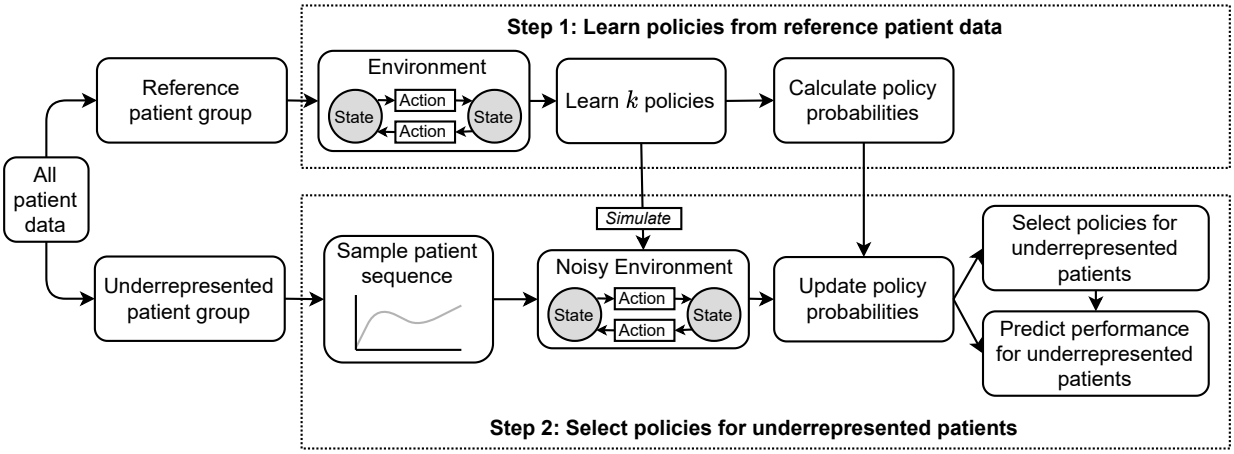

Figure 3.1: Overview of proposed noisy Bayesian policy updates (NBPU). In the first step, k candidate policies are learned from reference patient data. In the second step, a single underrepresented patient's data is used to learn a 'noisy' transition model, which is then used to estimate the candidate policies' performances for underrepresented patients. This information is used to update the probability associated with each policy, to select high-probability policies for the underrepresented patients, and to predict these policies' efficacy when evaluated on underrepresented patients.

3.2.2 Characterizing Subpopulation-Specific Transition Models

In training and evaluating heparin dosage policies, we use an on-policy RL approach [30, 46] which requires a transition model of how patients' aPTT responds to heparin. An RL agent interacts directly with this model to learn an optimal heparin dosage policy through trial and error [11], using state-of-the-art RL algorithms [83, 55].

We do *not* assume that all patients are well-described by the same underlying transition model. Instead, we assume there exists a *reference* transition model T that adequately describes the heparin response of most ICU patients. We also assume that there exists an *underrepresented* transition model T', which describes the heparin response of a small group of individuals for whom sufficient data are not available. In other contexts, T' might represent the treatment response of an underrepresented demographic group, patients with a rare comorbidity, or patients with a nontypical response to a particular treatment.

We model T and T' as autoregressive models, predicting patients' aPTT changes in response to heparin based on available data. Autoregressive models are frequently used in disease progression modeling [80, 62, 56], and their simple structure reduces the risk of overfitting to small datasets [160]. We first discuss the autoregressive models fit to the data, and then describe how we use this modeling technique to identify the patient subpopulations that correspond to T and T'.

3.2.2.1 Autoregressive Transition Models.

We normalize aPTT and heparin between -1 and 1, and then perform first-order differencing on patients' aPTT and heparin sequences (a common time series technique for reducing nonstationarity). Thus, for a patient with N hourly measurements, the aPTT sequence $\{y^{(1)}, \ldots, y^{(N)}\}$ and heparin dosage sequence $\{h^{(1)}, \ldots, h^{(N)}\}$ are converted to $\{\Delta y^{(2)}, \ldots, \Delta y^{(N)}\}$ and $\{\Delta h^{(2)}, \ldots, \Delta h^{(N)}\}$, where $\Delta y^{(t)} = y^{(t)} - y^{(t-1)}$ and $\Delta h^{(t)} = h^{(t)} - h^{(t-1)}$ for $t \in \{2, \ldots, N\}$. We then use an autoregressive model to predict $\Delta y^{(t)}$ from up to five hours of heparin dosage history (including the most recent dosage), i.e., $\Delta h^{(t-j)}$ for $j \in \{0, \ldots, l_h\}$, $0 \leq l_h \leq 4$, and up to five hours of prior aPTT measurements, i.e., $\Delta y^{((t-1)-j)}$ for $j \in \{0, \ldots, l_y\}$, $0 \leq l_y \leq 4$. The final autoregressive model predicting

aPTT change $\Delta\hat{y}^{(t)}$ at time t is given by

$$\Delta\hat{y}^{(t)} = \sum_{j=0}^{l_h} b_j^h \Delta h^{(t-j)} + \sum_{j=0}^{l_y} b_j^y \Delta y^{((t-1)-j)} + \epsilon, \tag{3.1}$$

where ϵ is a normally-distributed error term, b_j^h is the coefficient for heparin dosage change at time $t-j$, and b_j^y is the coefficient for aPTT change at time $(t-1)-j$. Note that per our preliminary results, after accounting for heparin and aPTT history, patients' demographic covariates (age, gender) and additional clinical variables did not meaningfully improve model fit (inclusion increases R^2 by only 0.001 and decreased model's Bayesian Information Criteria). Thus, we model patient aPTT as a function of patients' aPTT and heparin administration histories only.

3.2.2.2 Identifying the Reference and Underrepresented Patient Groups.

Since heparin is a weight-based drug, we identify weight-based groups with clinically distinct responses to heparin administration. Specifically, consistent with the literature, we assume that patients above a body weight quantile cutoff q respond to heparin differently than patients below that quantile cutoff [10, 52]. We denote the autoregressive treatment response model of the larger patient group as T, and the transition model of the smaller patient group as T'. That is, we treat the larger patient group as the 'reference group' for whom sufficient data are available, and the smaller patient group as the 'underrepresented group' for whom clinical data may be lacking. The models T and T' are then directly learned from clinical data of patients who fall on either side of a select body weight quantile.

To identify the optimal quantile cutoff q that best characterizes reference and underrepresented groups, we divide patients into 20 subsamples based on their body weight quantile, where subsample $m \in \{1, \ldots, 20\}$ contains patients whose body weight falls between the $0.05(m-1)$ and $0.05(m)$ quantiles. For each subsample m, we learn a separate autoregressive model (per Equation (3.1)) with regression coefficients $b_j^{y(m)}$ and $b_j^{h(m)}$, and define the heparin potency for each subsample as $\sum_{j=0}^{4} b_j^{h(m)}$. We then calculate the difference in average heparin potency below and above each possible cutoff $c \in \{1, \ldots, 20\}$, and choose the cutoff c^* that maximizes this difference, as follows:

$$c^* = \arg\max_c \left(\frac{1}{c} \sum_{m=1}^{c} \left[\sum_{j=0}^{4} b_j^{h(m)} \right] - \frac{1}{20-c} \sum_{m=c+1}^{20} \left[\sum_{j=0}^{4} b_j^{h(m)} \right] \right) \tag{3.2}$$

We then calculate the optimal body weight quantile $q \in \{0.05, 0.10, \ldots, 0.95\}$ that corresponds to the resulting cutoff, i.e., $q = 0.05c^*$.

3.2.3 Reinforcement Learning Problem

Our treatment planning problem seeks an optimal heparin dosage policy that maximizes the amount of time patients spend with aPTT in the therapeutic range of 60-80 seconds. We define this problem for the reference and underrepresented patient groups, respectively, by the tuples $< S, A, T, r >$ and $< S, A, T', r >$, where

- S is the set of patients' health states. The health state at time t is given by $s_t \in S$;
- A is the set of possible treatment actions. The treatment action at time t is given by $a_t \in A$;
- T and T' are the respective autoregressive transition models, which describe the patient groups' clinical progressions under treatment actions taken;
- $r : S \mapsto \mathbb{R}^1$ is the reward function, which specifies the utility associated with a given health state.

We define the health state as $s_t = \left(y^{(t-1)}, \ldots, y^{((t-1)-(l^*+1))}, h^{(t)}, \ldots, h^{(t-(l^*+1))}\right)$, where $l^* = \max\{l_y, l_h\}$, which is sufficient to predict patients' next aPTT measurement under transition models T and T'. We define the action at time t as $a_t = \Delta h^{(t)}$, i.e., the heparin dosage change to be administered at time t. The reward function $r(s_t)$ at each time point is given by

$$r(s_t) = \frac{2}{1 + e^{-\left(y^{(t)} - 60/150\right)}} - \frac{2}{1 + e^{-\left(y^{(t)} - 80/150\right)}} - 1. \tag{3.3}$$

This reward function, which is also used in [93], assigns a reward of (approximately) 1 when a patient's unstandardized aPTT value is between 60 and 80 seconds (a desirable therapeutic range; [136]), and a reward of (approximately) -1 otherwise.

A heparin dosage policy $\pi_\theta : S \mapsto A$ is a function (with parameter vector θ) that maps a patient's state to an optimal hourly heparin dosage, with the goal of maximizing the patient's expected cumulative reward $J(\theta) = \mathbb{E}_{(\pi_\theta)}\left[\sum_{t=0}^{t_{\max}} r(s_t)\right]$ over t_{max} time points (where the expectation is taken over the states s_t that result from following policy π_θ). More intuitively, the policy personalizes a patient's hourly heparin dosage based on their recent aPTT and heparin history, with the goal of maximizing the amount of time the patient spends in the therapeutic aPTT range of 60-80 seconds. Note that, following [93], we do not apply a discount factor to r_t, given that many patients receive heparin therapy over a relatively brief time horizon (less than one week; [76]).

We model the policy π_θ as a single-layer neural network, with parameter vector θ, that takes as input the patient state vector s_t and outputs an optimal heparin dosage change $\Delta h(t) \in [-1, 1]$ (via an output node with a $tanh()$ activation). During training, $\Delta h^{(t)}$ is treated as the mean of a Gaussian distribution to encourage exploration, with the network also outputting a standard deviation which is discarded during testing.

Lastly, note that a heparin dosage policy's expected cumulative reward (i.e., the policy's 'performance') will vary depending on whether it is used in the reference patient group (i.e., under model T) or the underrepresented patient group (i.e., under model T'). Thus, we use $J_T(\theta)$ and $J_{T'}(\theta)$ to refer to a policy's performance under transition models T and T', respectively.

3.2.4 Noisy Bayesian Policy Updates: Learning Optimal Policies from Limited Data

We address the problem of learning an optimal heparin dosage policy for an underrepresented patient group, *using just one underrepresented patient's historical data*, along with data from a reference patient group. That is, we assume that a clinical decision maker has sufficient data to estimate T (the treatment response model for a reference patient group), but only a single patient's data generated from T' (the treatment response model for the underrepresented patient group). Noisy Bayesian Policy Updates (NBPU) accomplishes this through a two-step process, i.e., it first learns a distribution of candidate heparin dosage policies from

transition model T, and then uses a single patient record from T' to 'noisily' estimate the policies' effectiveness in T'.

For the first step, we use Stein Variational Policy Gradient (SVPG; [75]) to learn a probability distribution over k candidate policies $\{\pi_{\theta_1}, \ldots, \pi_{\theta_k}\}$ from *reference* patient data (i.e., under model T). SVPG is a variational policy learning technique that, rather than learning a *single* optimal policy, seeks a *probability distribution* over policy parameter vectors, which we denote by P_T for the reference patient model T. This probability distribution is the solution to the maximization

$$\max_{P_T} \mathbb{E}_{\theta \sim P_T}[J_T(\theta)] - \alpha_T \, \mathbb{D}(P_T || P_0), \tag{3.4}$$

where $\mathbb{D}(\cdot)$ is Kullback-Leibler divergence, P_0 is a prior distribution for P_T, and α_T is a hyper-parameter. More intuitively, P_T maximizes the expected performance $J_T(\theta)$ (i.e., expected performance for reference patients) of the policies defined by its policy parameter vectors, subject to regularization to a prior distribution. The parameter α_T governs the strength of the prior regularization, with P_T collapsing to a Dirac delta distribution at the best-performing policy parameter vector as $\alpha_T \mapsto 0$, and approaching the prior P_0 as $\alpha_T \mapsto \infty$.

The SVPG algorithm learns k policy parameter vectors $\{\theta_1, \ldots, \theta_k\}$, corresponding to policies $\{\pi_{\theta_1}, \ldots, \pi_{\theta_k}\}$, such that $\{\theta_1, \ldots, \theta_k\}$ are sampled according to P_T. After training, the probability density value for a given policy parameter vector θ_i, i.e., $P_T(\theta_i)$, is given by

$$P_T(\theta_i) \propto \exp\left(\frac{1}{\alpha_T} J_T(\theta_i)\right) P_0(\theta_i), \tag{3.5}$$

where $P_0(\theta_i)$ is the prior probability placed on θ_i. The prior P_0 is typically set to be uniform to encourage exploration of the entire policy space [75], in which case SVPG reduces to entropy regularization of each policy's performance. As in [75], the kernel bandwidth used for SVPG updates is $\frac{\tilde{\mu}^2}{\log(k+1)}$, where $\tilde{\mu}$ is the median pairwise distance between policy parameter vectors; this allows the bandwidth to adapt to changing distances between policies.

The second step of NBPU is a novel extension of SVPG that seeks to estimate $P_{T'}(\theta_i)$ for $i \in \{1, \ldots, k\}$, which, if $J_{T'}(\theta_i)$ were known, would be given by

$$P_{T'}(\theta_i) \propto \exp\left(\frac{1}{\alpha_{T'}} J_{T'}(\theta_i)\right) P_0(\theta_i). \tag{3.6}$$

However, because $J_{T'}(\theta_i)$, i.e., the performance of policy i in the underrepresented patient model T', is not known, NBPU calculates an estimated probability distribution $\hat{P}_{T'}$. This distribution uses P_T as a prior and updates it with limited information (i.e., a single patient's historical data) from an underrepresented patient group. That is, given a single underrepresented patient's historical data, we learn $\hat{T}'$, a noisy estimate of the underrepresented group's treatment response model (in our case, an autoregressive model learned from a single training example). We then calculate $\hat{P}_{T'}(\theta_i)$, an estimate of $P_{T'}(\theta_i)$, for $i \in \{1, \ldots, k\}$, such that

$$\hat{P}_{T'}(\theta_i) \propto \exp\left(\frac{1}{\alpha_{T'}} J_{\hat{T}'}(\theta_i)\right) P_T(\theta_i), \tag{3.7}$$

where $J_{\hat{T}'}(\theta_i)$ is the performance of policy π_{θ_i} in $\hat{T}'$, which is a noisy estimate (learned from a single patient's historical data) of the underrepresented patient model T'. The policy that produces the highest probability (i.e., $\{\pi_{\theta_{i^*}} : i^* = \arg\max_i \hat{P}_{T'}(\theta_i)\}$) is then selected for the underrepresented patient group.

Lastly, NBPU can be used to predict a given policy's expected cumulative reward when applied to the underrepresented patient group. This can be seen by substituting Equation (3.5) into Equation (3.7) and rearranging the terms as follows:

$$\begin{aligned}
\hat{P}_{T'}(\theta_i) &\propto \exp\left(\frac{1}{\alpha_{T'}} J_{\hat{T}'}(\theta_i)\right) \exp\left(\frac{1}{\alpha_T} J_T(\theta_i)\right) P_0(\theta_i) \\
&\propto \exp\left(\frac{1}{\alpha_{T'}} J_{\hat{T}'}(\theta_i) + \frac{1}{\alpha_T} J_T(\theta_i)\right) P_0(\theta_i) \\
&\propto \exp\left(\frac{1}{\alpha'}((\beta) J_{\hat{T}'}(\theta_i) + (1-\beta) J_T(\theta_i))\right) P_0(\theta_i),
\end{aligned} \tag{3.8}$$

where $\alpha' = (\alpha_{T'}^{-1} + \alpha_T^{-1})^{-1}$ and $\beta = \alpha'/\alpha_{T'} = \alpha_T/(\alpha_T + \alpha_{T'})$. Note that the last line of Equation (3.8) resembles Equation (3.6), with $\alpha_{T'}$ replaced with α' and $J_{T'}(\theta_i)$ replaced

with $(\beta)J_{\hat{T}'}(\theta_i) + (1-\beta)J_T(\theta_i)$. Thus, NBPU estimates $J_{T'}(\theta_i)$ (policy π_{θ_i}'s performance for underrepresented patients, which is unknown) with the weighted average $(\beta)J_{\hat{T}'}(\theta_i) + (1-\beta)J_T(\theta_i)$. We refer to this as NBPU's *performance prediction*, which can be used to predict a given policy's performance for underrepresented patients before it is used for treatment.

Intuitively, NBPU is designed to leverage limited information from T', while incorporating the information from T for regularization. Note that the transition model learned from the single patient, $\hat{T}'$, is expected to overfit to this patient's data. Thus, it is only a noisy estimate of T'. As such, the performance of given policy π_{θ_i} for $i \in \{1, \ldots, k\}$ in $\hat{T}'$ (i.e., $J_{\hat{T}'}(\theta_i)$) is only a noisy estimate of its true performance $J_{T'}(\theta_i)$. We thus regularize information from $\hat{T}'$ with information from the more well-known patient model T (through the policy distribution P_T). Pseudocode for NBPU is presented in Algorithm 1.

3.2.5 Benchmarks

In this section, we summarize each of the benchmark methods against which we compare NBPU results. We describe how each benchmark learns a treatment policy for the underrepresented patient group and how it predicts that policy's performance when implemented for that group (i.e., its *performance prediction*). Table 3.1 summarizes all benchmark methods along with NBPU.

Stein variational policy gradient (SVPG) serves as the first step of NBPU and is therefore included as a benchmark. SVPG selects policy $\pi_{\theta_{\tilde{i}}}$, where $\tilde{i} = \arg\max_i P_T(\theta_i)$. That is, SVPG selects the policy that performs best in the *reference* patient group, without incorporating limited data from underrepresented patients. SVPG's performance prediction is simply $J_T(\theta_{\tilde{i}})$, i.e., the best policy performance in the reference transition model. That is, SVPG assumes that policies will perform similarly for underrepresented patients as for reference patients.

We use *standard reinforcement learning* (standard RL) to train a *single* policy $\pi_{\theta_{RL}}$ in the reference patient transition model T. Standard RL's performance prediction for underrepresented patients is simply $J_T(\theta_{RL})$, i.e., the policy's performance in the reference patient transition model.

Algorithm 1: Noisy Bayesian Policy Updates

Input: Reference patient transition model T, noisy transition model $\hat{T}'$ (learned from a single underrepresented patient's historical data), reward function r, size of policy set k, number of RL training epochs e, number of iterations v, number of timesteps t_{max}, regularization parameters α_T, $\alpha_{T'}$

for *i in* $\{1, \ldots, e\}$ **do**
  Iterate Stein Variational Policy Gradient (SVPG) in T;
end
Return policy parameter vectors $\{\theta_1, \ldots, \theta_k\}$
for *i in* $\{1, \ldots, k\}$ **do**
  Select policy parameter vector θ_i;
  for *j in* $\{1, \ldots, v\}$ **do**
    Execute π_{θ_i} in T for t_{max} time steps; $R_T^j \leftarrow \sum_{t=0}^{t_{\max}} r(s_t)$
    Execute π_{θ_i} in $\hat{T}'$ for t_{max} time steps; $R_{\hat{T}'}^j \leftarrow \sum_{t=0}^{t_{\max}} r(s_t)$
  end
  Return $J_T(\theta_i) \leftarrow \frac{1}{v} \sum_{j=1}^{v} R_T^j$
  Return $J_{\hat{T}'}(\theta_i) \leftarrow \frac{1}{v} \sum_{j=1}^{v} R_{\hat{T}'}^j$
end
for *i in* $\{1, \ldots, k\}$ **do**
  Calculate $P_T(\theta_i) \leftarrow \exp\left(\frac{1}{\alpha_T} J_T(\theta_i)\right) P_0(\theta_i)$
end
Normalize $P_T(\theta_i)$ for $i \in \{1, \ldots, k\}$ to sum to 1;
for *i in* $\{1, \ldots, k\}$ **do**
  Calculate $\hat{P}_{T'}(\theta_i) \leftarrow \exp\left(\frac{1}{\alpha_{T'}} J_{\hat{T}'}(\theta_i)\right) P_T(\theta_i)$
end
Normalize $\hat{P}_{T'}(\theta_i)$ for $i \in \{1, \ldots, k\}$ to sum to 1;
Output: Policy parameter vectors θ_i and policy probabilities $\hat{P}_{T'}(\theta_i)$ for $i \in \{1, \ldots, k\}$

Table 3.1: Summary of policy selection and performance predictions for NBPU and all benchmarks.

Method	**Description**	**Policy**	**Performance Prediction**
NBPU	Learns policies $\{\pi_{\theta_1}, \ldots, \pi_{\theta_k}\}$ with probabilities $\hat{P}_{T'}(\theta_i)$, using reference patients (T) and one underrepresented patient ($\hat{T}'$). Selects $\pi_{\theta_{i^*}}$ where $i^* = \arg\max_i \hat{P}_{T'}(\theta_i)$.	$\pi_{\theta_{i^*}}$	$(\beta)J_{\hat{T}'}(\theta_{i^*}) + (1-\beta)J_T(\theta_{i^*})$
SVPG	Learns policies $\{\pi_{\theta_1}, \ldots, \pi_{\theta_k}\}$ with probabilities $P_T(\theta_i)$, using reference patients (T). Selects $\pi_{\theta_{\tilde{i}}}$ where $\tilde{i} = \arg\max_i P_T(\theta_i)$.	$\pi_{\theta_{\tilde{i}}}$	$J_T(\theta_{\tilde{i}})$
Standard RL	Learns one policy $\pi_{\theta_{RL}}$ using reference patients (T).	$\pi_{\theta_{RL}}$	$J_T(\theta_{RL})$
Transfer Learning	Learns policy $\pi_{\theta_{TL}}$ using one underrepresented patient ($\hat{T}'$). Initializes θ_{TL} with standard RL policy parameters θ_{RL}.	$\pi_{\theta_{TL}}$	$J_{\hat{T}'}(\theta_{TL})$
Clustering	Clusters reference patients. Learns transition model T_g from reference patients in each cluster and learns policy π_{θ_g} in each T_g. Samples one underrepresented patient and picks $\pi_{\theta_{g^*}}$ for cluster g^* that best fits patient.	$\pi_{\theta_{g^*}}$	$J_{T_{g^*}}(\theta_{g^*})$
HiP-MDP	Learns body weight-dependent transition model T_w from reference patients, and uses T_w to learn body weight-dependent policy π_{θ_w}. Samples one underrepresented patient's body weight w to impute into π_{θ_w}.	π_{θ_w}	$J_{T_w}(\theta_w)$
Clinical	Standard weight-based dosage policy $\pi_{\theta_{CL}}$, taken from clinical literature.	$\pi_{\theta_{CL}}$	$J_T(\theta_{CL})$

Transfer learning is often used in the medical literature [23, 121, 58] and is therefore included as a benchmark. Similar to standard RL, transfer learning first trains a single treatment policy $\pi_{\theta_{RL}}$ in the reference patient transition model T. It then samples a single underrepresented patient and learns a noisy transition model $\hat{T}'$ from their historical data. Finally, it trains a policy $\pi_{\theta_{TL}}$ in $\hat{T}'$, with the parameter vector θ_{TL} initialized with θ_{RL} (the policy learned in the reference patient transition model T). This allows transfer learning to leverage the treatment policy from the reference patient group and adjust it using a single underrepresented patient's historical data. Transfer learning's performance prediction is $J_{\hat{T}'}(\theta_{TL})$, i.e., the performance of policy $\pi_{\theta_{TL}}$ in the noisy underrepresented patient transition model $\hat{T}'$ in which it is trained.

Clustering has been used in recent research on personalized treatment planning [100, 142, 31] and is therefore included as a benchmark. This approach separates reference patients into clusters, trains separate treatment policies for each, then uses single underrepresented patients' data to select which cluster-specific policy to use for underrepresented patients. Specifically, we calculate the percentage of time each reference patient's prescribed heparin dosage fell into one of five quantile bins (0%, 1-25%, 26-50%, 51-75%, 76-100%) and use k-means to cluster the reference patients based on these dosage distributions, gender age, body weight, and percentage of time spent in the aPTT therapeutic window (60-80 seconds). We then choose the final number of clusters based on a scree plot of within-cluster sums of squares. We learn a separate transition model T_g (Equation (3.1)) for each cluster g, and learn a cluster-specific policy π_{θ_g}. We then sample a single underrepresented patient and identify the cluster g^* that best fits the patient's historical data, and select the cluster policy $\pi_{\theta_{g^*}}$ for the underrepresented patient group. The clustering performance prediction is $J_{T_{g^*}}(\theta_{g^*})$, i.e., the performance of the best-fitting cluster's policy in its own cluster-specific transition model.

Hidden parameter Markov decision processes (HiP-MDP; [8, 156, 157, 159]) learn a transition model that depends on a patient-level parameter that captures individual differences between patients. We define this patient-level parameter as patient body weight w, since this is the clinical characteristic separating the reference and underrepresented patient groups. We use reference patient data to learn a weight-dependent transition model

T_w, which is identical to Equation (3.1) but with patient body weight added as a main effect and interaction effect for all predictors. We use T_w to learn a treatment policy π_{θ_w} whose state space includes patient body weight w. We then sample an underrepresented patient's body weight and impute it into π_{θ_w}, which is then evaluated in the underrepresented patient model T'. HiP-MDP's performance prediction is $J_{T_w}(\theta_w)$, i.e., the HiP-MDP policy's performance in the weight-dependent transition model T_w in which it is trained.

Lastly, we benchmark against a *clinical baseline*, i.e., a weight-based heparin dosage policy [79] that prescribes dosages in units/kg of body weight, starting with an initial bolus and adjusting based on the patient's distance from the therapeutic aPTT window (in this case, 60-80 seconds). Note that initial bolus data are not available in the MIMIC dataset [42]; thus, we simulate weight-based heparin dosages prescribed after the patient's initial bolus. We denote this standard clinical policy by $\pi_{\theta_{CL}}$, and its performance prediction, $J_T(\theta_{CL})$, is the policy's performance in the reference patient transition model T.

3.2.6 Testing Procedure

All treatment policies are trained over 1000 epochs using policy gradient (with the exception of the clinical baseline policy, which is defined based on medical literature). Note that, consistent with past work on variational policy learning [75], we set $\alpha_T = \alpha_{T'} = 10$ for SVPG and NBPU. This corresponds to setting $\beta = 0.5$ for NBPU's performance prediction. We evaluate treatment policies by their expected cumulative reward under the underrepresented patient transition model T', averaged over 5000 iterations. Each iteration simulates one week of ICU stay ($t_{\max} = 168$ hours), which is typical of heparin ICU patients [76].

Since NBPU and most benchmarks (transfer learning, clustering, HiP-MDP, clinical baseline) depend on underrepresented patients' historical data, they may be sensitive to the particular patient being sampled. Thus, for these methods, we repeat our testing procedure across 100 trials to ensure our findings remain consistent, regardless of the specific underrepresented patient selected (i.e., we test our methods across 100 different $\hat{T}'$ models). We use a different underrepresented patient's historical data in each trial, where each data sequence is generated from T' (Equation (3.1)) using actual underrepresented patients' heparin regimens. Note that we only use data from underrepresented patients with

at least $t_{max} = 26$ hours of data (the median sequence length in the dataset). We select 100 sequences whose log-likelihoods according to T fall below the 5% quantile. This ensures that the underrepresented patient trajectories used to compute $\hat{T}'$ in each trial possess features that distinguish them from reference patient trajectories.

3.3 Results

3.3.1 Patient Characteristics

Table 3.2 presents descriptive statistics for our patient cohort. Patients skew male with a median age of 70.5 and a median body weight of 79.5 kg. Median sequence length is 26 hours of heparin administration. Patients' median aPTT is 58.6 seconds, slightly below the desirable therapeutic range of 60-80 seconds.

3.3.2 Transition Models

In this section, we describe the transition models used for all analyses. Specifically, in Section 3.3.2.1, we describe our procedure for splitting the data into weight-based subpopulations (i.e., the reference and underrepresented patient groups) and report the final transition models T and T'. In Section 4.3.2, we present model validation results, justifying our use of separate transition models for the reference and underrepresented groups. Finally, in Section 3.3.2.3, we describe the transition models T_g and T_w used for the clustering and HiP-MDP benchmarks.

3.3.2.1 Reference and Underrepresented Transition Models T and T'.

As described in Section 3.2.2.2, we split the patient data into 20 equally-sized subsamples by body weight quantiles and learn a separate autoregressive transition model for each one per Equation (3.1). Based on Equation (3.2), the optimal body weight quantile cutoff, q, is obtained as $q = 0.85$. Figure 3.2 summarizes the effect of heparin by body weight percentile and presents this quantile cutoff. As seen in the figure, heparin is least potent for patients above the 85th body weight percentile. Thus, patients below the 85th body weight percentile

Table 3.2: Descriptive statistics for 2,020 patients used for analysis.

Variable	Value
% Female	42.3%
Median Age	70.5 (IQR 58.4-79.9)
Median Body Weight (kg)	79.5 (IQR 66.0-93.5)
Median Treatment Length (hrs)	26 (IQR 19-35)
Median aPTT (sec)	58.6 (IQR 55.9-66.5)
Median CO_2 (mmHG)	24 (IQR 22-27)
Median HR (bpm)	82 (IQR 71-94)
Median Creatinine (mg/dL)	1.1 (IQR 0.8-1.6)
Median GCS (range 0-15)	15 (IQR 11-15)
Median Hematocrit (%)	31.4% (IQR 28.7%-34.9%)
Median Hemoglobin (g/dL)	10.6 (IQR 9.5-11.8)
Median INR	1.3 (IQR 1.2-1.5)
Median Platelet Count (billions/L)	212 (IQR 160-279)
Median Prothrombin Time (sec)	14.2 (IQR 13.4-15.6)
Median SAO_2	97.0 (IQR 96.0-99.0)
Median Temperature (°F)	98.4 (IQR 97.7-99.0)
Median Urea (mg/dL)	23.0 (IQR 15.0-40.0)
Median WBC (billions/L)	11.1 (IQR 8.4-14.6)

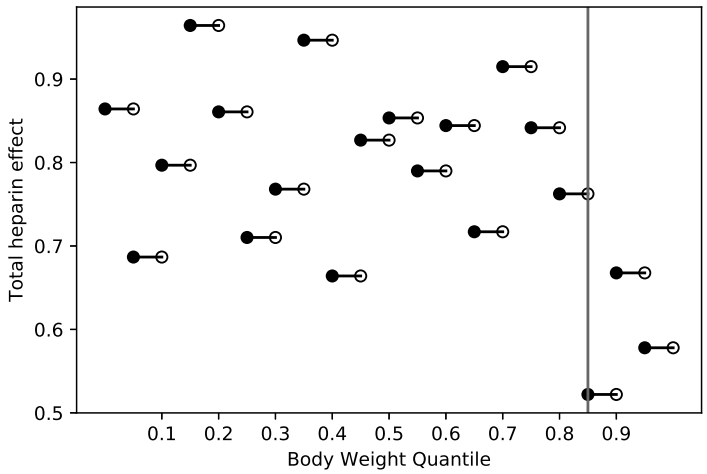

Figure 3.2: Effect of heparin by body weight percentile. Vertical line indicates the optimal quantile cutoff of $q = 0.85$. This cutoff maximizes the average heparin effect between the reference patient model T (patients below the 85% quantile) and underrepresented model T' (patients above the 85% quantile).

are used for the reference transition model T, and those above this cutoff are used for the underrepresented transition model T'.

We perform a grid search over aPTT and heparin lags $l_y \in \{0, 1, 2, 3, 4\}$ and $l_h \in \{0, 1, 2, 3, 4\}$. For each l_y and l_h, we learn separate transition models for reference and underrepresented patients and compute the models' mean absolute error (MAE) on a 20% holdout set. We select heparin and aPTT lags $l_y = 0$ and $l_h = 1$, which yield the lowest average MAE across the reference and underrepresented patient transition models. Thus, the final transition models predict the current change in aPTT ($\Delta\hat{y}^{(t)}$) from the most recent aPTT change ($\Delta y^{(t-1)}$) and the two most recent heparin dosage changes ($\Delta h^{(t)}$ and $\Delta h^{(t-1)}$). The transition models T and T' are given respectively by:

$$\Delta\hat{y}^{(t)} = 0.230\Delta h^{(t)} + 0.137\Delta h^{(t-1)} - 0.498\Delta y^{(t-1)} + \epsilon. \tag{3.9}$$

$$\Delta\hat{y}^{(t)} = 0.166\Delta h^{(t)} + 0.109\Delta h^{(t-1)} - 0.498\Delta y^{(t-1)} + \epsilon. \tag{3.10}$$

Thus, according to the parameters in T', underrepresented (high-weight) patients are less sensitive to heparin than reference patients under T. Note that we use the same aPTT coefficient b_0^y in both models, since this coefficient does not significantly vary between the reference and underrepresented patient groups (p-value> 0.05). In addition, because the residual standard deviations do not significantly differ between groups (p-value> 0.05), both models are assigned a disturbance of $\epsilon \sim \mathcal{N}(0, \lambda\sigma)$, where σ is the group-level residual standard deviation (0.087) and λ adjusts σ so that the standard deviation of simulated sequences matches the standard deviation of patient sequences (0.0746). We test $\lambda \in \{0.1, 0.25, 0.5, 1, 2\}$ and select $\lambda = 0.5$, which yields simulated sequences with standard deviation 0.0754.

3.3.2.2 Model Validation.

Across all patients, transition models T and T' predict aPTT changes within 5.4 seconds, that is, within 8.9% of the average aPTT value. Moreover, prediction error is less than 10% of the average aPTT value for 76.5% of patients. Given that aPTT differences of less than 10% are not considered clinically significant [114], this suggests that the models explain patients' aPTT changes within a clinically acceptable degree of accuracy.

The use of a separate transition model T' for underrepresented (high-weight) patients is justified by the fact that the reference transition model, T, overestimates underrepresented (high-weight) patients' observed cumulative reward. High-weight patients spend 6.0 hours in the therapeutic aPTT range on average. Simulations from T using high-weight patients' actual heparin dosages predict an average of 8.5 hours, while simulations from T' using the same data predict an average of 7.9 hours. Thus, the prediction from T' is closer to the true patient outcome. While T' also overestimates the amount of time high-weight patients spend in the therapeutic range, this is mainly due to overestimation at the high end of the distribution. For instance, the bottom 70% of high-weight patients (by cumulative reward) spend an average of 0.79 hours in the therapeutic range. Compare this with 0.81 hours as predicted by T' (2.5% error) and 1.37 hours as predicted by T (73.4% error).

Figure 3.3 presents examples of simulated aPTT trajectories for a high-weight patient under T and T'. Simulated trajectories use the patients' starting aPTT value and physician-administered heparin dosage policy. As seen in the figure, compared with T', the trajectories simulated based on T generally overestimate aPTT changes in response to a given heparin dosage.

3.3.2.3 Transition Models T_g and T_w for Clustering and HiP-MDP Benchmarks.

For the clustering benchmark, a scree plot analysis suggests an eight-cluster solution offers the best balance between parsimony and solution quality; the eight-cluster solution yields a 23.5% reduction in within-cluster sum of squares compared with a seven-cluster solution and only an 8.3% increase compared with a nine-cluster solution. We thus learn eight cluster-specific autoregressive transition models T_g for $g \in \{1, \ldots, 8\}$ from reference patient data, and learn a separate RL treatment policy π_{θ_g} for each. Note that we use the same lags $l_y = 0$ and $l_h = 1$ as in Equations (3.9) and (3.10) when calculating cluster-specific transition models.

For the HiP-MDP benchmark, our autoregressive transition model T_w uses the same lags $l_y = 0$ and $l_h = 1$ as in Equations (3.9) and (3.10), along with patient body weight w (normalized between 0 and 1) as a main effect and interaction. This yields the following

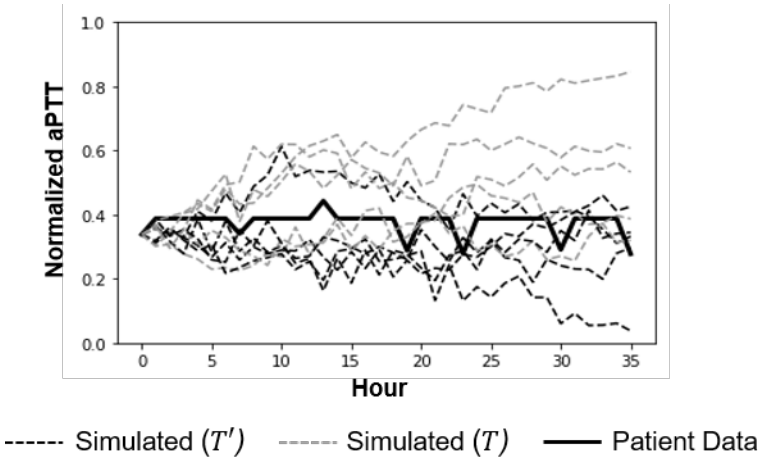

Figure 3.3: Example of aPTT trajectories for an underrepresented (high-weight) patient, under T (reference transition model) and T' (underrepresented/high-weight transition model). Note that T overestimates aPTT values compared with T'.

autoregressive transition model T_w:

$$\Delta\hat{y}^{(t)} = (0.202 + 0.101w)\Delta h^{(t)} + (0.136 + 0.0075)\Delta h^{(t-1)} + (-0.512 + 0.0813w)\Delta y^{(t-1)} + 0.0037w + \epsilon.$$

We consequently learn a body weight-dependent policy π_{θ_w} using T_w.

3.3.3 NBPU Performance and Comparison with Benchmarks

In this section, we assess the performance of NBPU-selected policies when tested on the underrepresented (high-weight) transition model T' over one week of ICU stay. We also assess NBPU's performance predictions for its policies, i.e., we assess its predictions for each policy's expected cumulative reward when tested on underrepresented patients.

Table 3.3 presents the main experimental results, and Figures 3.4 and 3.5 separately visualize each method's policy performances and the mean absolute error (MAE) of their performance predictions. As seen in Table 3.3 and Figure 3.4, NBPU-selected policies significantly outperform the standard RL policy, transfer learning, clustering, HiP-MDP, and the clinical baseline (all p-values< 0.01). Also, as seen in Table 3.3, NBPU results in relatively small interquartile range, compared with other benchmarks, across the 100 trials (i.e., 100 different $\hat{T}'$ models). This suggests that NBPU-selected policies perform rather consistently, regardless of which underrepresented patient is selected and used to develop $\hat{T}'$. Similarly, as seen in Table 3.3 and Figure 3.5, NBPU's performance predictions (i.e., predictions of its policies' performance with underrepresented patients) significantly outperform SVPG, the standard RL policy, transfer learning, clustering, and HiP-MDP (all p-values< 0.01).

3.3.3.1 Comparison with SVPG.

Policies selected by NBPU and SVPG yield similar expected cumulative returns (57.21 and 56.84, respectively, difference not significant) when tested in the underrepresented (high-weight) patient model T'. Yet NBPU more accurately predicts its policies' performances

Table 3.3: Results for NBPU and all benchmarks. 'Policy Performance in T'' refers to expected cumulative reward in underrepresented patient transition model, and 'Performance Prediction Error' refers to absolute error between predicted and actual policy performances in T'. 'IQR' refers to interquartile range from sampling 100 different underrepresented patient sequences (not defined for SVPG and standard RL, which do not utilize underrepresented patient sequences). 'NBPU Diff' refers to mean difference from NBPU, with bolded values significant at the 0.05 level based on bootstrapping.

Method	Policy Performance in T' ($J_{T'}(\theta)$)			Performance Prediction Error		
	Mean	IQR	NBPU Diff	Mean	IQR	NBPU Diff
NBPU	57.21	56.84, 57.66	-	8.79	4.13, 11.78	-
SVPG	56.84	-	-0.38	13.77	-	**4.97**
Standard RL	52.62	-	**-4.59**	11.64	-	**2.84**
Transfer Learning	46.57	35.48, 57.60	**-10.65**	15.07	5.37, 19.49	**6.27**
Clustering	53.10	51.23, 56.04	**-4.11**	10.59	5.04, 22.21	**1.80**
HiP-MDP	54.57	53.15, 57.28	**-2.65**	10.52	9.18, 11.87	**1.72**
Clinical	39.5	38.95, 39.91	**-17.76**	1.74	0.68, 2.21	**-7.73**

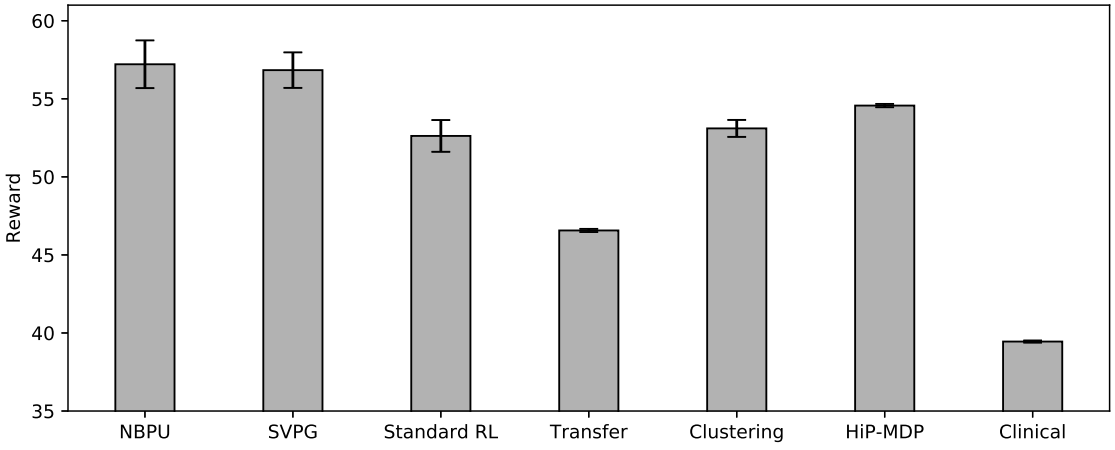

Figure 3.4: Mean policy performance in transition model T' (i.e., expected cumulative reward, $J_{T'}(\theta)$) for NBPU and all benchmark methods. Error bars represent 95% confidence intervals based on bootstrapping.

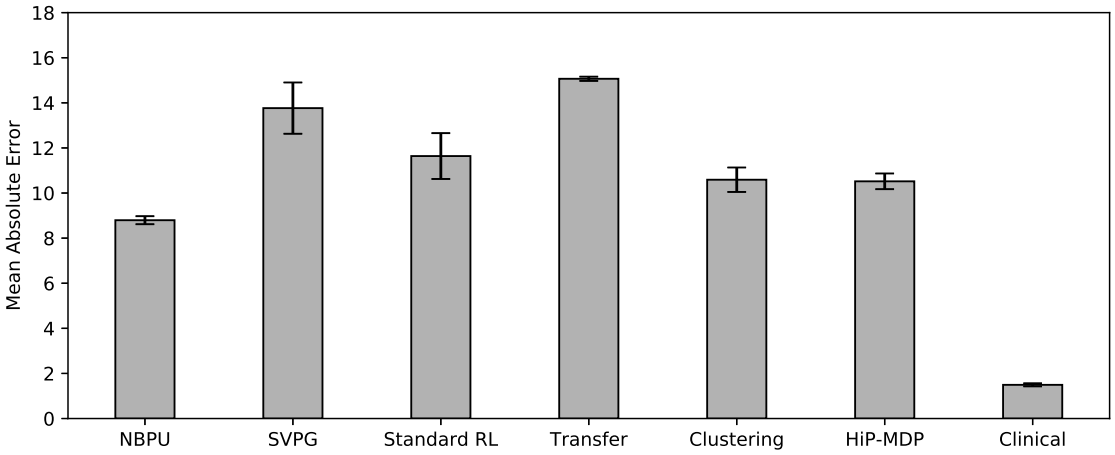

Figure 3.5: Mean absolute error (MAE) of performance prediction (i.e., difference between predicted and actual expected cumulative reward) for NBPU and all benchmark methods. Error bars represent 95% confidence intervals based on boostrapping.

than SVPG; that is, $(\beta)J_{\hat{T}'}(\theta_{i^*}) + (1-\beta)J_T(\theta_{i^*})$ more accurately predicts $J_{T'}(\theta_{i^*})$ than $J_T(\theta_{\tilde{i}})$ predicts $J_{T'}(\theta_{\tilde{i}})$ (MAE of 8.79 and 13.77, respectively, p-value< 0.01).

Figure 3.6 provides more detail on these comparisons between NBPU and SVPG. As seen in Figure 3.6(a), both methods' policy probability distributions ($\hat{P}_{T'}$ and P_T, respectively) resemble $P_{T'}$, the ground truth policy probability distribution for T'. Thus, when choosing the highest-probability policy for underrepresented patients, both methods yield policies with similar performances. Yet NBPU's probability distribution, $\hat{P}_{T'}$ more closely resembles $P_{T'}$ than SVPG's distribution P_T, with respective Kullback-Leibler divergences of 0.005 and 0.028. This is because, as seen in Figure 3.6(b), NBPU's performance prediction $(\beta)J_{\hat{T}'}(\theta_{i^*})+(1-\beta)J_T(\theta_{i^*})$ more accurately predicts policies' performance with underrepresented patients than SVPG's performance prediction $J_T(\theta_{\tilde{i}})$, which does not incorporate limited patient data from T'.

3.3.3.2 Comparison with Other Benchmarks.

NBPU significantly outperform all other benchmarks in terms of expected cumulative reward. NBPU also more accurately predicts its policies' performance with underrepresented patients compared with standard RL, transfer learning, clustering, and HiP-MDP. Note that the clinical baseline policy performs similarly across reference and underrepresented patients (expected rewards of 38.61 and 39.45, respectively), which explains this method's high accuracy in predicting its performance with underrepresented patients.

3.3.4 Robustness Check

To verify the robustness of our findings, we vary the parameters of T', the ground-truth transition model used to assess policy performances, to account for potential model misspecification. Specifically, we vary b_0^h and b_1^h (the autoregressive parameters that determine patients' heparin sensitivity) in T' by two standard errors in either direction, yielding alternate transition models T'_{-2SE} and T'_{+2SE}. Thus, T'_{-2SE} assumes that high-weight patients are *less* sensitive to heparin than assumed in T', and T'_{+2SE} assumes that high-weight patients are *more* sensitive to heparin than assumed in T' (i.e., more similar to

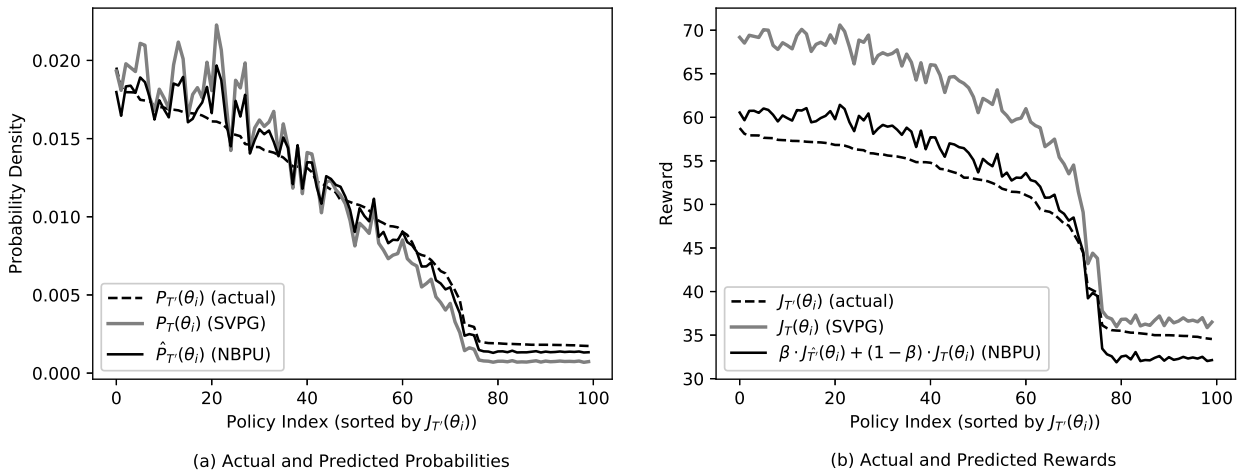

Figure 3.6: Comparison of NBPU and SVPG policies with ground truth. Sub-figure (a) shows the policy probabilities $P_{T'}(\theta_i)$ (ground truth), $\hat{P}_{T'}(\theta_i)$ (NBPU), and $P_T(\theta_i)$ (SVPG), for all $i \in \{1, \ldots, 100\}$. Sub-figure (b) shows the ground truth policy performances $J_{T'}(\theta_i)$ and performance predictions $(\beta)J_{\hat{T}'}(\theta_i) + (1-\beta)J_T(\theta_i)$ (NBPU) and $J_T(\theta_i)$ (SVPG), for all $i \in \{1, \ldots, 100\}$. Policies are sorted by their performance with underrepresented patients $(J_{T'}(\theta_i))$ in both plots. Note that NBPU results are averaged across 100 trials.

reference patients). We assess all treatment policies in both T'_{-2SE} and T'_{+2SE} to evaluate our findings' sensitivity to misspecifications in high-weight patients' heparin sensitivity.

Figure 3.7 presents each method's policy performances under T'_{-2SE}, T', and T'_{+2SE}. In T'_{-2SE}, in which high-weight patients are less sensitive to heparin than assumed in T', NBPU still outperforms standard RL, transfer learning, clustering, HiP-MDP, and the clinical baseline. The same results hold for T'_{+2SE}, in which high-weight patients are more similar to reference patients than in T'. Thus, NBPU's high performance is robust to possible misspecifications in T', regardless of the direction of the misspecification.

Figure 3.8 presents the MAE of each method's performance predictions. In T'_{-2SE}, in which underrepresented patients are less sensitive to heparin than assumed in T', NBPU's performance predictions outperforms SVPG, standard RL, transfer learning, cluster, and HiP-MDP. Understandably, this advantage dissipates in T'_{+2SE}, in which underrepresented patients are more sensitive to heparin than in T' and thus more closely resemble reference patients. In this case, methods that *only* leverage reference patient information can accurately predict their policies' performance in T'_{+2SE} given its similarity to T.

3.4 Discussion

Here, we address the problem of learning treatment policies for underrepresented patients using limited data. We propose a new approach, namely NBPU, and demonstrate that noisy Bayesian policy updates can extract useful information from a single patient's historical data even when the patient's clinical characteristics are underrepresented in training data. Such data can be used to (1) select high-performing policies for the underrepresented patient population, and (2) accurately predict these policies' performance, despite the inherent noise in such small ($n = 1$) sample sizes. We also demonstrate that NBPU is robust to misspecifications in the underrepresented patient model, and that it outperforms other benchmarks when the dissimilarity between the reference and underrepresented patient groups is magnified.

We conduct thorough benchmarking to compare NBPU with other state-of-the-art methods and clinical practice, and draw insights. One of the main benchmarks used is SVPG,

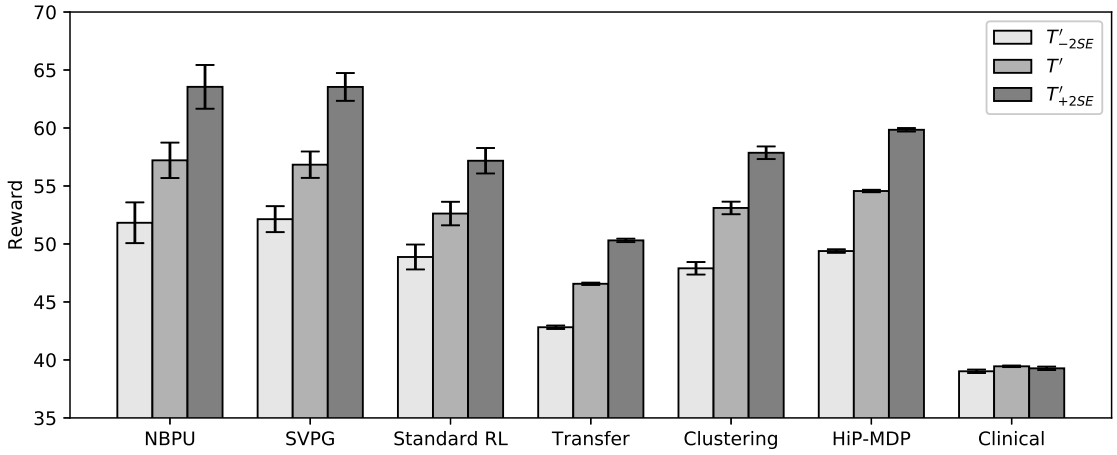

Figure 3.7: Mean policy performance in transition models T'_{-2SE}, T', and T'_{+2SE} (i.e., expected cumulative rewards, $J_{T'_{-2SE}}(\theta)$, $J_{T'}(\theta)$, and $J_{T'_{+2SE}}(\theta)$) for NBPU and all benchmark methods. Error bars represent 95% confidence intervals based on bootstrapping. Comparative performances of all algorithms are consistent across T'_{-2SE}, T', and T'_{+2SE}.

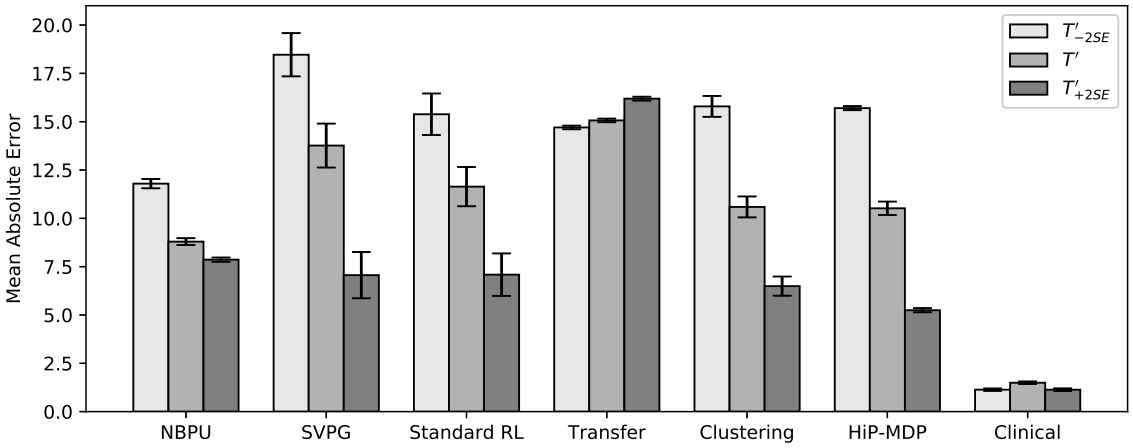

Figure 3.8: Mean absolute error (MAE) of performance prediction (i.e., difference between predicted and actual expected cumulative reward), in transition models T'_{-2SE}, T', and T'_{+2SE}, for NBPU and all benchmark method. Error bars represent 95% confidence intervals based on bootstrapping.

on which NBPU is built. Our results show that NBPU provides value above and beyond standard SVPG. Although high-probability policies under both NBPU and SVPG perform well on underrepresented patients, SVPG overestimates these policies' performance. That is, unlike NBPU, SVPG does not leverage underrepresented patient information to adjust its predictions for underrepresented patients' outcomes under its treatment policies. NBPU's use of underrepresented patient information (and its ability to regularize this information with reference patient data) allows it to more accurately predict patient outcomes.

Because NBPU uses reference patient data to regularize its performance predictions, it also outperforms transfer learning, which learns new treatment policies directly from noisy patient-specific transition models. Transfer learning has shown promise in domains where sufficient data are available from the transition model of interest. However, in the case of limited data, simply initializing the transfer learning policy parameters with those learned from reference patients does not sufficiently regularize the final policies. This explains the large error between the transfer learning policies' performance in their own (noisy) transition models and their actual performance in the underrepresented patient model T'.

NBPU also outperforms two other state-of-the-art methods, clustering and HiP-MDP, because it does *not* assume that underrepresented patients are well-represented in the training data. Clustering and HiP-MDP attempt to account for inter-patient variation, but they are not as effective as NBPU in extending their policies to patient populations that are not represented in their training data.

3.5 Conclusion and Future Work

In this study, we develop a new approach, namely, Noisy Bayesian policy updates (NBPU), for selecting high-performing reinforcement learning-based treatment policies for underrepresented patient subpopulations using limited observations. We demonstrate that NBPU can be used to effectively select treatment policies for underrepresented patients and to predict these policies' performance. It outperforms several state-of-the-art benchmarks and adds value beyond variational policy learning with reference patients only. Our results

show that even a single patient's historical data can aid decision-makers in selecting policies that perform well for patient groups, despite the inherent noisiness in such small sample sizes.

While our analysis relies on a dataset of ICU patients, our findings are expected to generalize well to other healthcare applications. Such efforts are left for future work. We also focus specifically on policy selection when only a single underrepresented patient is available, to demonstrate our method's effectiveness with such limited data. Future work also includes investigating the impact of larger underrepresented patient group sample sizes, and how to update the candidate policies as more reference patient data are collected.

Chapter 4

Optimizing Patient-Specific Medication Regimen Policies Using Wearable Sensors in Parkinson's Disease

4.1 Introduction

Effectively managing Parkinson's disease (PD) symptoms is a formidable challenge for healthcare providers. There are approximately one million Americans with PD, and this is expected to grow to 1.2 million by 2030 [102]. PD patients exhibit considerable clinical heterogeneity in their symptoms, yet many patients struggle with receiving specialized care to personalize their treatment strategies. Shortages in neurologists [125], racial and gender disparities [154], and long travel times for PD patients in rural areas [129] complicate sufficient and equitable access to care. For instance, in one multi-year study of PD patients, only 42% received neurologist care during the four-year study period [154].

It is crucial for healthcare providers to leverage emerging technologies to supplement face-to-face neurologist care in treating PD. One such technology is wearable sensors, which represent a quickly-growing consumer market in the U.S. and globally. One in five U.S. adults

currently uses a wearable health tracker or 'smart watch' [105]. Given such widespread use, wearable sensors are increasingly being utilized to monitor patient health states and behaviors [108, 21, 22, 59], and also to *recommend* optimal health interventions [31, 72, 164, 134, 28].

Such personalized interventions (often referred to as 'just-in-time adaptive interventions', or JITAIs) recommend optimal health behaviors (e.g., medication administration) based on continuous monitoring of patient health states [89, 90]. Learning wearable-based JITAIs is a sequential decision-making problem, with the goal of learning an optimal *medication policy* which maps a patient's health state to an optimal recommendation (e.g., "take medicine," "exercise for 15 minutes," etc.). Patients are assumed to stochastically transition between health states in response to recommended health behaviors, and the optimal recommendation policy maximizes some measure of expected cumulative reward over a given time horizon.

Wearable-based JITAIs are especially promising for PD patients, who demonstrate considerable heterogeneity in their symptoms and treatment responses. One promising application area is optimizing medication administration. Levodopa (L-dopa), which facilitates dopamine replacement, is generally deemed the most effective therapy for PD. Yet the interaction between patient symptoms and L-dopa is complex. Because L-dopa has a short half-life, clinicians must carefully prescribe dosages to maximize the amount of time patients' symptoms are well-managed [95, 70]. Furthermore, while L-dopa is effective in reducing *bradykinesia* (slowing of movement), it increases the likelihood of *dyskinesia* (involuntary hyperkinetic movements). Thus, effective L-dopa therapy must effectively balance these two symptoms.

The complexities of L-dopa therapy highlight the need for data-driven approaches to optimize its prescription and administration. Yet there has been no research on the application of wearable-based JITAIs to chronic medication management, which poses several challenges such as:

- Limiting the frequency of medication administration to clinically-approved intervals

- Limited the total daily dosage of each medication.

- Accounting for the complementary or competing effects of multiple medications

- Incorporating delays between medication recommendations and administration

It is also challenging and time-consuming to identify which patients may benefit from advanced therapies, such as continuous-administration L-dopa pumps or deep brain stimulation (DBS) [99, 51, 128]. Data-driven approaches to L-dopa administration can facilitate this task by assessing the marginal benefit to specific changes in patients' L-dopa regimens (such as medication frequency).

We therefore present the first implementation of a data-driven reinforcement learning (RL) framework for optimizing PD patients' medication regimens, which is also the first application of wearable-based JITAIs to chronic medication management. Our data-driven model of patients' medication responses allows us to personalize medication policies to individual patients' demographic and clinical characteristics. We also develop a novel approach for allowing the medication policy to simultaneously prescribe multiple medications.

4.1.1 Related Work

Our work is the first to combine reinforcement learning (RL) with wearable sensor data to optimize medication regimens for patients with chronic diseases (specifically PD). Thus, here we first review recent advances in the use of data analytics and machine learning to optimize PD treatment. Next, we review research on the application of RL to treatment planning problems. Lastly, we review existing work on the use of wearable sensors for patient monitoring and treatment recommendations.

4.1.1.1 Data-driven approaches to PD management.

Many studies apply machine learning to improve the accuracy of PD diagnoses based on patients' clinical characteristics [87, 155, 101, 13, 61, 113, 147, 149]. Other studies apply machine learning to predict PD patients' clinical characteristics. For instance, [120] use machine learning to predict PD patients' disease progression state, [104] predict PD patients' tremor severity, and [39] predict PD patients' risk of falls.

Despite these applications of machine learning to PD diagnosis and symptom prediction, few studies have applied machine learning to PD *treatment*. Such studies often focus on

DBS therapy (see [146]), in which surgically-implanted electrodes are used to stimulate parts of the patient's brain responsible for motor function. For instance, [40] use machine learning to process PD patients' suitability for DBS treatment from electroencephalography (EEG) readings. Other studies use machine learning to tune and regulate DBS patients' neurostimulation implants [144, 71]. [127] use machine learning in the context of both DBS and L-dopa administration, by predicting optimal changes to patients' L-dopa dosages following DBS surgery.

Two recent studies focus on applying machine learning and RL exclusively to PD medication management. [149] use RL to train optimal dosage policies for PD patients. This work is a proof of concept framework that relies on hypothetical patient profiles; in contrast, we seek to extend this framework to real-world patient data collected from wearable sensors. [63] use machine learning to identify discrete disease progression states in a cohort of PD patients, based on patients' scores on the Unified Parkinson's Disease Rating Scale (UPDRS). The authors then use RL to identify the optimal combination of drug classes to be prescribed for each disease state. In contrast, we seek to leverage objective and highly granular sensor data (rather than subjective clinical assessments such as UPDRS scores) to identify optimal medication *regimens* (i.e., specific medications, their dosages, and their timing) for individual patients.

Machine learning has thus been extensively applied to PD, though existing research primarily focuses on diagnosis and symptom prediction, rather than treatment. Studies that address treatment planning almost exclusively focus on advanced therapies (particularly DBS), rather than medication administration (despite the fact that less than 10% of patients undergo DBS therapy; [85]). Those that do focus on PD more broadly either concern detailed decisions about medication regimens but only rely on synthetic sensor data, or use subjective UPDRS data and concern high-level decisions about medication classes for different stages of PD [63]. We thus seek to build on the limited past work that uses ML for L-dopa therapy optimization [149, 63], leveraging actual patient data collected from wearable sensors to optimize patient-level medication regimens.

4.1.1.2 Reinforcement learning for treatment planning.

RL is often employed for treatment planning in data-rich inpatient critical care settings (e.g., intensive care units), which allow for continuous bedside symptom monitoring. These represent 'physician-in-the-loop' applications, since attending physicians can review algorithmic recommendations before deployment. For instance, [111] and [122] develop an RL algorithm for optimizing sepsis treatment in the intensive care unit, while [93] and [11] use RL to optimize blood anticoagulant dosing. In another example, [158] and [110] develop RL algorithms to optimally wean patients off of mechanical ventilation.

RL has also been studied for its ability to regulate implanted devices. For instance, [107] use RL to regulate electrode implants that provide neurostimulation therapy for epilepsy patients. Other studies [28, 103] use RL to regulate insulin pumps for patients with diabetes. These represent 'closed-loop' applications, because algorithmic recommendation does not rely on physician or patient involvement.

'Patient-in-the-loop' applications, in which RL-based systems recommend treatment behaviors directly to patients, are currently limited to mobile health interventions. For instance, [164] and [72] both develop RL models for learning optimally-timed exercise recommendation for users of wearable or mobile fitness trackers. [137] use RL to learn optimal app-based recommendation strategies for users of a mobile health application, and test their algorithm in a clinical trial. While such applications are potentially valuable for preventative health, RL has not been deployed for 'patient-in-the-loop' applications designed for disease treatment or symptom management.

4.1.1.3 Wearables for smart health monitoring.

Wearable sensors have been used for a wide variety of health monitoring applications. Most applications are solely 'descriptive' in that they monitor and report on a patient's health state or behavior over time. For instance, [59] use commercial fitness trackers to monitor sleep duration for insomnia patients, while [34] use commercial fitness trackers to monitor behavioral symptoms of dementia. Other approaches are 'predictive' in that they use wearable sensor data to predict clinical events of interest. [108] develop a model for

predicting the risk of post-partum depression in adolescent mothers based on wearable sensor data. Models built from wearable sensor data have also been used to predict PD patients' symptom severity [69] and medication responses [4] and to predict adverse health events in elderly patients [5].

Studies have also investigated the applications of wearable sensors to PD. Multiple studies have shown that accelerometer data from wearable sensors can produce accurate estimates of PD symptoms that correlate with patient self-report and clinical assessments, such as the UPDRS [97, 48, 53]. [88] conducted a clinical study in which one such wearable sensor, the Personal Kinetigraph (PKG; Global Kinetics Corporation), was used to monitor patients' PD symptoms for the purposes of medication management. Patients wore wrist-mounted movement sensors for two separate six-day periods; after the first six-day period, patients participated in a clinic visit in which the 'descriptive' report of symptoms produced by the PKG was used by the physicians to potentially update the patients' medication regimens. The study found improvements in physician- and patient-reported symptom severity, though PKG scores remained mostly unchanged. Note that in this study, *only the clinical intuition of the physicians was used* to adjust patients' medication regimens. Building on this work, [147] use the wearable sensors data from [88] to cluster patients based on their medication regimens and symptoms, with the aim of using new patients' PKG data to immediately 'guess' their optimal cluster allocation. As such, [147] use clustering and predictive modeling to roughly estimate optimal medication regimens (at the cluster level) using PKG data. In this study, we seek to further build on this line of research by developing a data-driven, *prescriptive* framework that uses PD patients' wearable sensor data to optimize *patient-specific* L-dopa regimens.

4.1.2 Objectives and Proposed Contributions

New RL frameworks must be developed to leverage wearable sensors for optimizing PD medication regimens. We develop the first data-driven framework for leveraging wearable sensor data to optimize medication management for chronic disease treatment. Our work specifically focuses on optimizing L-dopa therapy for PD patients, since research shows that

PD symptoms can be accurately monitored in high frequency through wearable sensors [69, 4].

Our data-driven framework extends the use of RL to 'patient-in-the-loop' treatment planning. Previous RL treatment applications are either 'closed-loop' (i.e., requiring no human interaction; [107, 28, 103]), 'physician-in-the-loop' (i.e., designed for inpatient settings with physician oversight; [111, 93, 158, 110]), or are designed for preventative health rather than treatment [164, 72, 137]. Therefore, this study extends the literature by accounting for challenges involved with 'patient-in-the-loop' systems, including minimum dosing intervals, capped daily dosages, competing effects of multiple medications, and delays between recommendations and subsequent medication administrations.

Our data-driven framework also extends research on wearable health sensors by leveraging wearable sensor data for *prescriptive* purposes. Research on wearable health sensors largely focuses on 'descriptive' analytics (i.e., patient monitoring) or 'predictive' analytics (i.e., forecasting patient outcomes or health events). By applying RL to high-frequency wearable sensor data, we develop a 'prescriptive' framework that leverages such data to improve patients' medication regimens.

Lastly, our data-driven framework produces rich clinical insights by personalizing RL-based medication policies to patient-level demographic and clinical characteristics. We particularly apply this to PD where neurologist care is necessary to improve patient quality of care and avoid preventable PD-related hospitalization, but access is limited and fraught with logistical challenges. Our patient-specific approach allows for augmenting medical decision making well beyond the descriptive values of wearable data, demonstrated by the improvements in quality of decisions despite physicians having access to the same data. In addition, our approach provides flexibility in medication regimen planning, enabling physicians to account for various factors such as propensity to adherence. Perhaps more importantly, it provides a quantitative and objective framework for facilitating the challenging and time-consuming task of identifying patients who may be candidates for advanced therapies.

4.2 Methods

In this section, we discuss the data and provide the methods and evaluations schemes. Specifically, in Section 4.2.1, we describe our data. In Section 4.2.2, we introduce the data-driven 'symptom model' that is learned from historical patient data. This model is used as part of the RL framework to simulate patients' bradykinesia and dyskinesia in response to medication and to train RL-based medication policies. In Section 4.2.3, we formalize our RL problem. Finally, in Section 4.2.4, we describe our protocol for training and evaluating patient-specific medication policies with RL. Figure 4.1 provides an overview of our methodology.

4.2.1 Data

We use data from [88], which consists of 26 PD patients who wore wrist-mounted movement trackers for two separate six-day periods. All patients were taking L-dopa at the time of the study. At the beginning of the study, patients were provided a Personal Kinetigraph (PKG; Global Kinetics Corporation), shown in Figure 4.2, and were asked to wear it daily for six days on the side of the body most affected by PD symptoms. The PKG monitored patients' movement symptoms between 5:00am and 10:00pm each day (17 hours per day) and produced scores estimating patients' bradykinesia and dyskinesia levels every two minutes. The PKG also used vibration-based medication alerts to remind patients when to administer L-dopa (based on the patients' existing L-dopa medication regimens).

After the first six-day period of wearing the PKG, patients participated in a clinical visit in which they completed routine clinical evaluations, including the Unified Parkinson's Disease Rating Scale (UPDRS) assessment. During this visit, study physicians reviewed the 'descriptive' reports of patients' PKG data from the preceding six days and re-adjusted patients' medication regimens per the PKG data, in-person evaluations, and their clinical intuition. Note that some patients received no change to their L-dopa regimens, while others did. After the first clinical visit, patients completed another six days of symptom monitoring with the PKG, followed by a second clinical visit. During the second clinical visit, patients

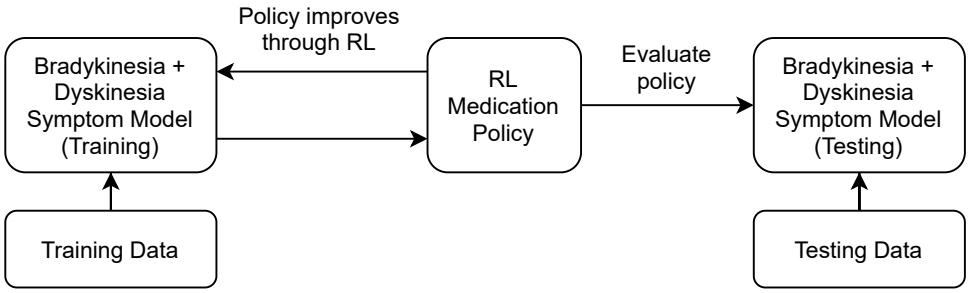

Figure 4.1: Overview of the methodology.

Figure 4.2: PKG worn by participants in [88].

participated in routine clinical evaluations (including UPDRS assessment) and completed a survey regarding their experience with the PKG.

We refer to patients' PKG data from the first six-day wear period as patients' 'Visit 1' data, and data from the second as 'Visit 2' data. We refer to patients' L-dopa regimens for their Visit 2 wear periods (i.e., those prescribed by the study physician during the first clinical visit) as 'physician-updated regimens' (since the study physicians 'updated' patients' original regimens based on their PKG reports). Patients' Visit 1 and Visit 2 data are summarized into a single PKG plot per patient, which shows the median, 25th, and 75th percentile for bradykinesia and dyskinesia scores at each two-minute time interval from 5:00am until 10:00pm, yielding two PKG plots per participant. Figure 4.3 presents an example PKG plot. Note that higher bradykinesia and dyskinesia scores indicate greater symptom severity, though bradykinesia scores appear reversed in PKG plots. Thus, distance from plot midline indicates symptom severity. Lastly, note that all patients were taking one of three L-dopa formulations at the time of the first clinical visit, namely, L-dopa IR ('immediate release'), L-dopa CR ('controlled release'), and Rytary, each of which differs in how quickly it releases L-dopa into patients' bloodstreams.

4.2.2 Bradykinesia and Dyskinesia Symptom Model

In this section, we develop a statistical model of patients' bradykinesia and dyskinesia responses to L-dopa. This enables us to simulate and evaluate patients' responses to medication dosages even if such dosages differ from those that patients actually received. It also allows us to simulate patients' bradykinesia and dyskinesia scores under the medication regimens they received during the study, and compare their simulated versus actual symptom trajectories for model validation.

We learn two symptom models (with identical structures) from patients' Visit 1 and Visit 2 data. In doing so, we use Visit 1 data as our training set and Visit 2 data as our test set, and therefore refer to the Visit 1 and Visit 2 models as the 'training' and 'testing' models, respectively. After fitting the *training model* from Visit 1 data, we validate it by confirming that it can accurately reproduce symptom trajectories from the test set (i.e., Visit

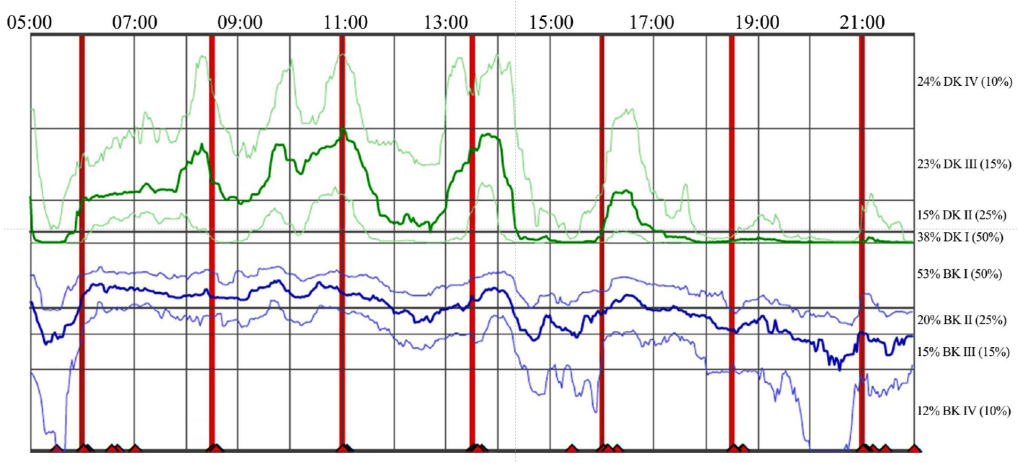

Figure 4.3: Example PKG plot [88], summarizing bradykinesia (bottom) and dyskinesia (top) scores over a six-day period. Higher bradykinesia and dyskinesia scores reflect greater symptom severity, and bradykinesia scores are reversed in all plots. Thus, distance from plot midline indicates symptom severity. Bold lines represent median bradykinesia and dyskinesia scores; lower and upper faded lines represent 25th and 75th percentiles, respectively. Vertical lines indicate prescribed medication administration and red markers at bottom of plot indicate self-reported medication administration. Score categories BK/DK I, II, III, and IV respectively indicate bradykinesia and dyskinesia scores experienced by control patients 50%, 25%, 15%, and 10% of the time, respectively, with DK/BK IV being the most severe symptom classification.

2 data). We then use it to train all RL medication policies, and use the testing model (fitted from Visit 2 data) to evaluate the RL medication policies. Thus, we assume that Visit 1 data are available from *all* patients of interest when training RL policies, and we seek to train RL policies that generalize well to each patient's future symptom trajectories (i.e., their Visit 2 data). This mirrors the design of [88], in which study physicians use clinical intuition to revise patients' L-dopa regimens *after* viewing data from each patient's Visit 1 wear period.

4.2.2.1 Data preprocessing.

Bradykinesia and dyskinesia scores are normalized between -1 and 1, with higher scores indicating greater symptom severity. We also apply an *arctanh* transformation to normalized bradykinesia scores, which allows all model predictions to be reconverted to the data's original range using a *tanh* transformation and prevent out-of-range model predictions. Since patients in this sample experience dyskinesia less frequently than bradykinesia [88], normalized dyskinesia scores were highly clustered around -1 and yield better modeling results without a *tanh* transformation (which distorts data values near 1 and -1). We then perform first-order differencing on all bradykinesia and dyskinesia scores, a common time series technique for reducing nonstationarity [29].

Although each patient was prescribed to take L-dopa at the same time each day during the six day wear period, self-report data suggests that medication was often taken up to 30 minutes early or late. We therefore adjust each patient's prescribed medication times up to 30 minutes in either direction. The adjusted times were chosen to maximize each patient's median bradykinesia score increase in the following 30 minutes (based on L-dopa absorption time; [78]). Adjusted medication administration times were, on average, 3.3 minutes earlier than the original prescribed times, with a mean absolute change of 21.7 minutes in either direction.

4.2.2.2 L-dopa pharmacokinetic models.

Recall that each patient in this study takes a subset of three different L-dopa formulations (L-dopa IR, L-dopa CR, Rytary), which differ in their *pharmacokinetic properties*. That is, these medications differ in how they deliver L-dopa to the patients' bloodstreams once

they are administered. We therefore build an *L-dopa pharmacokinetic model*, which converts patients' prescribed L-dopa IR, L-dopa CR, and Rytary dosages to L-dopa-equivalent dosages (LED) at each point. LED is a common clinical metric for summarizing the therapeutic effect of multiple PD medications [138, 124]. This allows us to estimate how patients' symptoms respond to L-dopa at each time point, regardless of the L-dopa formulation (L-dopa IR, L-dopa CR, Rytary) that was administered.

We build a model of bloodstream L-dopa concentration that depends on each L-dopa medication's peak concentration, time to peak concentration, and elimination half-life (i.e., rate of decrease after peak concentration), with parameters taken from existing clinical literature [82, 54]. For L-dopa IR, we assume a peak concentration of 10.90 ng/(mL·mg), time to peak concentration of 1 hour, and elimination half-life of 1.6 hours. For L-dopa CR, we assume a peak concentration of 8.55 ng/(mL·mg), time to peak concentration of 1.5 hours, and elimination half-life of 1.6 hours. For Rytary, we assume a peak concentration of 3.4 ng/(mL·mg), time to peak concentration of 4.5 hours, and elimination half-life of 1.9 hours. Additionally, we assume that Rytary's time to peak concentration of 4.5 hours is a result of it reaching its peak concentration in 1 hour and then holding that concentration for 3.5 hours [54, 82, 47].

Figure 4.4 plots simulated concentrations of 100mg of L-dopa IR, L-dopa CR, and Rytary under our pharmacokinetic model. Note that we assume the bloodstream concentration for each medication increases linearly from zero to its peak concentration, then declines exponentially according to its elimination half-life (consistent with pharmacological literature, [96, 123]). At each time point, we calculate each patient's total LED as the sum of the concentrations of each medication, weighted by their L-dopa equivalence factors (1.0 for L-dopa IR, 0.75 for L-dopa CR, 0.5 for Rytary; [138, 124]).

4.2.2.3 Bradykinesia and dyskinesia symptom model structure.

Figure 4.5 provides an overview of our bradykinesia and dyskinesia symptom model. We develop a model that jointly predicts patients' bradykinesia and dyskinesia scores at time $t+1$ from their most recent t_{past} bradykinesia and dyskinesia scores, time since last L-dopa dosage, and current bloodstream L-dopa concentration. We also include four patient-level

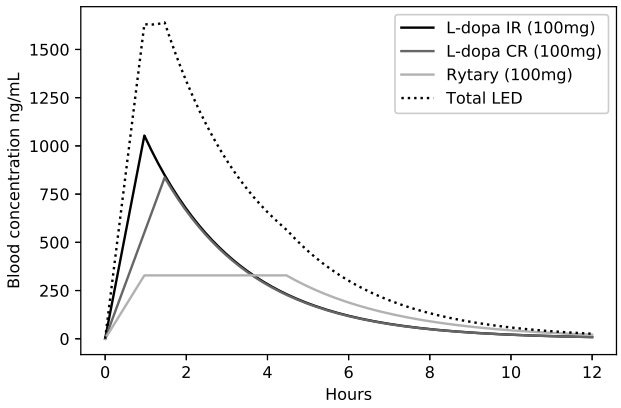

Figure 4.4: Simulated bloodstream concentrations for 100mg of L-dopa IR, L-dopa CR, and Rytary. 'Total LED' line shows total L-dopa-equivalent dosage (LED) concentration of all three medications.

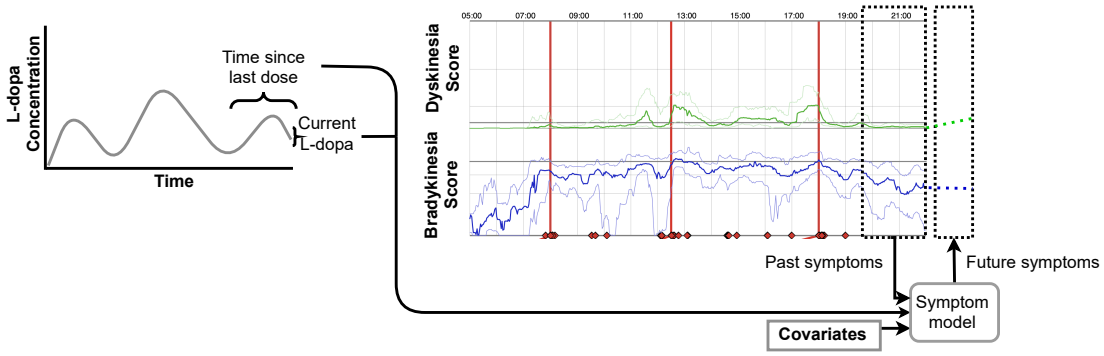

Figure 4.5: Diagram of bradykinesia and dyskinesia symptom models. At each time point, data from patients' L-dopa concentrations, their previous bradykinesia and dyskinesia scores, and demographic and clinical covariates are used to predict future bradykinesia and dyskinesia scores.

covariates: gender, age, number of years since PD diagnosis, and total daily dosage of non-L-dopa PD medications (e.g., monoamine oxidase B, or MAO-B, inhibitors), expressed in LED. This allows the model to be patient-specific, i.e., the model's predictions (and policy learning) can be tuned to each patient's specific demographic characteristics and medication regimens. Note that, since patients' maximum dyskinesia scores vary considerably more than patients' maximum bradykinesia scores, we cap each patient's simulated dyskinesia values at their 75th percentile band, plus the difference between the 75th percentile and median bands. This allows model simulations to exceed patients' observed dyskinesia scores while avoiding unreasonably high predictions for patients with overall low dyskinesia levels.

Since the number of candidate models grows quickly with the number of variables, we perform model selection in two steps. First, using a grid search, we test different lengths of t_{past} and three candidate model structures, namely, a linear autoregressive model, a neural network with one hidden layer, and a neural network with two hidden layers (where the number of hidden units is equal to t_{past}). We evaluate model structures using five-fold cross-validation on Visit 1 data, based on their root mean square error (RMSE) in predicting patients' next bradykinesia and dyskinesia scores (at time $t+1$). Note that, as discussed in Section 4.2.2, we only use Visit 1 data for model selection as Visit 2 data are reserved for validating the symptom model and evaluating the RL policies. RMSE results can be interpreted in the models' original units (i.e., normalized bradykinesia and dyskinesia scores), and is particularly selected because it penalizes large error in the models' predictions, which can yield unstable or unrealistic symptom trajectories.

After finalizing the length of t_{past} and the model structure, we add Gaussian noise with standard deviation σ_b to the model's bradykinesia predictions and Gaussian noise with standard deviation σ_d to its dyskinesia predictions. This incorporates stochasticity into the model's generated symptom trajectories. Specifically, it allows our bradykinesia and dyskinesia models to produce *distributions* over future patient states, rather than single predictions, and is common practice in healthcare RL to prevent overfitting [163, 100].

We again use five-fold cross-validation to assess different values of σ_b and σ_d. In each fold of the cross-validation, we use Visit 1 data from 80% of patients to fit bradykinesia and dyskinesia models. We then use these models to simulate 100 bradykinesia and dyskinesia

trajectories for the remaining 20% of patients under these patients' corresponding Visit 1 medication regimens. We then calculate the proportion of simulated bradykinesia and dyskinesia values that fall within each patient's 25th and 75th percentile bands. We select σ_b and σ_d such that the corresponding proportion of simulated bradykinesia and dyskinesia scores that fall within each patient's 25th and 75th percentile bands is closest to 0.5 (i.e., yields a similar degree of score variability as seen in actual patient data).

4.2.3 Reinforcement Learning Problem

Recall that optimal PD medication administration seeks to balance the competing effects of bradykinesia (primary symptom of interest) and dyskinesia (side effect of L-dopa). Thus, for each patient, we seek an optimal L-dopa dosage policy that jointly minimizes the patient's bradykinesia and dyskinesia. We define this medication planning problem as the tuple $< S, A, T, r, t_{max} >$, where

- S is the set of possible symptom states the patient can occupy, with $s_t \in S$ denoting the symptom state at time t;
- A is the set of available treatment actions, with $a_t \in A$ denoting the action taken at time t, i.e., a medication recommendation made to the patient at time t. Since patients cannot act on medication recommendations instantaneously, we assume a 10 minute delay between medication recommendation and administration. Thus, the medication recommendation a_t is implemented at time $t + 5$;
- T is the patient's state transition model; specifically, T is the symptom model defined in Section 4.2.2. Since medication recommendations are implemented with a 10 minute delay, T maps (s_t, a_{t-5}) to s_{t+1};
- $r : S \mapsto \mathbb{R}^1$ defines the immediate reward obtained in each state $s_t \in S$;
- t_{max} is the finite time horizon. Consistent with the PKG data collection interval, we discretize time in units of two minutes where time epochs are $t \in \{0, 1, \ldots, t_{max}\}$.

We define a medication policy as $\pi : S \mapsto A$, which maps a patient's state s_t to an action a_t. For a given patient, we seek an optimal medication policy π^* which maximizes the

expected cumulative reward $\mathbb{E}_\pi\left[\sum_{t=0}^{t_{\max}} r(s_t)\right]$, where the expectation is taken over the states that result from executing the policy. We set $t_{max} = \frac{60}{2} \times 17 + 1 = 511$ (equal to 17 hours of two-minute intervals), i.e., we seek to maximize the patient's reward over the course of one day of symptom measurement (5:00am to 10:00pm inclusive). In doing so, we focus our analysis on patients' waking hours since PD symptoms generally subside during sleep ([91]).

We use a simulation-based approach, in which an RL agent interacts with T (i.e., the symptom model described in Section 4.2.2) to train dosage policies through on-policy algorithms, which are considered state-of-the-art [83]. Figure 4.6 outlines our RL framework for training medication policies.

4.2.3.1 State and action definitions and spaces.

We define the state at time t as $s_t = \left(\boldsymbol{b}_{(t-t_{past}+1):t},\ \boldsymbol{d}_{(t-t_{past}+1):t}, \Delta t, l_t^{(IR)}, l_t^{(CR)}, l_t^{(Ryt)}, L_t, \boldsymbol{c}\right)$, where

- $\boldsymbol{b}_{(t-t_{past}+1):t}$ is a vector of the most recent t_{past} normalized bradykinesia scores (i.e., past $2 \cdot t_{past}$ minutes),
- $\boldsymbol{d}_{(t-t_{past}+1):t}$ is a vector of the most recent t_{past} normalized dyskinesia scores (i.e., past $2 \cdot t_{past}$ minutes),
- Δt is the time since the most recent medication administration.
- $l_t^{(IR)}$ is the patient's current bloodstream concentration of L-dopa IR
- $l_t^{(CR)}$ is the patient's current bloodstream concentration of L-dopa CR
- $l_t^{(Ryt)}$ is the patient's current bloodstream concentration of Rytary
- L_t is the total LED administered since $t = 0$.
- $\boldsymbol{c}$ is a vector of the patient-level covariates used in T (specifically, gender, age, years since PD diagnosis, and total daily dosage of non-L-dopa PD medications. Note that these covariates do not change within-patient, and therefore their values remain constant for any given patient.

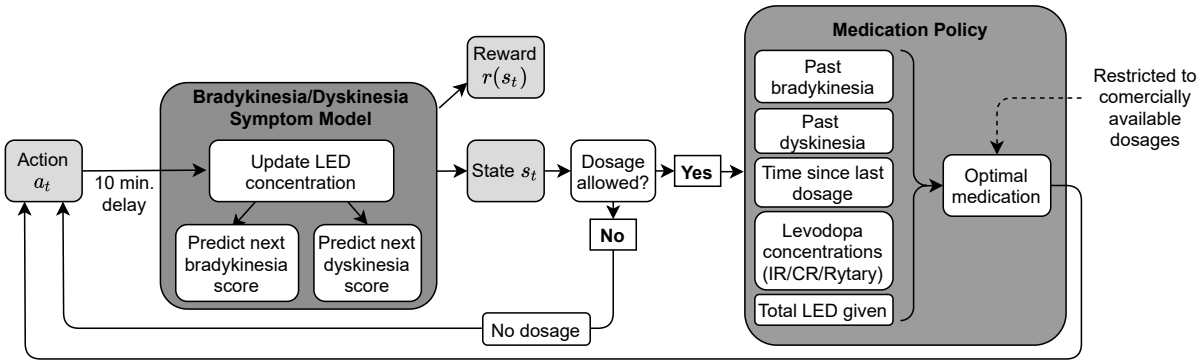

Figure 4.6: Overview of RL framework. An RL agent interacts with the bradykinesia and dyskinesia symptom model to train dosage policies through on-policy RL algorithms.

The action space A includes any commercially available dosage of L-dopa IR, L-dopa CR, and Rytary. Since some of the patients in our dataset (4 out of 26) took two L-dopa medications concurrently, and since clinical research suggests some patients may benefit from such two-drug combinations [132, 135], A also includes a set of 'allowed' two-drug L-dopa combinations. Individual L-dopa medication dosages are typically reduced when multiple medications are taken together to limit the total LED of the drug combiantion [132]; thus, A only includes two-drug L-dopa combinations whose total LED does not exceed the highest-LED medication (which is 250mg of L-dopa IR). More specifically, there are 22 available actions, $a \in A$, as follows:

- No medication
- L-dopa IR: 100mg or 250mg (dosages taken from [139])
- L-dopa CR: 100mg or 200mg (dosages taken from [140])
- Rytary: 95mg, 145mg, 195mg, or 245 mg (dosages taken from [141])
- L-dopa IR (100mg) + L-dopa CR (100mg or 200mg)
- L-dopa IR (100mg) + Rytary (95mg, 145mg, 195mg, or 245mg)
- L-dopa CR (100mg) + Rytary (95mg, 145mg, 195mg, or 245mg)
- L-dopa CR (200mg) + Rytary (95mg, 145mg, 195mg)

Dosages are only allowed every three hours (i.e., every 90 time epochs), which is a common dosing interval for PD treatment [118, 117]. Setting a minimum dosing interval ensures that patients' RL medication policies do not over-prescribe L-dopa with clinically impermissible frequencies. Each patient's daily LED is capped at the LED of their physician-updated regimen (i.e., their daily LED for the Visit 2 wear period). Thus, the subset of available actions in state s_t, which we denote A_{s_t}, is given by

$$A_{s_t} = \begin{cases} a : \ell(a) \leq L_{max} - L_t, & \text{for } \Delta t \geq 90 \\ \varnothing, & \text{otherwise,} \end{cases} \tag{4.1}$$

where $\ell(a)$ denotes the LED of a medication action a, L_{max} is a patient's maximum allowed daily LED, and $\varnothing$ is the empty set.

4.2.3.2 Reward function.

Our model assumes the following reward function:

$$r(s_t) = \mathbb{1}_{b_t \leq 0.279} - \alpha \cdot \mathbb{1}_{d_t > -0.775}, \tag{4.2}$$

where b_t is the normalized bradykinesia score at time t, d_t is the normalized dyskinesia score at time t, α is a predetermined importance weight, and $\mathbb{1}_{(\cdot)}$ denotes an indicator function. Intuitively, the reward at time t is highest when both b_t and d_t are low (i.e., symptoms are not severe) and is lowest when b_t and d_t are high (i.e., symptoms are severe). More specifically, the indicator function $\mathbb{1}_{b_t \leq 0.279}$ equals 1 if a patient's bradykinesia is considered 'controlled' (defined as a normalized score less than or equal to 0.279) and 0 if it is 'uncontrolled' (normalized score greater than 0.279). The threshold 0.279 corresponds to an un-normalized bradykinesia score of 25, which serves as the cutoff for 'uncontrolled' bradykinesia in the clinical literature [88, 98]. Similarly, the indicator function $\mathbb{1}_{d_t > -0.775}$ equals 1 if a patient's dyskinesia is considered 'uncontrolled' (normalized score greater than -0.775) and 0 if it is 'controlled' (normalized score less than or equal to -0.775). The threshold -0.775 corresponds to an un-normalized dyskinesia score of 9, which serves as the cutoff for 'uncontrolled' dyskinesia in the clinical literature [88, 98].

Thus, the optimal policy prescribes sufficient L-dopa to maintain controlled bradykinesia, but avoids prescribing too much to cause uncontrolled dyskinesia, with these competing objectives balanced by the importance weight α. We proceed with $\alpha = 0.5$ for our experiments, which leads to stable policy learning in preliminary tests and reflects the clinical observation that patients generally prefer well-managed bradykinesia at the expense of some dyskinesia, over uncontrolled bradykinesia without any dyskinesia [27].

4.2.3.3 Policy network: Allowing multiple medications.

RL policy neural networks typically produce a softmax probability distribution over available actions, with one action being selected per time period. Our policy network structure is complicated by our policies' ability to recommend medication *combinations*, i.e., simultaneous actions.

RL policy networks typically allow for simultaneous actions either by (1) expanding the size of the action space to include all joint action combinations, or (2) treating each action node as a separate Bernoulli distribution, allowing multiple actions to be sampled independently (e.g., [50]). The former can quickly lead to combinatorial explosion of the model's parameter space, and preliminary tests suggests it yields poor medication policies. The latter assumes independence among the available actions (an untenable assumption for multiple medications with similar effects) and has no mechanism for prohibiting specific action combinations [50].

We therefore develop a customized policy network structure in which allowing for k possible action combinations requires only k additional model parameters, without requiring action independence. Figure 4.7 presents a schematic example of our policy network architecture. The state vector s_t first enters the network through an input layer and passes to a second layer with $m+1$ neurons, where each neuron represents one of m possible *single-medication* actions, plus a 'no medication' action. As described in Section 4.2.3.1, there are two possible L-dopa IR dosages, two L-dopa CR dosages, and four Rytary dosages, yielding $m = 8$. This layer is subject to a softmax activation (which produces values between 0 and 1) and then a subsequent logarithmic activation (which produces negative values, given the previous softmax activation).

The model's third and final layer contains $k + m + 1$ neurons, where each neuron corresponds to one of m available single-medication actions, k two-medication combinations, or the 'no medication' action. Note that only *allowed* medication combinations are represented with neurons in this layer. For the $m + 1$ neurons that correspond to single-medication actions or the 'no medication' action, their activation values are carried over from their corresponding neuron in the previous layer (via connection weights fixed to 1). For the

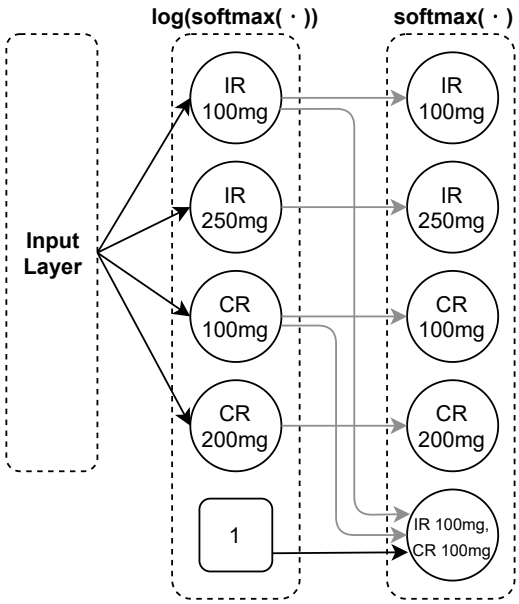

Figure 4.7: Example multi-action policy network, shown only for L-dopa IR and CR and one allowed medication combination (L-dopa IR 100 mg + L-dopa CR 100mg). Gray connections are set to unity and are not trainable, while black connections are trainable.

k neurons that correspond to two-medication combinations, their activation values are the *sum* of their constituent medications' activations from the previous layer - which, given that all activations from the previous layer are negative, is necessarily *less* than either constituent medication's activation. Intuitively, this is consistent with the rules of joint probability under independence, in that the activation of a two-medication combination is the logarithm of the product (i.e., the sum of the logarithms) of its two constituent medications' activations.

To correct for this independence assumption, each two-medication combination neuron has a *trainable bias* that can increase or decrease its activation relative to its default activation value. Positive biases 'up-regulate' the selection probability of a two-medication combination, relative to its selection probability under an independence assumption, and negative biases 'down-regulate' its selection probability. Finally, the entire output layer is subject to a softmax activation, producing a probability distribution over all single-medication actions and all allowed dual-medication actions.

4.2.4 Policy Training and Evaluation

We train one RL policy for each patient using the training model (i.e., the bradykinesia and dyskinesia symptom model learned from *Visit 1 data only*). For each patient's policy, we impute that patient's covariates (gender, age, years since PD diagnosis, and daily dosage of non-L-dopa PD medications) into the state space, as described in Section 4.2.3.1. Note that we only learn policies for patients in our dataset and do not extrapolate our method to combinations of covariates not observed in our dataset; this approach allows us to compare each patient-specific RL policy with the actual policy that a patient received in the clinical study. For each patient-specific RL policy, we also cap the policy's daily LED at that patient's physician-updated daily LED. This allows us to assess whether RL can improve patient outcomes without simply recommending more medication. We train each RL policy using the policy gradient algorithm over 2000 epochs, which proves sufficient for policy convergence in preliminary tests. We also entropy-regularize the policy network loss function to encourage action exploration. Since reinforcement learning algorithms can arrive at locally optimal solutions [162, 33, 2], we repeat all RL training runs from five randomly selected initializations and choose the best-performing policy (based on training loss) for evaluation.

To evaluate policies' performance, we calculate their expected cumulative rewards in the testing model (i.e., the symptom model learned from *Visit 2 data only*). This is a common evaluation approach in the wearable-based JITAI literature [134], and it allows us to assess whether medication policies learned from patients' initial data generalize well to their future symptom trajectories. Thus, no training data is used to evaluate policies. Note that, while policies randomly sample actions from the policy network's softmax output at training, only the highest-probability action is selected during testing.

As a benchmark, we also execute patients' physician-updated medication regimens in the testing model. We compare the performance of the physician-prescsribed regimens to those of the patient-specific RL policies. As described in section 4.2.1, physicians prescribed medication regimens for patients' Visit 2 wear period *after* viewing their Visit 1 PKG data. Thus, the physician-updated regimens were based on the same PKG data that are used to train the RL policies (i.e., Visit 1 PKG data), providing a rigorous assessment of our RL approach's ability to improve and augment symptom management *above and beyond* the added value of the descriptive PKG data itself.

4.3 Results

Here, we present the results of our numerical analysis. Specifically, in Section 4.3.1, we summarize patients' demographic and clinical characteristics. In Section 4.3.2, we present model selection and validation results for our bradykinesia and dyskinesia symptom model, demonstrating their ability to simulate realistic patient symptom trajectories. In Section 4.3.3, we present results for our patient-specific RL medication policies and compare them with physician-updated policies. In Section 4.3.4, we assess the impact of changing patients' dosing frequencies on symptom outcomes, thereby identifying patients who may be candidates for advanced therapies (such as continuous-administration L-dopa pumps or DBS) or who may be able to switch to lower-frequency administrations without sacrificing symptom control. Finally, in Section 4.3.5.1, we assess our results' robustness to model parameters used in the L-dopa pharmacokinetic model.

4.3.1 Patient Characteristics

Table 4.1 summarizes patients' demographic and clinical characteristics. Patients skew male (i.e., 9 females vs 17 males) with a median age of 73 and a median of 3.5 years since PD diagnosis. At the time of the first clinical visit (i.e., Visit 1), 22 patients were taking L-dopa IR (mean daily dosage 439mg), five were taking L-dopa CR (mean daily dosage 460mg), and two were taking Rytary (mean daily dosage 1360mg). Eighteen patients were taking L-dopa IR only, five patients were taking L-dopa IR and CR, two patients were taking Rytary only, and one patient was taking all three medications. Six patients were also taking non-L-dopa PD medications. The median daily LED during the Visit 1 wear period was 490mg, compared with 640mg during the Visit 2 wear period. Thus, study physicians tended to revise the regimen to prescribe higher daily LED from Visit 1 to Visit 2 wear periods.

4.3.2 Symptom Model Selection and Validation

Here, we first describe the structure of our final symptom models, based on the model selection process described in Section 4.2.2.3. Next, we present model validation results for our final symptom model.

4.3.2.1 Symptom model selection.

As discussed in Section 4.2.2.3, we perform model selection in two steps. We first use five-fold cross validation to select t_{past} and the symptom model's structure (i.e., linear model, one-layer neural network, two-layer neural network). We then fix t_{past} and the model structure and use another five-fold cross validation to select the standard deviations σ_b and σ_d for the Gaussian noise that is added to the model's bradykinesia and dyskinesia predictions.

Table 4.2 presents the RMSE for the combinations of t_{past} and model structure evaluated. We test $t_{past} \in \{15, 30\}$ (i.e., 30 or 60 minutes of previous bradykinesia and dyskinesia) across a linear model, a neural network with one hidden layer, and a neural network with two hidden layers (with t_{past} hidden units). A linear model with $t_{past} = 30$ significantly outperforms all neural network models (all p-values< 0.05 with paired t-tests). Although it does not significantly outperform the linear model with $t_{past} = 15$ (p-value $= 0.19$), we

Table 4.1: Summary of patient characteristics and medication regimens prior to Visit 1. 'IQR' refers to interquartile range. 'UPDRS III' refers to summary score of United Parkinson's Disease Rating Scale Motor Examination (possible scores range from 0-108). 'LED' refers to L-dopa equivalent dosage. UPDRS, bradykinesia, dyskinesia, and medication are reported as of Visit 1 (i.e., prior to intervention by study physicians). Mean dosages are calculated only from patients taking that medication.

Demographic Characteristics	
Total No. Patients	26
Females/Males	9/17
Age (median)	73 (IQR 64-80)
Clinical Characteristics	
Years with PD (median)	3.5 (IQR 2-9)
UPDRS III score (median)	25.5 (IQR 17.5-33)
Mean bradykinesia (normalized)	0.01
Mean dyskinesia (normalized)	-0.94
No. taking L-dopa IR	22
No. taking L-dopa CR	5
No. taking Rytary	2
No. taking non-L-dopa medication	6
L-dopa IR daily dose (mean)	439 mg
L-dopa CR daily dose (mean)	460 mg
Rytary daily dose (mean)	1360 mg
Daily LED (mean)	490 mg

Table 4.2: Model selection results. 'Linear' refers to linear autoregressive model, 'NN' refers to neural network with t_{past} hidden units. RMSE is root mean square error across, using five-fold cross validation on patients' Visit 1 data. Recall that data are collected every two minutes so t_{past} represents $2 \cdot t_{past}$ minutes of scores.

Model Structure	t_{past}	RMSE
Linear	15	0.0296
Linear	**30**	**0.0290**
1-layer NN	15	0.0330
1-layer NN	30	0.0319
2-layer NN	15	0.0319
2-layer NN	30	0.0307

proceed with $t_{past} = 30$ since it yields the lowest RMSE. Thus, we proceed with a linear model with $t_{past} = 30$ for all analyses.

Next, we identify the optimal standard deviations σ_b and σ_d to incorporate into the symptom model. For bradykinesia predictions, we test $\sigma_b \in \{0.00975, 0.0195, 0.039, 0.078\}$, which respectively equal 25%, 50%, 100%, and 200% of the standard deviation of the model's bradykinesia predictions. For dyskinesia predictions, we test $\sigma_d \in \{0.00325, 0.0065, 0.013, 0.026\}$, which respectively equal 25%, 50%, 100%, and 200% of the standard deviation of the model's dyskinesia predictions. The respective percentages of simulated trajectories that fall within patients' 25th and 75th percentile bands for $\sigma_b \in \{0.00975, 0.0195, 0.039, 0.078\}$ are 51.7%, 48.6%, 44.5%, and 40.7%, while the respective percentages for $\sigma_d \in \{0.00325, 0.0065, 0.013, 0.026\}$ are 54.5%, 51.6%, 46.2%, 38.9%. We thus proceed with standard deviations of $\sigma_b = 0.0195$ and $\sigma_d = 0.0065$, since these values yield proportions closest to 50% (48.6% and 51.6%, respectively).

4.3.2.2 Symptom model validation.

As described in Section 4.2.2.3, we assess whether the symptom model fitted to patients' Visit 1 data (the training model) can reproduce realistic symptom trajectories from the test set (i.e., patients' Visit 2 data). Specifically, we use the training model to generate 100 bradykinesia and dyskinesia symptom trajectories for each patient using their Visit 2 medication regimens. We then compare these simulated symptom trajectories with patients' actual Visit 2 data. For comparison purposes, we do the same for a 'benchmark' symptom model, which only take past bradykinesia and dyskinesia scores as input features. This allows us to validate our patient-specific modeling approach which leverages patient-level covariates and pharmacokinetic estimates of L-dopa concentration.

Table 4.3 compares simulated bradykinesia and dyskinesia symptom trajectories under our proposed symptom model and the benchmark model with patients' actual Visit 2 data. We compare the trajectories in terms of mean bradykinesia and dyskinesia scores, including mean scores before and after patients take their first L-dopa dosage. This allows us to assess our model's ability to successfully replicate patients' 'untreated' (i.e., before L-dopa) and 'treated' (i.e., after L-dopa) symptom levels. We also compare trajectories in terms

Table 4.3: Characteristics of actual and simulated Visit 2 bradykinesia and dyskinesia trajectories. 'Mean score' refers to mean bradykinesia and dyskinesia scores. 'Mean score (pre-L-dopa)' and 'Mean score (post-L-dopa)' respectively refer to mean bradykinesia and dyskinesia scores before and after patients' first L-dopa dosage. '% between 25th/75th' refers to proportion of simulated trajectories that fall between patients' 25th and 75th percentile bands.

	Mean score	**Mean score (pre-L-dopa)**	**Mean score (post-L-dopa)**	**% between 25th/75th**
Bradykinesia				
Patient data	0.023	0.200	-0.059	50.0%
Proposed model	0.011	0.189	-0.018	51.8%
Benchmark model	0.194	0.259	0.189	55.8%
Dyskinesia				
Patient data	-0.956	-0.986	-0.952	50.0%
Proposed model	-0.952	-0.986	-0.947	53.0%
Benchmark model	-0.950	-0.988	-0.944	53.2%

of their variability, i.e., the proportion of scores that fall within patients' 25th and 75th percentile bands.

As seen in Table 4.3, our model's simulations accurately reproduce patients' average symptom scores, including their average scores before and after taking their first medication dosage. Our model also produces trajectories with a similar degree of variability as patients' actual symptom scores. The benchmark model produces comparable dyskinesia trajectories but overestimates the severity and variability in patients' bradykinesia scores. Thus, our proposed symptom model produces realistic symptom trajectories and adds value beyond a simple benchmark model. Figure 4.8 presents simulated and actual Visit 2 bradykinesia and dyskinesia trajectories for three example patients.

4.3.3 RL Medication Policies

In this section, we first contrast our patient-specific RL policies with those of patients' physician-updated medication regimens (i.e., the medication regimens for patients' Visit 2 PKG wear period). We then present a case study that examines the RL policy's medication recommendations and symptom trajectories for a single patient in our dataset to provide further insights into the RL policy's recommendations.

4.3.3.1 RL policies versus physician-updated regimens.

Table 4.4 compares patient-specific RL policies with physician-updated regimens in terms of dosage recommendations, frequency of recommended medication administration, and expected return, averaged across all 26 patients. Compared with physician-updated regimens, the RL policies yield a higher expected cumulative reward. This is due to the RL policies yielding nearly double the rate of controlled bradykinesia compared with the physician-updated regimens.

Unlike the physician-updated medication regimens, the RL policies recommend L-dopa CR or Rytary for all patients. The RL policies also recommend lower daily dosages than the physician-updated regimens, which yields overall lower daily LED. The RL policies also break patients' daily dosages down into smaller, more frequent administrations. In fact, the

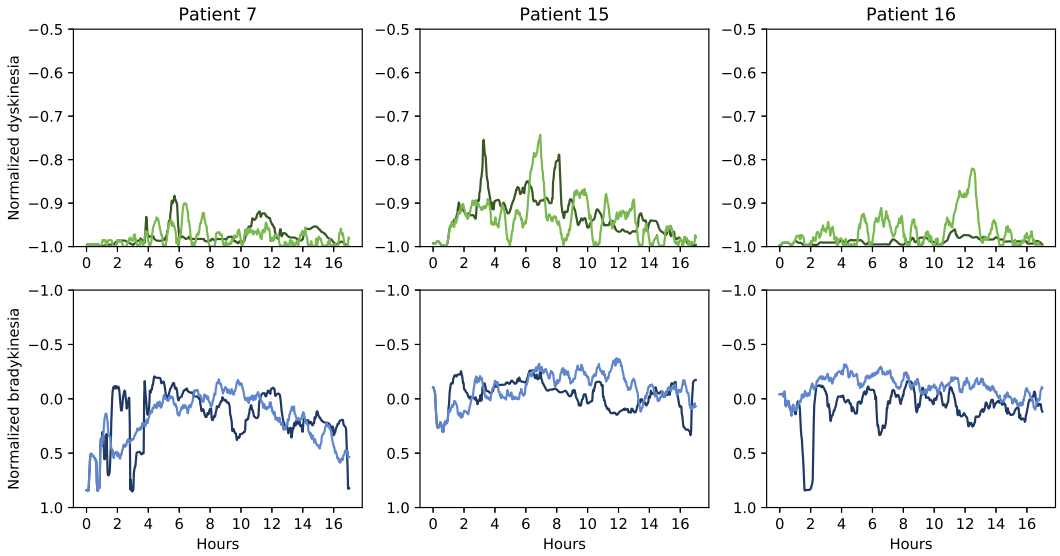

Figure 4.8: Actual (dark-colored) and simulated (light-colored) bradykinesia and dyskinesia trajectories for three example patients per their Visit 2 data. Symptom model used for simulation are learned from patients' Visit 1 data. Trajectories generally follow patients' increases and decreases in symptom scores, while still incorporating random variation into each simulation. Recall that, consistent with PKG literature, bradykinesia scores are reversed in all plots for visualization purposes.

Table 4.4: Characteristics of physician-updated (i.e., Visit 2) medication regimens and RL medication policies. 'LED' represents L-dopa-equivalent dosage across L-dopa IR, L-dopa CR, and Rytary. Mean dosages are calculated only from patients taking that medication. All metrics are based on median results across 100 simulations per patient.

	Physician	**RL**
Medications		
No. taking L-dopa IR	23	0
No. taking L-dopa CR	3	1
No. taking Rytary	3	25
L-dopa IR daily dose	554 mg	–
L-dopa CR daily dose	433 mg	400 mg
Rytary daily dose	1950 mg	534 mg
Daily LED (mg)	640 mg	269 mg
Doses per day	4.0	5.1
Dosing frequency (hrs)	3.8	3.1
Outcomes		
Avg. % controlled bradykinesia	20.5%	38.9%
Avg. % controlled dyskinesia	100.0%	99.9%
Avg. cumulative reward	97.7	183.7

average median dosage frequency across patients is every 3.1 hours, which is close to the minimum dosage frequency of 3 hours.

Altogether, patients' RL policies seek to induce stable L-dopa concentrations by frequently administering smaller L-dopa dosages and by relying on controlled-release L-dopa formulations (primarily Rytary). Indeed, the average standard deviations of patients' simulated L-dopa concentrations under the RL policies is 0.03, compared with 0.11 under physician-updated regimens. This suggests the RL policies induce more stable L-dopa concentrations than physician-updated medication regimens.

Lastly, it is worth noting that while two-medication actions (i.e., simultaneous administration of two L-dopa medications) were selected in the beginning of policy training, no patient's final RL policies recommended such actions. That is, the policies 'down-regulated' the activations of all two-medication actions such that patients were only recommended to administer one medication at a time during testing, which is in line with the policies' tendency to frequently administer small L-dopa dosages.

4.3.3.2 Case study.

To gather further insights on the RL-based policies, we investigate the RL policy recommendations for an 87-year old male patient (patient 16 in Figure 4.8), diagnosed with PD at age 77. The patient was originally prescribed 300mg of L-dopa CR before their Visit 1 wear period, which was modified to 500mg of L-dopa IR following physician evaluations in Visit 1 and ahead of their Visit 2 wear period.

Figure 4.9 shows one day of simulated bradykinesia and dyskinesia scores for this patient under their physician-updated regimen (500mg L-dopa IR, divided across three administrations) and RL policy. The patient's RL policy recommends 95mg of Rytary five times daily (475mg total), administered approximately every three hours. Under the physician-updated regimen, the patient's bradykinesia deteriorates around mid-day, with their few L-dopa IR dosages unable to reverse this decline. Under the RL policy, the patient's bloodstream L-dopa concentration increases slowly with each Rytary administration (95 mg each). This gradual increase in their bloodstream L-dopa concentration yields 'controlled'

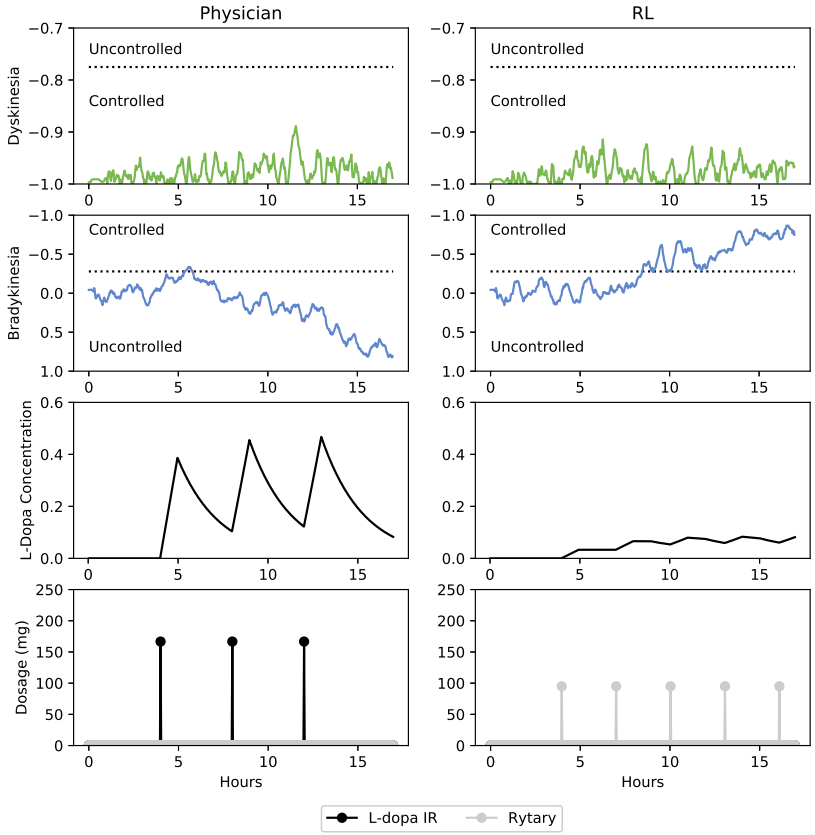

Figure 4.9: Simulated bradykinesia (top row), dyskinesia (second row), L-dopa concentrations (third row), and medication recommendations (bottom row) under the physician (left) and RL (right) policies for a single case study patient. Dashed lines in the top and second rows distinguish 'controlled'/'uncontrolled' bradykinesia and dyskinesia, respectively. Consistent with clinical literature, bradykinesia scores are reversed for visualization purposes.

bradykinesia around midday, and the patient's bradykinesia continues to improve throughout the remainder of the day.

4.3.4 Investigating the Impact of Dosing Interval Changes

Because we learn separate RL policies for each patient, our framework can predict which patients would benefit from different medication regimens or advanced therapies. While our original analysis uses a minimum three-hour dosing interval, in this section we investigate the patient-specific effects of both increasing and decreasing this parameter. We re-run all patients' RL policies with alternative restrictions on dosing intervals, and compare the expected reward under the new policies with those under the original policies.

4.3.4.1 Decreasing dosing frequency to promote adherence.

We investigate the patient-specific impact of increasing the minimum dosing interval to four hours (i.e., *decreasing* patients' dosing frequency), which is a commonly used dosing interval [68] and is the modal dosing interval in our dataset. Medication adherence has been shown to improve as dosing intervals increase [25]; thus, although three hours is a clinically permissible dosing interval, some patients may adhere better to less frequent dosing. We thus re-train new RL policies for each patient with a minimum four-hour dosing interval (which we refer to as 'four-hour RL policies') and compare patients' symptom control under these policies with the baseline three-hour RL policies to identify the patients who could be offered a lower dosing frequency without sacrificing outcomes.

Note that patients should fare no better under a four-hour minimum dosing interval than under a three-hour minimum dosing interval (since the latter is less restrictive). Due to local optimality (a well-documented issue with RL; [161, 2, 33]), in our results, one patient's four-hour policy outperforms their three-hour policy. After additional examination, this is resolved when different random initializations are used when training the policy. However, since the purpose of this analysis is to identify patients who respond *poorly* to a four-hour minimum dosing interval and to keep all analyses consistent, we simply report our

original results and classify this patient as experiencing 'no deterioration' in symptoms when switching from the three-hour to four-hour RL policy.

Table 4.5 compares the baseline three-hour RL policies to the four-hour RL policies. For nearly half of patients (12 out of 26), the number of hours spent with 'controlled' symptoms is predicted to decrease by one hour or less under a four-hour dosing interval. Such patients may be viable candidates to reduce their L-dopa dosage frequency (which could improve adherence) without considerably sacrificing symptom control. Yet 9 patients are predicted to experience more than two fewer hours of controlled symptoms under a four-hour dosing interval. Physicians might encourage these patients to adopt a three-hour dosing interval over a four-hour dosing interval to maximize their symptom control.

4.3.4.2 Increasing dosing frequency to identify candidates for advanced therapies.

We also investigate the patient-specific impact of decreasing the minimum dosing interval to two hours (i.e., *increasing* patients' dosing frequency). This allows us to investigate which patients might be candidates for advanced therapies, e.g., L-dopa pumps or DBS. We thus re-train new RL policies for each patient with a minimum two-hour dosing interval (which we refer to as 'two-hour RL policies') and compare patients' symptom control under these policies with the baseline three-hour RL policies to identify the patients who could be viable candidates for advanced therapies.

Note that patients should fare no worse under a two-hour minimum dosing interval than under a three-hour minimum dosing interval; however, in our results, two patients' two-hour policies underperform their three-hour RL policies due to local optimality. This is resolved when different random initializations are used. Since the purpose of this analysis is to identify patients who respond *well* to a two-hour minimum dosing interval, we simply report our original results and classify these patients as experiencing 'no improvement' in symptoms when switching from the three-hour to two-hour RL policy.

Table 4.6 compares the baseline three-hour RL policies to the two-hour RL policies. Most patients would benefit from higher-frequency L-dopa administration to varying degrees. Two patients are predicted to fare considerably better under a two-hour minimum dosing

Table 4.5: Change in patients' daily number of hours with controlled symptoms when switching from three-hour to four-hour minimum dosing intervals. Hours of controlled symptoms are calculated as the number of two-minute time epochs in which patients' bradykinesia and dyskinesia are both 'controlled,' divided by 30. All hour intervals are right-inclusive.

Change in # of Hours with controlled symptoms	**# of patients**
No deterioration	6
0-1 fewer hrs	6
1-2 fewer hrs	5
2-3 fewer hrs	5
>3 fewer hrs	4

Table 4.6: Change in patients' daily number of hours with controlled symptoms when switching from three-hour to two-hour minimum dosing interval. Hours of controlled symptoms are calculated as the number of two-minute time epochs in which patients' bradykinesia and dyskinesia are both 'controlled,' divided by 30. All hour intervals are right-inclusive.

Change in # of Hours with controlled symptoms	**# of patients**
No improvement	5
0-1 more hrs	6
1-2 more hrs	7
2-3 more hrs	6
>3 more hrs	2

interval, with more than three additional hours of controlled symptoms predicted each day. These patients may be viable candidates for advanced therapies, such as L-dopa pumps or DBS, which provide higher-frequency symptom management compared with traditional oral medication regimens.

4.3.5 Robustness Analysis

Finally, we assess our results' robustness to alternative model parameters. Specifically, we test whether our RL policies continue to outperform physician-updated regimens under alternative values for the L-dopa pharmacokinetic parameters and under alternative delays between medication recommendation and administration. This allows us to assess whether our approach yields viable medication policies under possible misspecifications in our model parameters.

4.3.5.1 Robustness analysis for L-dopa pharmacokinetic parameters.

The L-dopa pharmacokinetic model described in Section 4.2.2.2 uses published averages for L-dopa peak concentration, time to peak concentration, and half-life as 'baseline' values. Yet the values of these parameters can vary between patients, possibly impacting the utility of RL medication policies. To address this, we perform a robustness analysis in which we use the symptom models fitted under the baseline L-dopa pharmacokinetic model parameters to learn the RL policies, but allow these parameters to differ from their baseline values during testing. We then compare the performance of the RL policies with physician-updated regimens and provide insights.

Specifically, we identify 'low' and 'high' cases for each pharmacokinetic parameter (peak concentration, time to peak concentration, and half-life). In each case, the values of the parameter for all three medications (L-dopa IR, L-dopa CR, Rytary) are either lower or higher (respectively) than the base case assumed in Section 4.2.2.2. We then re-estimate participants' L-dopa concentrations under each combination of low and high cases for each of the three parameters (of which there are $2 \times 2 \times 2 = 8$ total). This allows us to account for uncertainty in patients' actual L-dopa absorption profiles.

For each of the eight sets of pharmacokinetic parameters, we fit a new testing model (i.e., symptom model learned from Visit 2 data) in which the patients' L-dopa concentrations are re-calculated according to these parameters. We then re-evaluate the RL policies from Section 4.3.3.1 and the physician-updated regimens in this testing model, which allows us to assess whether the results from Section 4.3.3.1 are robust to possible misspecifications in L-dopa's pharmacokinetic properties. We repeat this process for all eight sets of pharmacokinetic parameters and compute the RL policies' and physician-updated regimens' expected cumulative rewards for each case.

We base our 'low' and 'high' values for each pharmacokinetic parameter on clinical literature [54, 82]. For peak bloodstream concentration, we vary the base case values from Section 4.2.2.2 (i.e., 10.90 ng/(mL·mg) for L-dopa IR, 8.55 ng/(mL·mg) for L-dopa CR, 3.4 ng/(mL·mg) for Rytary) two standard deviations in either direction to yield the 'low' and 'high' case values. We use standard deviations of 4.0 ng/(mL·mg) for L-dopa IR, 3.0 ng/(mL·mg) for L-dopa CR, and 0.7 ng/(mL·mg) for Rytary, taken from [54]. Time to peak concentration (which is typically reported in terms of median/range, rather than mean/standard deviation) can vary from 0.5 to 2 hours for L-dopa IR, 1 to 2 hours for L-dopa CR, and 0.5 to 8 hours for Rytary [54, 82]. We thus use these values as the 'low' and 'high' case values. Recall that for Rytary, we initially assume its concentration increases for one hour and holds its peak concentration for 3.5 hours. Hence, we maintain this same ratio between its 'increase' and 'hold' times for the 'low' and 'high' time to peak concentrations of 0.5 hours and 8 hours. Lastly, we vary each medication's half-life by two standard deviations in each direction, using standard deviations of 0.2 hours for L-dopa IR, 0.2 hours for L-dopa CR, and 0.7 hours for Rytary [54, 82]. Table 4.7 summarizes the 'low' and 'high' values for each parameter across L-dopa IR, L-dopa CR, and Rytary.

Table 4.8 presents the results of the sensitivity analysis. The RL policies outperform physician-updated regimens across all tested combinations of pharmacokinetic parameters. These results suggest that the RL policies outperform physician-updated medication regimens even if the pharmacokinetic parameters describing patients' L-dopa responses differ from those used when training the models and learning the RL policies.

Table 4.7: L-dopa pharmacokinetic parameters for sensitivity analysis, based on clinical literature. 'Peak' refers to peak L-dopa concentration.

Medication	Parameter	Low	High
L-dopa IR	Peak	2.9 ng/(mL·mg)	18.9 ng/(mL·mg)
	Time to peak	0.5 hrs	2 hrs
	Half-life	1.2 hrs	2.0 hrs
L-dopa CR	Peak	2.55 ng/(mL·mg)	14.55 ng/(mL·mg)
	Time to peak	1 hr	2 hrs
	Half-life	1.2 hrs	2.0 hrs
Rytary	Peak	2.0 ng/(mL·mg)	4.8 ng/(mL·mg)
	Time to peak	0.5 hrs	8 hrs
	Half-life	0.5 hrs	3.3 hrs

Table 4.8: Robustness analysis results for L-dopa pharmacokinetic parameters. Table shows physician-updated regimens' and RL medication policies' expected cumulative reward under all combinations of pharmacokinetic parameters. RL policies are trained with the original symptom model from Section 4.3.2 and evaluated in testing models that are based on alternative L-dopa parameters. 'Peak' refers to peak L-dopa concentration.

Peak	Time to peak	Half-life	Avg. Cumulative Reward (Physician)	Avg. Cumulative Reward (RL)
Low	Low	Low	34.3	106.4
Low	Low	High	40.5	126.5
Low	High	Low	41.6	127.0
Low	High	High	51.4	137.4
High	Low	Low	89.1	144.9
High	Low	High	105.0	161.5
High	High	Low	116.0	179.0
High	High	High	122.1	177.4

4.3.5.2 Robustness analysis for medication intake delay.

Our original model assumes that patients take L-dopa 10 minutes after medication is recommended. We assess our RL policies' performance when this delay is stochastic, rather than deterministic. We re-evaluate the RL policies from Section 4.3.3 assuming the intake delay is uniformly distributed between 2 and 20 minutes. Note that we only assume a stochastic medication intake delay for the RL policies; as in Section 4.3.3, we assume medications in the physician-updated regimens are taken without delay since they are prescribed at fixed times throughout the day.

Table 4.9 presents the average cumulative reward for RL medication policies under a stochastic intake delay, alongside the rewards under the physician-updated regimens (copied from Table 4.4). While the RL policies' performance is lower than reported in Section 4.3.3 (under a fixed 10-minute delay), they still considerably outperform the physician-updated-regimens under a stochastic delay.

Note that the 'patient-in-the-loop' consideration of a 10-minute intake delay when learning the RL policies is indeed important and necessary. To further assess the benefits of this consideration, we re-train all RL policies assuming that patients take medication 'immediately,' i.e., in the next available time epoch (two minutes) upon receiving a medication recommendation. When evaluated in the stochastic delay testing model, these re-trained RL policies yield an expected cumulative reward of 138.8, compared with 163.6 for the RL policies trained with a 10-minute delay (as reported in Table 4.9). Thus, accounting for the 'patient-in-the-loop' medication intake delay consideration in policy training allows for improved performance when *testing* the policies in a realistic setting.

4.4 Discussion

We develop a data-driven, 'patient-in-the-loop' RL-based framework for leveraging high-frequency wearable sensor data to develop personalized medication strategies for PD patients. We incorporate realistic constraints on medication administration, including minimum dosing intervals, maximum daily dosages, and delays between medication recommendation and administration. We also develop a novel approach for incorporating multiple medications

Table 4.9: Robustness analysis results for medication administration delay. Table shows physician-updated regimens' expected cumulative reward from Table 4.4 alongside RL policies' expected cumulative reward under a stochastic medication delay.

Avg. Cumulative Reward (Physician)	**Avg. Cumulative Reward (RL)**
97.7	163.6

into the RL policy without leading to combinatorial explosion of the action space, allowing our method to generalize to larger medication sets in future studies. Lastly, we use our approach to identify candidates for switching to lower dosage frequencies that might improve adherence without sacrificing symptom control and candidates for advanced therapies (such as continuous-administration L-dopa pumps or DBS).

In our patient cohort, the RL policies learn to prescribe extended-release L-dopa (primarily Rytary) instead of L-dopa IR, while using lower overall daily LED than the physician-updated regimens. This suggests that the RL policies administer patients' allowed LED more efficiently throughout the day rather than simply recommending increases in each patient's dosages. The RL policies' reliance on L-dopa CR and Rytary yields more stable bloodstream L-dopa concentrations than physician-updated policies. Our case study visualizes how this leads to a gradual increase in patients' bloodstream L-dopa concentration (which leads to improved bradykinesia), rather than the swings in L-dopa concentrations induced from L-dopa IR. This reflects the clinical finding that PD symptoms can be exacerbated by large swings in bloodstream L-dopa concentrations [18], suggesting that the RL policies learn clinically valid treatment strategies. Note that, although patients' RL policies do not recommend simultaneous administration of multiple medications, our inclusion of multi-medication actions increases confidence in the learned policies and in the clinical sufficiency of administering one medication at a time.

Since physicians had access to patients' Visit 1 PKG data when prescribing patients' Visit 2 medication regimens, our RL policies' superior performance is not merely due to their access to PKG data. Rather, our approach synthesizes patients' wearable data in an objective and data-driven manner. The RL policies' superior performance is also robust to misspecifications in our L-dopa pharmacokinetic model. In addition, the RL policies also continue to outperform physician-updated regimens under a stochastic medication intake delay, in which patients take medication up to 20 minutes after it is recommended. We overall demonstrate that accounting for the 'patient-in-the-loop' consideration of medication intake delay when learning the RL policies provides improved performance in a realistic setting where patients' intake delays may be stochastic.

Our patient-specific approach to policy learning also allows us to predict patients' viability for extended dosing intervals or advanced therapies. By increasing the RL policies' minimum dosing interval, we identify patients who could switch to lower-frequency L-dopa dosing without sacrificing symptom control. Since patients are more likely to adhere to lower-frequency medication regimens, this approach offers considerable promise for promoting adherence among PD patients. By decreasing the RL policies' minimum dosing interval, our approach also identifies patients who may require high-frequency symptom management, which can be achieved through advanced therapies such as L-dopa pumps or DBS. Since identifying candidates for advanced therapies is time-consuming and requires heavy physician oversight, our approach can considerably improve the efficiency of PD care by helping physicians predict which patients will benefit the most from such therapies.

4.5 Conclusion and Future Works

In this study, we develop the first data-driven, 'patient-in-the-loop' framework for optimizing medication regimens using wearable sensor data. We incorporate realistic medication restrictions into our model and demonstrate that pairing wearable sensor data with RL yields high-performing medication policies that can improve patient outcomes.

Our study is limited in its relatively small sample size. Future work should continue to evaluate the utility of wearable-based RL medication policies in larger cohorts, which would allow for additional analyses such as patient clustering and phenotyping. We also evaluate our RL medication policies by simulating their performance in a statistical model learned from patient data; such policies will need to be validated in prospective pilot studies. While we address the issue of medication adherence by lowering the RL policies' administration frequency, future work will also need to evaluate real-world adherence rates and incorporate these rates into the modeling framework.

Although our RL-based policies did not recommend simultaneous administration of multiple medications, future research might examine the conditions under which such multi-medication actions would be optimal. Lastly, while wearables are capable of monitoring a wide array of patient symptoms and vital signs (e.g., sleep duration, blood oxygen level,

heart rate, etc.), some chronic conditions may require users to directly input information about their symptom severity. Future work might extend our approach to such contexts in which the RL-based medication policy must account for uncertainty or missingness in user-reported symptoms.

Bibliography

[1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org. 14

[2] Agarwal, A., Kakade, S., Lee, J., and Mahajan, G. (2019). On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *arXiv preprint 1908.00261*. 88, 98

[3] Ahmad, T., Lund, L., Rao, P., Ghosh, R., Warier, P., Vaccaro, B., and Desai, N. (2018). Machine learning methods improve prognostication, identify clinically distinct phenotypes, and detect heterogeneity in response to therapy in a large cohort of heart failure patients. *Journal of the American Heart Association*, 7(8):e008081. 33

[4] Aich, S., Youn, J., Chakraborty, S., Pradhan, P., Park, J., Park, S., and Park, J. (2020). A supervised machine learning approach to detect the on/off state in Parkinson's disease using wearable based gait signals. *Diagnostics*, 10(6):421. 71, 72

[5] Al-Khafajiy, M., Baker, T., Chalmers, C., Asim, M., Kolivand, H., Fahim, M., and Waraich, A. (2019). Remote health monitoring of elderly through wearable sensors. *Multimedia Tools and Applications*, 78(17):24681–24706. 71

[6] Alzheimer's Disease Neuroimaging Initiative (2005). 28

[7] Antipov, G., Baccouche, M., and Dugelay, J. (2017). *Face aging with conditional generative adversarial networks*, pages 2089–2093. IEEE. 3

[8] Atnekvist, I., Kragic, D., and Stork, J. (2019). *VPE: Variational policy embedding for transfer reinforcement learning*, pages 36–42. IEEE. 36, 37, 38, 50

[9] Ayer, T., Alagoz, O., and Stout, N. (2012). Or forum—a POMDP approach to personalize mammography screening decisions. *Operations Research*, 60(5):1019–1034. 24

[10] Barletta, J., DeYoung, J., McAllen, K., Baker, R., and Pendleton, K. (2008). Limitations of a standardized weight-based nomogram for heparin dosing in patients with morbid obesity. *Surgery for Obesity and Related Diseases*, 4(6):748–735. 42

[11] Baucum, M., Khojandi, A., and Vasudevan, R. (2020). Improving deep reinforcement learning with transitional variational autoencoders: A healthcare application. *IEEE Journal of Biomedical and Health Informatics*. ©2020 IEEE. Reprinted, with permission, from M. Baucum, A. Khojandi and R. Vasudevan,"Improving Deep Reinforcement Learning with Transitional Variational Autoencoders: A Healthcare Application," in IEEE Journal of Biomedical and Health Informatics, doi: 10.1109/JBHI.2020.3027443. iii, 23, 32, 33, 35, 41, 70

[12] Baum, L. and Petrie, T. (1966). Statistical inference for probabilistic functions of finite state Markov chains. *The Annals of Mathematical Statistics*, 37(6):1554–1563. 2, 11, 22, 29

[13] Belić, M., Bobić, V., Badža, M., Šolaja, N., urić-Jovičić, M., and Kostić, V. (2019). Artificial intelligence for assisting diagnostics and assessment of Parkinson's disease—A review. *Clinical Neurology and Neurosurgery*, 184:105442. 68

[14] Bica, I., Alaa, A., Lambert, C., and van der Schaar, M. (2021). From real-world patient data to individualized treatment effects using machine learning: current and future methods to address underlying challenges. *Clinical Pharmacology & Therapeutics*, 109(1):87–100. 38

[15] Blennow, K. (2004). Cerebrospinal fluid protein biomarkers for Alzheimer's disease. *NeuroRx*, 1(2):213–225]. 29

[16] Bowman, S., Vilnis, L., Vinyals, O., Dai, A., Jozefowicz, R., and Bengio, S. (2015). Generating sentences from a continuous space. *arXiv preprint 1511.06349*. 9

[17] Bridle, J. (1990). Alpha-nets: A recurrent 'neural' network architecture with a hidden Markov model interpretation. *Speech Communication*, 9(1):83–92. 23

[18] Brooks, D. (2008). Optimizing levodopa therapy for Parkinson's disease with levodopa/carbidopa/entacapone: Implications from a clinical and patient perspective. *Neuropsychiatric Disease and Treatment*, 4(1):39. 106

[19] Bussone, A., Stumpf, S., and O'Sullivan, D. (2015). The role of explanations on trust and reliance in clinical decision support systems. In *2015 International Conference on Healthcare Informatics*, pages 160–169. IEEE. 29

[20] Caelli, T., Guan, L., and Wen, W. (1999). Modularity in neural computing. *Proceedings of the IEEE*, 87(9):1497–1518. 23

[21] Chang, R., Lu, H., Yang, P., and Luarn, P. (2016). Reciprocal reinforcement between wearable activity trackers and social network services in influencing physical activity behaviors. *JMIR mHealth and uHealth*, 4(3):e84. 67

[22] Chen, C., Kehtarnavaz, N., and Jafari, R. (2014). A medication adherence monitoring system for pill bottles based on a wearable inertial sensor. In *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 4983–4986. IEEE. 67

[23] Chen, Y., Qin, X., Wang, J., Yu, C., and Gao, W. (2020). Fedhealth: A federated transfer learning framework for wearable healthcare. *IEEE Intelligent Systems*. 36, 50

[24] Choi, E., Biswal, S., Malin, B., Duke, J., Stewart, W., and Sun, J. (2017). Generating multi-label discrete patient records using generative adversarial network. *arXiv preprint 1703.06490*. 3

[25] Coleman, C., Limone, B., Sobieraj, D., Lee, S., Roberts, M., Kaur, R., and Alam, T. (2012). Dosing frequency and medication adherence in chronic disease. *Journal of Managed Care Pharmacy*, 18(7):527–539. 98

[26] Conti, E., Madhavan, V., Suche, F., Lehman, J., Stanley, K., and Clune, J. (2017). Improving exploration in evolution strategies for deep reinforcement learning via a population of novelty-seeking agents. *arXiv preprint arXiv:1712.06560*. 37

[27] Daneault, J., Carignan, B., Sadikot, A., Panisset, M., and Duval, C. (2013). Drug-induced dyskinesia in Parkinson's disease. Should success in clinical management be a function of improvement of motor repertoire rather than amplitude of dyskinesia? *BMC medicine*, 11(1):1–18. 85

[28] Daskalaki, E., Diem, P., and Mougiakakou, S. (2013). Personalized tuning of a reinforcement learning control algorithm for glucose regulation. In *2013 35th Annual international conference of the IEEE engineering in medicine and biology society (EMBC)*, pages 3487–3490. IEEE. 35, 67, 70, 72

[29] Dickey, D. and Pantula, S. (1987). Determining the order of differencing in autoregressive processes. *Journal of Business & Economic Statistics*, 5(4):455–461. 77

[30] Dudik, M., Langford, J., and Li, L. (2011). Doubly robust policy evaluation and learning. *arXiv preprint 1103.4601*. 41

[31] el Hassouni, A., Hoogendoorn, M., van Otterlo, M., and Barbabo, E. (2018). *Personalization of health interventions using cluster-based reinforcement learning*, pages 467–475. Springer. 36, 50, 67

[32] Estebanez, B., del Saz-Orozco, P., Rivas, I., Bauzano, E., Muñoz, V., and Garcia-Morales, I. (2012). *Maneuvers recognition in laparoscopic surgery: Artificial neural network and hidden Markov model approaches*, pages 1164–1169. IEEE. 23

[33] Fairbank, M. and Alonso, E. (2011). The local optimality of reinforcement learning by value gradients, and its relationship to policy gradient learning. *arXiv preprint 1101.0428*. 88, 98

[34] Favela, J., Cruz-Sandoval, D., Morales-Tellez, A., and Lopez-Nava, I. (2020). Monitoring behavioral symptoms of dementia using activity trackers. *Journal of Biomedical Informatics*, 109:103520. 70

[35] Fisher, C., Smith, A., and Walsh, J. (2018). Deep learning for comprehensive forecasting of Alzheimer's disease progression. *arXiv preprint 1807.03876*. 3

[36] Fohner, A., Greene, J., Lawson, B., Chen, J., Kipnis, P., Escobar, G., and Liu, V. (2019). Assessing clinical heterogeneity in sepsis through treatment patterns and machine learning. *Journal of the American Medical Informatics Association*, 26(12):1466–1477. 33

[37] Folstein, M., Folstein, S., and McHugh, P. (1975). Mini-mental state: A practical method for grading the cognitive state of patients for the clinician. *Journal of Psychiatric Research*, 12(3):189–198. 28

[38] Gabler, N., Ratcliffe, S., Wagner, J., Asch, D., Rubenfeld, G., Angus, D., and Halpern, S. (2013). Mortality among patients admitted to strained intensive care units. *American Journal of Respiratory and Critical Care Medicine*, 188(7):800–806. 32

[39] Gao, C., Sun, H., Wang, T., Tang, M., BN.I. ohnen, M. M., Herman, T., Giladi, N., Kalinin, A., and Spino, C. (2018). Model-based and model-free machine learning techniques for diagnostic prediction and classification of clinical outcomes in Parkinson's disease. *Scientific reports*, 8(1):1–21. 68

[40] Geraedts, V., KochM, M., Contarino, M., Middelkoop, H., Wang, H., van Hilten, J., Bäck, T., and Tannemaat, M. (2021). Machine learning for automated eeg-based biomarkers of cognitive impairment during deep brain stimulation screening in patients with Parkinson's disease. *Clinical Neurophysiology*. 69

[41] Gettings, E., Brush, K., Van Cott, E., and Hurford, W. (2006). Outcome of postoperative critically ill patients with heparin-induced thrombocytopenia: an observational retrospective case-control study. *Critical Care*, 10(6). 14

[42] Ghassmi, M., Richter, S., Eche, I., Chen, T., Danziger, J., and Celi, L. (2014). A data-driven approach to optimized medication dosing: A focus on heparin. *Intensive Care Medicine*, 40(9):1332–1339. 51

[43] Ghazi, M., Nielson, N., Pai, A., Cardoso, M., Modat, M., Ourselin, S., and Sorensen, L. (2019). Training recurrent neural networks to incomplete data: Application to alzheimer's disease progression modeling. *Medical Image Analysis*, 53:39–46. 2

[44] Goodfellow, I., Pouget-Avadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). *Generative adversarial nets*, pages 2672–2680. 3

[45] Gottesman, O., Johansson, F., Komorowski, M., Faisal, A., Sontag, D., Doshi-Velez, F., and Celi, L. (2019). Guidelines for reinforcement learning in healthcare. *Nature medicine*, 25(1):16–18. 38

[46] Gottesman, O., Johansson, F., Meier, J., Dent, J., Lee, D., Srinivasan, S., and Yao, J. (2018). Evaluating reinforcement learning algorithms in observational health settings. *arXiv preprint 1805.12298*. 41

[47] Greig, S. and McKeage, K. (2016). Carbidopa/levodopa ER capsules (Rytary®, numient™): A review in Parkinson's disease. *CNS drugs*, 30(1):79–90. 78

[48] Griffiths, R., Kotschet, K., Arfon, S., Xu, Z., Johnson, W., Drago, J., Evans, A., Kempste, P., Raghav, S., and Horne, M. (2012). Automated assessment of bradykinesia and dyskinesia in Parkinson's disease. *Journal of Parkinson's Disease*, 2(1):47–55. 71

[49] Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. (2017). Reinforcement learning with deep energy-based policies. In *International Conference on Machine Learning*, pages 1352–1361. PMLR. 37

[50] Harmer, J., Gisslén, L., del Val, J., Holst, H., Bergdahl, J., Olsson, T., Sjöö, K., and Nordin, M. (2018). Imitation learning with concurrent actions in 3D games. In *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 1–8. IEEE. 86

[51] Hauser, R., Banisadr, G., Vuong, K., Freilich, D., Fisher, S., and D'Souza, R. (2021). Real-world experience with carbidopa-levodopa extended-release capsules (Rytary®): Results of a nationwide dose conversion survey. *Parkinson's Disease*, 2021. 68

[52] Ho, J., Moghim, N., Becj, J., and Jolliff, J. (2018). Weight-based dosing vs. standard care nomogram for IV heparin. 42

[53] Horne, M., Kotschet, K., and McGregor, S. (2016). The clinical validation of objective measurement of movement in Parkinson's disease. In *CNS*, volume 2, pages 16–23. 71

[54] Hsu, A., Yao, H., Gupta, S., and Modi, N. (2015). Comparison of the pharmacokinetics of an oral extended-release capsule formulation of carbidopa-levodopa (IPX066) with immediate-release carbidopa-levodopa (Sinemet®), sustained-release carbidopa-levodopa (Sinemet® CR), and carbidopa-levodopa-entacapone (Stalevo®). *The Journal of Clinical Pharmacology*, 55(9):995–1003. 78, 102

[55] Jaques, N., Ghandeharioun, A., Shen, J., Ferguson, C., Lapedriza, A., Jones, N., Gu, S., and Picard, R. (2019). Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456*. 36, 41

[56] Jedynak, B., Liu, B., Lang, A., Gel, Y., and Prince, J. (2015). A computational method for computing an Alzheimer's disease progression score: Experiments and validation with the ADNI dataset. *Neurobiology of Aging*, 36:S178–S184. 41

[57] Johnson, A. E., Pollard, T. J., Shen, L., Li-wei, H. L., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., and Mark, R. G. (2016). MIMIC-III, a freely accessible critical care database. *Scientific data*, 3:160035. 5, 39

[58] Kandaswamy, C., Silva, L., Alexandre, L., and Santos, J. (2016). High-content analysis of breast cancer using single-cell deep transfer learning. *Journal of Biomolecular Screening*, 21(3):252–259. 36, 50

[59] Kang, S., Kang, J., Ko, K., Park, S., Mariani, S., and Weng, J. (2017). Validity of a commercial wearable sleep tracker in adult insomnia disorder patients and good sleepers. *Journal of Psychosomatic Research*, 97:38–44. 67, 70

[60] Karsoliya, S. (2012). Approximating number of hidden layer neurons in multiple hidden layer BPNN architecture. *International Journal of Engineering Trends and Technology*, 3(6):714–717. 9, 11

[61] Khojandi, A., Shylo, O., Mannini, L., Kopell, B., and Ramdhani, R. (2017). Stratifying Parkinson's patients with STN-DBS into high-frequency or 60 hz-frequency modulation using a computational model. *Neuromodulation: Technology at the Neural Interface*, 20(5):450–455. 68

[62] Kim, H., Kam, H., Lee, J., Yoo, S., Woo, K., Noh, J., and Yoo, S. (2013). *Monitoring for disease progression via mathematical time-series modeling: Actigraphy-based monitoring patients with depressive disorder*, pages 56–61. IEEE. 41

[63] Kim, Y., Suescun, J., Schiess, M., and Jiang, X. (2021). Computational medication regimen for Parkinson's disease using reinforcement learning. *Scientific Reports*, 11(1):1–9. 69

[64] Kingma, D., Mohamed, S., Rezende, D., and Welling, M. (2014). *Semi-supervised learning with deep generative models*, pages 3581–3589. 3

[65] Kingma, D. and Welling, M. (2014). Auto-encoding variational bayes. *ICLR*. 3, 6

[66] Komorowski, M., Celi, L., Badawi, O., Gordon, A., and Faisal, A. (2018). The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care. *Nature Medicine*, 24(11):1716–1720. 35

[67] Kravitz, R., Duan, N., and Braslow, J. (2004). Evidence-based medicine, heterogeneity of treatment effects, and the trouble with averages. *The Milbank Quarterly*, 82(4):661–687. 33

[68] Kuoppamäki, M., Korpela, K., Marttila, R., Kaasinen, V., Hartikainen, P., Lyytinen, J., Kaakkola, S., Hänninen, J., Löyttyniemi, E., and Kailajärvi, M. (2009). Comparison of pharmacokinetic profile of levodopa throughout the day between levodopa/carbidopa/entacapone and levodopa/carbidopa when administered four or five times daily. *European journal of clinical pharmacology*, 65(5):443–455. 98

[69] Lee, S., Daneault, J., Golabchi, F., Patel, S., Paganoni, S., Shih, L., and Bonato, P. (2015). A novel method for assessing the severity of levodopa-induced dyskinesia using wearable sensors. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 8087–8090. IEEE. 71, 72

[70] Leeman, A., O'Neill, C., Nicholson, P., Deshmukh, A., Denham, M., Royston, J., Dobbs, R., and Dobbs, S. (1987). Parkinson's disease in the elderly: Response to and optimal

spacing of night time dosing with levodopa. *British Journal of Clinical Pharmacology*, 24(5):637–643. 67

[71] LeMoyne, R., Mastroianni, T., Whiting, D., and Tomycz, N. (2020). Application of deep learning to distinguish multiple deep brain stimulation parameter configurations for the treatment of Parkinson's disease. In *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1106–1111. IEEE. 69

[72] Liao, P., Greenewald, K., Klasnja, P., and Murphy, S. (2020). Personalized heartsteps: A reinforcement learning algorithm for optimizing physical activity. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(1):1–22. 67, 70, 72

[73] Lin, C., Zhangy, Y., Ivy, J., Capan, M., Arnold, R., Huddleston, J., and Chi, M. (2018). *Early diagnosis and prediction of sepsis shock by combining static and dynamic inforamtion using convolutional-LSTM*, pages 219–228. IEEE. 2

[74] Liu, Y., Li, S., Li, F., Song, L., and Rehg, J. (2015). Efficient learning of continuous-time hidden Markov models for disease progression. *Advances in Neural Information Processing Systems*, pages 3600–3608. 2, 22

[75] Liu, Y., Ramachandran, P., Liu, Q., and Peng, J. (2017). Stein variational policy gradient. *Proceedings of Conference on Uncertainty in Artificial Intelligence*. 36, 37, 45, 51

[76] Mark, D., Cowper, P., Berkowitz, S., Davidson-Ray, L., DeLong, E., Turpie, A., and Cohen, M. (1998). Economic assessment of low-molecular-weight heparin (enoxaparin) versus unfractionated heparin in acute coronary syndrome patients. *Circulation*, 97(17):1702–1707. 44, 51

[77] Mawulolo, M., Beltzer, M., Cai, L., Boukhechba, M., Teachman, B., and Barnes, L. (2020). Offline contextual multi-armed bandits for mobile health interventions: A case study on emotion regulation. In *Fourteenth ACM Conference on Recommender Systems*, pages 249–258. 35

[78] Mayo Clinic (n.d.). Carbidopa and levodopa (oral route). https://www.mayoclinic.org/drugs-supplements/carbidopa-and-levodopa-oral-route/proper-use/drg-20095211. 77

[79] McLaughlin, K., Rimsans, J., Sylvester, K., Fanikos, J., Dorfman, D., Senns, P., Goldhaber, S., and Connors, J. (2019). Evaluation of antifactor-xa heparin assay and activated partial thromboplastin time values in patients on therapeutic continuous infusion unfractionated heparin therapy. *Clinical and Applied Thrombosis/Hemostasis*, 25. 6, 19, 51

[80] Minhas, S., Khanum, A., Riaz, F., Khan, S., and Alvi, A. (2017). Predicting progression from mild cognitive impairment to alzheimer's disease using autoregressive modeling of longitudinal and multimodal biomarkers. *IEEE Journal of Biomedical and Health Informatics*, 22(3):818–825. 41

[81] Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint 1411.1784*. 3

[82] Mittur, A., Gupta, S., and Modi, N. (2017). Pharmacokinetics of Rytary®, an extended-release capsule formulation of carbidopa–levodopa. *Clinical Pharmacokinetics*, 56(9):999–1014. 78, 102

[83] Mnih, V., Badia, A., Mirza, M., Graves, A., Lillicrap, T., Harley, T., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. *International Conference on Machine Learning*, pages 1928–1937. 2, 13, 33, 41, 82

[84] Monroe, T. and Carter, M. (2012). Using the Folstein mini mental state exam (MMSE) to explore methodological issues in cognitive aging research. *European Journal of Ageing*, 9(3):265–274. 28, 31

[85] Morgante, L., Morgante, F., Moro, E., Epifanio, A., Girlanda, P., Ragonese, P., Antonini, A., Barone, P., Bonuccelli, U., and Contarino, M. (2007). How many Parkinsonian patients are suitable candidates for deep brain stimulation of subthalamic nucleus? Results of a questionnaire. *Parkinsonism & Related disorders*, 13(8):528–531. 69

[86] Morris, J. (1993). Clinical dementia rating: Current version and scoring rules. *Neurology*, 43:2412–2414. 29

[87] Mostafa, S., Mustapha, A., Mohammed, M., Hamed, R., Arunkumar, N., Ghani, M. A., Jaber, M., and Khaleefah, S. (2019). Examining multiple feature evaluation and classification methods for improving the diagnosis of Parkinson's disease. *Cognitive Systems Research*, 54:90–99. 68

[88] Nahab, F., Abu-Hussain, H., and Moreno, L. (2019). Evaluation of clinical utility of the Personal Kinetigraph® in the management of Parkinson's disease. *Advances in Parkinson's Disease*, 8(3):42–61. xv, 71, 73, 74, 76, 77, 85

[89] Nahum-Shani, I., Hekler, E., and Spruijt-Metz, D. (2015). Building health behavior models to guide the development of just-in-time adaptive interventions: A pragmatic framework. *Health Psychology*, 34(S):1209. 67

[90] Nahum-Shani, I., Smith, S., Spring, B., Bonnie, J., Collins, L., Witkiewitz, K., Tewari, A., and Murphy, S. (2018). Just-in-time adaptive interventions (JITAIs) in mobile health: Key components and design principles for ongoing health behavior support. *Annals of Behavioral Medicine*, 52(6):446–462. 67

[91] National Institute of Neurological Disorders and Stroke (n.d.). Parkinson's disease: Hope through research. https://www.ninds.nih.gov/Disorders/Patient-Caregiver-Education/Hope-Through-Research/{P}arkinsons-Disease-Hope-Through-Research. Last accessed 4/7/2021. 82

[92] Negoescu, D., Bimpikis, K., Brandeau, M., and Iancu, D. (2018). Dynamic learning of patient response types: An application to treating chronic diseases. *Management science*, 64(8):3469–3488. 34

[93] Nemati, S., Ghassemi, M., and Clifford, G. (2016). Optimal medication dosing from suboptimal clinical examples: A deep reinforcement learning approach. *38th Annual International Conference of IEEE Engineering in Medicine and Biology Society*, pages 2978–2981. 1, 2, 6, 13, 17, 22, 23, 32, 33, 35, 39, 43, 44, 70, 72

[94] Ng, K., Sun, J., Hu, J., and Wang, F. (2015). *Personalized predictive modeling and risk factor identification using patient similarity*, pages 132–136. 37

[95] Nutt, J., Woodward, W., Hammerstad, J., Carter, J., and Anderson, J. (1984). The "on–off" phenomenon in Parkinson's disease: Relation to levodopa absorption and transport. *New England Journal of Medicine*, 310(8):483–488. 67

[96] Nyholm, D., Odin, P., Johansson, A., Chatamra, K., Locke, C., Dutta, S., and Othman, A. (2013). Pharmacokinetics of levodopa, carbidopa, and 3-o-methyldopa following 16-hour jejunal infusion of levodopa-carbidopa intestinal gel in advanced Parkinson's disease patients. *The AAPS Journal*, 15(2):316–323. 78

[97] Ossig, C., Gandor, F., Bosredon, C., Fauser, M., Reichmann, H., and Horne, M. (2015). Correlation of objective measurement of motor states using a kinetograph and patient diaries in advanced Parkinson's disease. *PLos One*. 71

[98] Pahwa, R., Isaacson, S., Torres-Russotto, D., Nahab, F., Lynch, P., and Kotschet, K. (2018). Role of the Personal Kinetigraph in the routine clinical assessment of Parkinson's disease: Recommendations from an expert panel. *Expert Review of Neurotherapeutics*, 18(8):669–680. 85

[99] Pahwa, R. and Lyons, K. (2017). Outpatient titration of carbidopa/levodopa enteral suspension (duopa). *International Journal of Neuroscience*, 127(5):459–465. 68

[100] Parbhoo, S., Bogojeska, J., Zazzi, M., Roth, V., and Doshi-Velez, F. (2017). Combining kernel and model based learning for hiv therapy selection. *AMIA Summits on Translational Science Proceedings*, pages 239–248. 1, 2, 33, 36, 37, 50, 80

[101] Parisi, L., RaviChandran, N., and Manaog, M. (2018). Feature-driven machine learning to improve early diagnosis of Parkinson's disease. *Expert Systems with Applications*, 110:182–190. 68

[102] Parkinson's Disease Foundation (2019). Statistics on Parkinson's. https://www.{P}arkinson.org/Understanding-{P}arkinsons/Statistics. Last accessed 7/10/19. 66

[103] Paula, M. D., Acosta, G., and Martínez, E. (2015). On-line policy learning and adaptation for real-time personalization of an artificial pancreas. *Expert Systems with Applications*, 42(4):2234–2255. 70, 72

[104] Pedrosa, T., Vasconcelos, F., Medeiros, L., and Silva, L. (2018). Machine learning application to quantify the tremor level for Parkinson's disease patients. *Procedia Computer Science*, 138:215–220. 68

[105] Pew Research (2020). About one-in-five Americans use a smart watch or fitness tracker. https://www.pewresearch.org/fact-tank/2020/01/09/about-one-in-five-americans-use-a-smart-watch-or-fitness-tracker/. Last accessed 2/27/21. 67

[106] Pineau, J., Guez, A., Vincent, R., Panuccio, G., and Avoli, M. (2009a). Treating epilepsy via adaptive neurostimulation: A reinforcement learning approach. *International Journal of Neural System*, 19(4):227–240. 35

[107] Pineau, J., Guez, A., Vincent, R., Panuccio, G., and Avoli, M. (2009b). Treating epilepsy via adaptive neurostimulation: A reinforcement learning approach. *International Journal of Neural Systems*, 19(4):227–240. 70, 72

[108] Poudyal, A., van Heerden, A., Hagaman, A., Maharjan, S., Byanjankar, P., Subba, P., and Kohrt, B. (2019). Wearable digital sensors to identify risks of postpartum depression and personalize psychological treatment for adolescent mothers: Protocol for a mixed methods exploratory study in rural Nepal. *JMIR Research Protocols*, 8(9):e14734. 67, 70

[109] Prasad, N., Cheng, L., Chivers, C., Draugelis, M., and Engelhardt, B. (2017a). A reinforcement learning approach to weaning of mechanical ventilation in intensive care units. *arXiv preprint 1704.06300*. 32, 35

[110] Prasad, N., Cheng, L., CHIVers, C., Draugelis, M., and Engelhardt, B. (2017b). A reinforcement learning approach to weaning of mechanical ventilation in intensive care units. *arXiv preprint 1704.06300*. 70, 72

[111] Raghu, A., Komorowski, M., Celi, L., Szolovits, P., and Ghassemi, M. (2017). Continuous state-space models for optimal sepsis treatment- A deep reinforcement learning approach. *arXiv preprint 1705.08422*. 1, 32, 35, 70, 72

[112] Rajapakshe, T., Rana, R., and Khalifa, S. (2021). A novel policy for pre-trained deep reinforcement learning for speech emotion recognition. *arXiv preprint arXiv:2101.00738*. 36

[113] Ramdhani, R., Khojandi, A., Shylo, O., and Kopell, B. (2018). Optimizing clinical assessments in Parkinson's disease through the use of wearable sensors and data driven modeling. *Frontiers in Computational Neuroscience*, 12:72. 68

[114] Reich, D., Yanakakis, M., Vela-Cantos, F., DePerio, M., and Jacobs, E. (1993). Comparison of bedside coagulation monitoring tests with standard laboratory tests in patients after cardiac surgery. *Anesthesia and Analgesia*, 77(4):673–679. 15, 54

[115] Rezaei, M., Harmuth, K., Gierke, W., Kellermeier, T., Fischer, M., Yang, H., and Meinel, C. (2017). *A conditional adversarial network for semantic segmentation of brain tumor*, pages 241–252. Springer. 3

[116] Riachi, E., Mamdani, M., Fralick, M., and Rudzicz, F. (2021). Challenges for reinforcement learning in healthcare. *arXiv preprint arXiv:2103.05612*. 38

[117] Rodriguez, M., Lera, G., Vaamonde, J., Luquin, M., and Obeso, J. (1994). Motor response to apomorphine and levodopa in asymmetric Parkinson's disease. *Journal of Neurology, Neurosurgery & Psychiatry*, 57(5):562–566. 84

[118] Rosin, A., Devereux, D., Eng, N., and Calne, D. (1979). Parkinsonism with 'on-off' phenomena: Intravenous treatment with levodopa after major abdominal surgery. *ArcHIVes of Neurology*, 36(1):32–34. 84

[119] Rydén, T. (2008). Em versus Markov chain monte carlo for estimation of hidden Markov models: A computational perspective. *Bayesian Analysis*, 3(4):659–688. 22

[120] Salmanpour, M., Shamsaei, M., Saberi, A., Klyuzhin, I., Tang, J., Sossi, V., and Rahmim, A. (2020). Machine learning methods for optimal prediction of motor outcome in Parkinson's disease. *Physica Medica*, 69:233–240. 68

[121] Samala, R., Chan, H., Hadjiiski, L., Helvie, M., Cha, K., and Ritcher, C. (2017). Multi-task transfer learning deep convolutional neural network: Application to computer-aided diagnosis of breast cancer on mammograms. *Physics in Medicine and Biology*, 62(23):8894–8908. 36, 50

[122] Saria, S. (2018). Individualized sepsis treatment using reinforcement learning. *Nature Medicine*, 24(11):1641–1642. 70

[123] Sathe, A., Tuite, P., Chen, C., Ma, Y., Chen, W., Cloyd, J., Low, W., Steer, C., Lee, B., and Zhu, X. (2020). Pharmacokinetics, safety, and tolerability of orally administered ursodeoxycholic acid in patients with Parkinson's disease— A pilot study. *The Journal of Clinical Pharmacology*, 60(6):744–750. 78

[124] Schade, S., Mollenhauer, B., and Trenkwalder, C. (2020). Levodopa equivalent dose conversion factors: An updated proposal including opicapone and safinamide. *Movement Disorders Clinical Practice*, 7(3):343. 78

[125] Schneider, R. and Biglan, K. (2017). The promise of telemedicine for chronic neurological disorders: The example of Parkinson's disease. *The Lancet Neurology*, 16(7):541–551. 66

[126] Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464. 11

[127] Shamir, R., Dolber, T., Noecker, A., Walter, B., and McIntyre, C. (2015). Machine learning approach to optimizing combined stimulation and medication therapies for Parkinson's disease. *Brain Stimulation*, 8(6):1025–1032. 69

[128] Silver, D. and Trosch, R. (2016). Physicians' experience with Rytary (carbidopa and levodopa) extended-release capsules in patients who have Parkinson disease. *Neurology*, 86(14 Supplement 1):S25–S35. 68

[129] Singh, R., Bush, E., Hidecker, M., Carrico, C., and Sundin, S. (2020). Considering health care needs in a rural Parkinson disease community. *Progress in Community Health Partnerships: Research, Education, and Action*, 14(1):15–28. 66

[130] Sohn, K., Lee, H., and Yan, X. (2015). Learning structured output representation using deep conditional generative models. *Advances in Neural Information Processing Systems*, pages 3483–3491. 3

[131] Stanculescu, I., Williams, C., and Freer, Y. (2013). Autoregressive hidden Markov models for the early detection of neonatal sepsis. *IEEE Journal of Biomedical and Health Informatics*, 18(5):1560–1570. 22, 24

[132] Stocchi, F., Quinn, N., Barbato, L., Patsalos, P., O'Connel, M., Ruggieri, S., and Marsden, C. (1994). Comparison between a fast and a slow release preparation of levodopa and a combination of the two: A clinical and pharmacokinetic study. *Clinical Neuropharmacology*, 17(1):38–44. 84

[133] Taylor, M. and Stone, P. (2009). Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(7). 36

[134] Tejedor, M., Woldaregay, A., and Godtliebsen, F. (2020). Reinforcement learning application in diabetes blood glucose control: A systematic review. *Artificial Intelligence in Medicine*, page 101836. 67, 89

[135] Tetrud, J., Nausieda, P., Kreitzman, D., Liang, G., Nieves, A., Duker, A., Hauser, R., Farbman, E., Ellenbogen, A., and Hsu, A. (2017). Conversion to carbidopa and levodopa extended-release (ipx066) followed by its extended use in patients previously taking controlled-release carbidopa-levodopa for advanced Parkinson's disease. *Journal of the Neurological Sciences*, 373:116–123. 84

[136] Ting, C., Sylvester, K., and Schurr, J. (2018). Time in the therapeutic range for assessing anticoagulation quality in patients receiving continuous unfractionated heparin. *Clinical and Applied Thrombosis/Hemostasis*, 24(9):178–181. 39, 43

[137] Tomkins, S., Liao, P., Klasnja, P., Yeung, S., and Murphy, S. (2020). Rapidly personalizing mobile health treatment policies with limited data. *arXiv preprint 2002.09971*. 70, 72

[138] Tomlinson, C., Stowe, R., Patel, S., Rick, C., Gray, R., and Clarke, C. (2010). Systematic review of levodopa dose equivalency reporting in Parkinson's disease. *Movement disorders*, 25(15):2649–2653. 78

[139] U.S. Food and Drug Administration (2008a). Sinemet. https://www.accessdata.fda.gov/drugsatfda_docs/label/2008/017555s069lbl.pdf. 84

[140] U.S. Food and Drug Administration (2008b). Sinemet CR. https://www.accessdata.fda.gov/drugsatfda_docs/label/2008/019856s025lbl.pdf. 84

[141] U.S. Food and Drug Administration (2015). Rytary. https://www.accessdata.fda.gov/drugsatfda_docs/label/2015/203312s000lbl.pdf. 84

[142] Utomo, C., Kurniawati, H., Li, X., and Pokharel, S. (2019). *Personalized medicine in critical care using Bayesian reinforcement learning*, pages 648–657. Springer. 36, 37, 38, 50

[143] Walker, J., Doersch, C., Gupta, A., and Herbert, M. (2016). An uncertain future: Forecasting from static images using variational autoencoders. *European Conference on Computer Vision*, pages 835–851. 3

[144] Wan, K., Maszczyk, T., See, A., Dauwels, J., and King, N. (2019). A review on microelectrode recording selection of features for machine learning in deep brain stimulation surgery for Parkinson's disease. *Clinical Neurophysiology*, 130(1):145–154. 69

[145] Wang, L., Zhang, W., He, X., and Zha, H. (2018). *Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation*, pages 2447–2456. 32

[146] Watts, J., Khojandi, A., Shylo, O., and Ramdhani, R. (2020a). Machine learning's application in deep brain stimulation for Parkinson's disease: A review. *Brain Sciences*, 10(11):809. 69

[147] Watts, J., Khojandi, A., Vasudevan, R., Nahab, F., and Ramdhani, R. (2021). Predicting Parkinson's disease medication regimen using sensor technology. *Preprint.* 68, 71

[148] Watts, J., Khojandi, A., Vasudevan, R., and Ramdhani, R. (2020b). Optimizing individualized treatment planning for Parkinson's Disease using deep reinforcement learning. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE. 35

[149] Watts, J., Khojandi, A., Vasudevan, R., and Ramdhani, R. (2020c). Optimizing individualized treatment planning for Parkinson's disease using deep reinforcement learning. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 5406–5409. IEEE. 68, 69

[150] Weng, W., Gao, M., He, Z., Yan, S., and Szolovits, P. (2017). Representation and reinforcement learning for personalized glycemic control in septic patients. *arXiv preprint 1712.00564*. 35

[151] Wessels, T. and Omlin, C. (2000). Refining hidden Markov models with recurrent neural networks. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*, volume 2, pages 271–276. 23

[152] Wierstra, D., Foerster, A., Peters, J., and Schmidhuber, J. (2007). *Solving deep memory POMDPs with recurrent policy gradients*, pages 697–706. Springer. 2

[153] Wilcox, M., Harrison, D., Patel, A., and Rowan, K. (2020). Higher ICU capacity strain is associated with increased acute mortality in closed ICUs. *Critical Care Medicine*, 48(5):709–716. 32

[154] Willis, A., Schootman, M., Evanoff, B., Perlmutter, J., and Racette, B. (2011). Neurologist care in Parkinson disease: A utilization, outcomes, and survival study. *Neurology*, 77(9):851–857. 66

[155] Wroge, T., Özkanca, Y., Demiroglu, C., Si, D., Atkins, D., and Ghomi, R. (2018). Parkinson's disease diagnosis using machine learning and voice. In *2018 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, pages 1–7. IEEE. 68

[156] Yang, J., Petersen, B., Zha, H., and Faissol, D. (2019). Single episode policy transfer in reinforcement learning. *arXiv preprint arXiv:1910.07719*. 36, 37, 38, 50

[157] Yao, J., Killian, T., Konidaris, G., and Doshi-Velez, F. (2018). *Direct policy transfer via hidden parameter Markov decision processes*. 36, 37, 50

[158] Yu, C., Ren, G., and Dong, Y. (2020). Supervised-actor-critic reinforcement learning for intelligent mechanical ventilation and sedative dosing in intensive care units. *BMC Medical Informatics and Decision Making*, 20(3):1–8. 35, 70, 72

[159] Yu, W., Tan, J., Liu, C., and Turk, G. (2017). *Preparing for the unknown: Learning a universal policy with online system identification*. 36, 37, 50

[160] Zanotti, C., Rotiroti, M., Sterlacchini, S., Cappellini, G., Fumagalli, L., Stefania, G., Nannucci, M., Leoni, B., and Bonomi, T. (2019). Choosing between linear and nonlinear models and avoiding overfitting for short and longer term groundwater level forecasting in a linear system. *Journal of Hydrology*, 578. 41

[161] Zhang, C., Yu, Y., and Zhou, Z. (2018). Learning environmental calibration actions for policy self-evolution. In *IJCAI*, pages 3061–3067. 98

[162] Zhang, K., Koppel, A., Zhu, H., and Basar, T. (2020). Global convergence of policy gradient methods to (almost) locally optimal policies. *SIAM Journal on Control and Optimization*, 58(6):3586–3612. 88

[163] Zhou, Z., Wang, Y., Mamani, H., and Coffey, D. (2019). How do tumor cytogenetics inform cancer treatments? Dynamic risk stratification and precision medicine using multi-armed bandits. *Preprint*. 1, 2, 22, 33, 34, 80

[164] Zhu, F., Guo, J., Xu, Z., Liao, P., Yang, L., and Huang, J. (2018). Group-driven reinforcement learning for personalized mhealth intervention. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 590–598. Springer. 67, 70, 72

Vita

Matt Baucum graduated from Pepperdine University in 2016 with a Bachelor of Arts in Psychology. He then received a Master of Arts in Psychology from the University of Southern California in 2018, with an emphasis in Quantitative Psychology. He then began his PhD in Industrial Engineering at the University of Tennessee, graduating in 2021. His doctoral research focuses on the application of sequential decision making models and reinforcement learning in healthcare. His research interests include machine learning, Markov decision processes, and model-based systems engineering.