Representation Learning for Reinforcement Learning in Healthcare

Taylor W. Killian TWKILLIAN@CS.TORONTO.EDU
University of Toronto, Vector Institute

Jayakumar Subramanian JAYAKUMAR.SUBRAMANIAN@GMAIL.COM
Adobe Research India

Mehdi Fatemi MEHDI.FATEMI@MICROSOFT.COM
Microsoft Research

Marzyeh Ghassemi MARZYEH@CS.TORONTO.EDU
University of Toronto, Vector Institute

Editor: Editor's name

Abstract

Reinforcement Learning (RL) has recently been applied to several problems in healthcare, with a particular focus on *offline* learning with observational data. RL relies on the use of latent states that embed sequential observations sufficient to approximately predict the next observation. The appropriate construction of such states in healthcare settings is an open question, as the variation in steady-state human physiology is poorly-understood. In this work, we evaluate several information encoding schemes using data from electronic health records (EHR), specifically focusing on the evaluation of *representations* that lead to optimal future *predictions* of patient observations. These representations are constrained to correlate with established acuity scores to ensure they contain clinically relevant information. We use observations from septic patients in the MIMIC-III dataset to evaluate several embedding approaches in two ways. First, we highlight the impact of representation space dimensionality on predicting the next set of observations. Second, we qualitatively investigate the learned state representations through their joint association with established acuity scores. Our experiments demonstrate that the best performing state representation learning approaches are recurrent neural architecture with adequate dimensionality; three of the five non-baseline models achieved best performance with state spaces at/below 64. State representations with good performance on observation prediction also correlate with clinical acuity scores. Our results demonstrate that a range of models are able to create coherent predictions of future observations, and the resulting states can enable RL approaches for healthcare tasks.

1. Introduction

Many problems in healthcare are a form of sequential decision making, e.g., clinical staff making decisions about the best "next step" in care (Ghassemi et al., 2019). Solving these problems is similar to finding an optimal decision making policy — requiring estimation and optimization of the cumulative effects of decisions over time — where the effects of

Figure 1: Visualization of patient health observations, via principal components analysis. Shown here are the first and final observations made of septic patients in the MIMIC-III dataset, colored by the SOFA score at the time of observation. On the left is a composite of these representations, with the right connecting corresponding beginning and terminal observations: blue lines signify patients who recovered, while red lines signify those that did not.

each decision may be non-deterministic. Previous work has modelled clinical tasks as controlled stochastic processes with rewards such as Markov decision processes (MDPs) (Puterman, 2014) or partially observable Markov decision processes (POMDPs) (Smallwood and Sondik, 1973). More recently, reinforcement learning (RL) has been proposed as a promising approach for finding an optimal policy for such processes from data, under certain assumptions (Gottesman et al., 2019). RL is a machine learning paradigm that associates sequential observations and actions with an eventual outcome, often through some learned latent state (Sutton and Barto, 2018). In a health setting, actions correspond to the treatments made by clinicians, and the desired outcome is context specific (e.g. patient survival in critical care, maintaining insulin levels within healthy range for diabetic patients, etc.).

Within critical care, sepsis has been well-studied with both a machine learning framing of supervised prediction (Futoma et al., 2017; Raith et al., 2017; Henry et al., 2015), and finding an optimal decision policy with RL (Komorowski et al., 2018; Saria, 2018; Peng et al., 2018). Sepsis is critical care syndrome (Vincent et al., 2013) characterized by disregulated infection response and organ dysfunction (Singer et al., 2016), and septic patients often have detailed data on a their vital signs, treatments, and outcome. While there are many proposed state construction approaches (Chang et al., 2019; Peng et al., 2018; Komorowski et al., 2018; Prasad et al., 2017; Raghu et al., 2017a,b), to our knowledge there has been no rigorous evaluation of what state representations lead to improved predictive performance on health tasks.

We anchor our evaluation in the offline RL POMDP framework, and focus on learning informative representations of the patient condition to facilitate better policy learning. We specifically target representations that are sufficient to capture the effect of selected actions on the next observation while maintaining information relevant to clinically established patient acuity (Silva et al., 2012; Hug and Szolovits, 2009). We use observations from the

MIMIC-III dataset (Johnson and Pollard, 2017) septic cohort defined in (Komorowski et al., 2018), and target the prediction of future observations.

We train six different embedding approaches that provide a decoding function for probabilistic next observation prediction: a recurrent neural network (RNN) autoencoder, Approximate Information State (AIS) (Subramanian and Mahajan, 2019), Decoupled Dynamics Module (DDM) (Zhang et al., 2018), Deep Signature Transforms (DST) (Kidger et al., 2019), latent ordinary differential equations (ODE) (Rubanova et al., 2019), and a baseline non-recurrent Autoencoder (AE). To ensure that the learned state representations are clinically relevant, we constrain representations to be correlated with three clinical acuity scores (Le Gall et al., 1993b; Vincent et al., 1996; Johnson and Mark, 2017) during the training process. We first empirically evaluate the prediction quality of the proposed methods by investigating the impact of state representation dimension on model accuracy in predicting the next observation. We follow this by evaluating how model error evolves when "rolling out" predicted observations for "k", or several, steps. Here, we allow the learned dynamics to cascade forward by up to 48 hours. A k-step roll-out complements forecasting the next observation, as the implicit quality of a policy depends on accounting for future transitions, not just the immediate one.

We find that recurrent architectures provide the best prediction of 1-step observations with higher dimension representations. In our work, a dimension between 64-128 usually provided sufficient capacity, and increases did not significantly improve predictions. Additionally, the learned state representations from these approaches maintain high Pearson correlation (Benesty et al., 2009) with the clinical acuity scores. Further, we find that by explicitly modeling the dynamics between patient observations, the learned state representations from the Latent ODE model are capable of stably predicting long sequences into the future (up to 48 hours in our experiments). This indicates that learned state representations encode meaningful associations between patient acuity and eventual outcome, a critical characteristic for the development of sequential decision-making policies. To the best of our knowledge, this study is the first rigorous evaluation of *learned state representations*, constructed through predicting sequential observations using real EHR data.

2. Background and related work

POMDPs require a state representation to be specified, often deriving from a history of observations and actions. This representation acts as a sufficient statistic for dynamic programming if it can adequately predict transition dynamics and rewards (Sondik, 1978; Pineau et al., 2003; Krishnamurthy, 2016). Prior work in state construction has ranged from concatenation of a finite number of consecutive observations (Mnih et al., 2013) to RNN approximation via use of the final layer to collectively embed a sequence of inputs (Silver et al., 2017). Other work has applied or developed RL solutions in the context of healthcare, but do not explicitly attempt to learn state representations. For reference, Table 6 summarizes these approaches, found in Appendix B. As discussed by Yu et al. (2019), state representation learning in healthcare is an open problem, an area of research that we seek to begin addressing in this work.

2.1. Related Work

State representation learning has a long history within RL as a primary means of making complex control tasks computationally tractable (Sutton et al., 1999). These approaches, sometimes referred to as state abstraction, attempt to meaningfully summarize the observation space to reduce the dimensions of observations to offer a more coherent representation for better planning and policy development (Abel et al., 2016). The concepts of state abstraction have motivated research in state representation learning to more fully separating feature extraction from policy learning (Jonschkowski and Brock, 2014; Lesort et al., 2018; Raffin et al., 2019). The goal of such approaches is to isolate the relevant features of the recorded observations in the state representation so as to provide more salient information to the policy learning algorithm on the effect that selected actions make on the underlying representation. Representation learning on supervised prediction tasks in healthcare settings, has mainly focused on finding static representations for electronic health records (EHR) (Choi et al., 2016; Sadati et al., 2018; Weng and Szolovits, 2019). Other healthcare representation learning approaches incorporate indicators of feature missingness and other underlying contextual variables (Agor et al., 2019; Fleming et al., 2019; Sharafoddini et al., 2019; Lipton et al., 2016).

To the best of our knowledge, such state representation learning approaches have not yet been evaluated for RL approaches in healthcare (Chang et al., 2019; Cheng et al., 2019; Komorowski et al., 2018; Prasad et al., 2017; Raghu et al., 2017a, 2018; Guez et al., 2008) (see Table 6). The most closely related works have presented a representation that accounts for patient health evolution alongside disease progression (Bai et al., 2018), and an representation of patient observation history using a RNN for input into a mixture of experts treatment policy (Peng et al., 2018). However, neither of these works provide any analysis justifying their specific choice of state representation, nor the parameters influencing it's construction. Our goal is to rigorously evaluate multiple state representation learning approaches for use in healthcare tasks under RL settings.

3. Data

In this paper we consider the problem of treatment of sepsis using data from the Medical Information Mart for Intensive Care (MIMIC-III) dataset (v1.4). MIMIC-III was sourced from the Beth Israel Deaconess Medical Center (BIDMC) in Boston, Massachusetts (Johnson et al., 2016; Johnson and Pollard, 2017), comprised of deidentified treatment records of patients. We follow the approach described by Komorowski et al. (2018) and the associated code repository[1] to extract a cohort of patients and their relevant observables for the study of sepsis treatment. This includes all ICU patients over 18 years of age who have some presumed onset of sepsis (following the Sepsis 3 criterion) during their initial encounter in the ICU after admission, with a duration of at least 12 hours. These criteria provide a cohort of 19,418 patients, among which there is an observed mortality rate just above 9%, where mortality is determined by patient expiration within 48h of the final observation. Observations are processed and aggregated into 4h windows with treatment decisions (ad-

1. The (Komorowski, 2018) repository can be found at https://github.com/matthieukomorowski/AI_Clinician

ministering fluids, vasopressors, or both) discretized into 5 volumetric categories. All data is normalized to zero-mean and unit variance following Komorowski et al. (2018).

The primary deviation we make from the Komorowski et al. cohort is that we remove all static, binary or derived columns that correspond to either demographic information, manually compiled values, or discrete events in a patient's care (e.g. age, gender, weight, acuity scores, etc.) as these quantities are largely unaffected by the selected action when considering the subsequent observation. We include a list of features in Table 1 with additional detail included in Section A of the Appendix. This creates a dataset of 33 features with an discrete categorical action space with 25 possible choices of combination between fluid and vasopressor amounts.

Table 1: Patient features used for learning state representations for predicting future observations

Glascow Coma Scale	Heart Rate	Sys. BP	Dia. BP
Mean BP	Respiratory Rate	Body Temp (C)	FiO2
Potassium	Sodium	Chloride	Glucose
INR	Magnesium	Calcium	Hemoglobin
White Blood Cells	Platelets	PTT	PT
Arterial pH	Lactate	PaO2	PaCO2
PaO2 / FiO2	Bicarbonate (HCO3)	SpO2	BUN
Creatinine	SGOT	SGPT	Bilirubin
Base Excess			

3.1. Acuity Scores

Patient acuity scores are used in clinical practice to estimate the severity a patient's illness, and have historically been used as a predictor of mortality (Silva et al., 2012). In order to constrain the learning of state representations we extract three acuity scores computed from the full patient observations from each 4h time step (Hug and Szolovits, 2009): Sepsis-related Organ Failure Assessment (SOFA) (Vincent et al., 1996), Simplified Acute Physiology Score II (SAPS II) (Le Gall et al., 1993b) and Oxford Acute Severity of Illness Score (OASIS) (Johnson and Mark, 2017). More information can be found in Section A.3.

4. Methods

In this section, we briefly describe relevant notation, each approach used in evaluations, model training details, and the constrained learning of state representations with clinically relevant acuity scores. We also describe how models are evaluated in predicting future patient observations.

4.1. Notation and Terminology

Let $\mathcal{D} = \{\tau_j\}_{j=1}^{n}$ denote the batch data of n observed patient trajectories obtained from clinical interactions using unknown, possibly different policies (e.g. various clinicians fol-

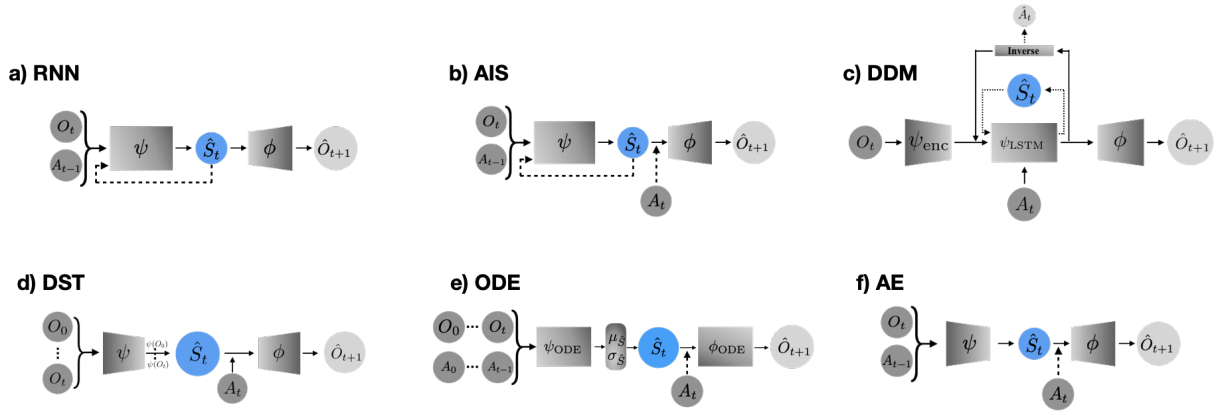

Figure 2: The architectures used to construct state representations via predicting future observations. **a)** The basic RNN autoencoder **b)** Approximate Information State (Subramanian and Mahajan, 2019) **c)** The Decoupled Dynamics Module from Zhang et al. (2018) **d)** The Deep Signature Transform (Kidger et al., 2019) inspired model **e)** The Latent ODE model from Rubanova et al. (2019) **f)** The non-recurrent Autoencoder. Aside from **d** DST and **e)** ODE, each model receives a single observation as input and generally uses a recurrent encoding function ψ to embed the observation and associate it with prior observations. See Table 2 for a comparative summarization

lowing their best judgement). We assume that this data is generated from a POMDP where each trajectory τ_j has a finite number of transitions m_j. Each transition between patient observations is a tuple with four entries $(O_{t,j}, A_{t,j}, R_{t,j}, O_{t+1,j})$, where $j \in \{1, \ldots, n\}$, $t \in \{1, \ldots, m_j\}$. Specific details for the observations O, actions A and rewards R are discussed in Section 3. For convenience, we drop the index j through the remainder of this paper unless the surrounding context is unclear.

Model predictions provide some approximation $\hat{O}_{t+1}$ to the true observation O_{t+1} by virtue of some intermediate learned state representation $\hat{S}_t$. Our objective is to learn a state construction function $\psi : \{O_{0:t}, A_{0:t-1}\} \mapsto \hat{S}_t$, $t \geq 1$, and $\hat{S}_t \in \hat{\mathcal{S}} \subset \mathbb{R}^{\hat{d}_s}$, where $\hat{d}_s$ denotes the dimension of the learned state representation $(\hat{S})$, which is a chosen hyper-parameter.

In addition to ψ, the approaches outlined in the next section also involve another function approximator: a dynamics predictor ϕ that involves predicting the next observation $\hat{O}_{t+1}$, at times conditioned on the current action A_t. Hence, the function $\phi : \hat{\mathcal{S}} \times \mathcal{A} \to \Delta(\mathcal{O})$, where $\Delta(x)$ denotes a probability distribution of x, estimates the conditional distribution of the next observation given the current state representation and action.

4.2. State representations models

We target five modeling approaches: a basic RNN Autoencoder (RNN), Approximate Information State (AIS), a Decoupled Dynamics Module (DDM), a variant on Deep Signature Transforms (DST), and finally the Latent ODE approach introduced by Rubanova et al. (2019) (ODE). These approaches are depicted in Figure 2). We compare these approaches to two simple baselines: a non-recurrent Autoencoder (AE) (Sec. C.6) and an untrained RNN (Rand. RNN). Detailed model descriptions are in the Appendix, Section C. A comparative overview of the features that differentiate each approach is presented in Table 2.

Table 2: Overview of approaches for state representation learning under evaluation

Approach	Recurrent	Additional context	Sequence as input	Num. Parameters
AE		✗		$27k - 76k$
AIS	✗	✗		$28k - 339k$
DDM	✗	✗		$6k - 1.25m$
DST		✗	✗	$47k - 256k$
ODE	✗	✗	✗	$2.1k - 358k$
Rand. RNN	✗			$26k - 337k$
RNN	✗			$26k - 337k$

We primarily consider approaches that encode sequential information in the state representation with an RNN through the function ψ, with the signature layer of DST being the notable exception. The state representation is then "decoded" with the function ϕ to predict the next observation. The loss function for all approaches, beside DDM and ODE[2], is the negative log likelihood of the true next observation within a unit variance mutlivariate Gaussian distribution with the predicted observation as it's mean. That is,

$$\mathcal{L}(O_{t+1}, \hat{O}_{t+1}) = -\sum_{i}^{d_o} \log \mathcal{N}(O_{t+1}^{(i)}; \mu_i, \sigma_i^2) \tag{1}$$

where $\mu_i = \hat{O}_{t+1}^{(i)}$, the ith feature of the prediction, and $\sigma_i^2 = 1$.

To ensure that the intermediate state representations $\hat{S}_t$ retains clinically relevant features, we constrain Equation 1 with the Pearson correlation between the state representation and a set of acuity scores derived from the patient observations. We utilize three independent acuity scores — SOFA, SAPS II and OASIS — and linearly combine the correlation coefficients to subtract from Equation 1. The complete objective function is then,

$$Loss = \mathcal{L}(O_{t+1}, \hat{O}_{t+1}) - \lambda\ \rho(\hat{S}_t) \tag{2}$$

where $\lambda\ \rho(\hat{S}_t) = \lambda_1\ \rho^{\text{SOFA}}(\hat{S}_t) + \lambda_2\ \rho^{\text{SAPS II}}(\hat{S}_t) + \lambda_3\ \rho^{\text{OASIS}}(\hat{S}_t)$ and the parameters λ are chosen to scale the correlation coefficients so that they appropriately affect the loss values. The general algorithmic procedure for learning the functions ϕ and ψ is demonstrated in Algorithm 1[3]. Experimental evaluation of these approaches is described in Section 5.

4.3. Next Observation Prediction

Consider a single trajectory τ_j, with recorded observations O_t and actions A_{t-1} (excepting the terminal observation). In the RNN, AE, AIS, and DDM, these are passed to the trained encoding function ψ to produce a state representation $\hat{S}_t$. This state representation

2. To ensure representative performance, we use the loss functions these models were designed with
3. It is our full intention to publish the code used to define, train and evaluate all approaches presented in this paper upon publication

Data: $\mathcal{D}$ with obs. trajectories τ_j, num. of train epochs n_e
Result: Functions ψ, ϕ that produce state representations $\hat{S}$ and predicted observations $\hat{O}$ respectively
init: randomly set parameters of ψ and ϕ **for** n_e *training epochs* **do**
  for *each trajectory* τ_j *in* $\mathcal{D}$ **do**
    init. $\hat{S}_0$ as a vector of zeros of dim. $\hat{d}_s$ **for** *each obs.* $O_t \in \tau_j$ **do**
      encode $(O_t, A_{t-1}, \hat{S}_{t-1})$ as $\hat{S}_t$ with ψ; decode $(\hat{S}_t, A_t)$ using ϕ to predict $\hat{O}_{t+1}$
      calculate $\mathcal{L}(O_{t+1}, \hat{O}_{t+1})$ and $\rho(\hat{S}_t)$; evaluate loss following Eqt. 2
    end
    Update parameters of ψ and ϕ according to accumulated loss over τ_j
  end
end

Algorithm 1: General training algorithm for learning state representations

is concatenated with the recorded action A_t and then passed to the trained decoder ϕ to produce a prediction of the next observation $\hat{O}_{t+1}$. In contrast, the ODE and DST approaches use full sequences of prior observations in a trajectory τ_j as input to the encoder (described in Sec. C.4 and C.5). These approaches also only can predict $\hat{O}_{t+1}$ after collecting two or more observations because the ODE and DST encoding functions do not not have semantic meaning when applied to single data points.

In each case, we compare predicted values $\hat{O}_{t+1}$ to the true next observation O_{t+1} with both the mean squared error (MSE), and a measure of relative error to account for variance in relative size of the features:

$$Rel.Err(\hat{O}_{t+1}, O_{t+1}) = \sum_{i}^{d_o} \frac{|\hat{O}_{t+1}^{(i)} - O_{t+1}^{(i)}|}{\max O^{(i)}} \tag{3}$$

4.4. K-step Roll Out Prediction

For each observation O_t in a trajectory τ_j, if a k-step sequence of observations $\{O_{t+1}, \ldots, O_{t+k}\}$ remain, we use the trained encoding and decoding functions ψ and ϕ to iteratively approximate—or roll out—this k-step sequence, $\{\hat{O}_{t+1}, \ldots, \hat{O}_{t+k}\}$ using only predicted observations after the first step as input to the encoding function ψ.

The MSE and relative error (Eq. 3) are measured using the kth prediction $\hat{O}_{t+k}$ in comparison with the true O_{t+k} observation. The errors of the kth prediction are averaged over all k-step sequences from each trajectory in the test dataset $\mathcal{D}_{\text{test}}$ using only the best models next observation prediction. This second task is designed to evaluate the consistency of future state representations with the data, leveraging the learned dynamics of each model, highlighting how each approach utilizes the embedded history in the state representation to infer future observations.

5. Experiments

We evaluate the six learning approaches on patient data in two quantitative tasks and two qualitative evaluations.

Data: $\mathcal{D}_{\text{test}}$ with obs. trajectories τ_j, trained functions ψ, ϕ
Result: Average MSE and Rel.Error
for *each trajectory* τ_j *in* $\mathcal{D}$ **do**
  init. $\hat{S}_0$ as a vector of zeros of dim. $\hat{d}_s$ **for** *each obs.* $O_t \in \tau_j$ **do**
    if $|\tau_j| < t + k$ **then**
      move to the next trajectory τ_{j+1}
    else
      $\hat{O}_t \leftarrow Ot$ **for** k *steps* **do**
        encode $(\hat{O}_t, A_{t-1}, \hat{S}_{t-1})$ as $\hat{S}_t$ with ψ; decode $(\hat{S}_t, A_t)$ using ϕ to predict $\hat{O}_{t+1}$
        {**Note:** Use $\hat{O}_{t+1}$ as input for the next step}
      end
      evaluate MSE and Relative error (Eq. 3) between $\hat{O}_{t+k}$ and O_{t+k}
    end
  end
end

Algorithm 2: General procedure for evaluating a k-step roll out using predicted observations

5.1. Quantitative Evaluations

We evaluate each state representation approach presented in Table 2 in two ways: (1) Impact of Model Dimension on Next-Step Prediction Error, and (2) K-Step Prediction Error. We compare the MSE and relative error of each model's predictions of future observations across various settings of $\hat{d}_s$, the dimension of the state representation, averaged over 5 independent random initializations of each model.

5.1.1. Impact of Model Dimension on Next-Step Prediction Error

We evaluate accuracy of models when predicting the next observation $\hat{O}_{t+1}$ from O_t and A_t, varying the dimension $\hat{d}_s$, the dimension of the learned state representation $\hat{S}_t$.

We test models except DST with $\hat{d}_s \in \{4, 8, 16, 32, 64, 128, 256\}$ to evaluate the appropriate information capacity needed in the state representation $\hat{S}$ to adequately predict future observations. With DST this dimension is jointly influenced by the size of the output from the pointwise encoding function ψ as well as the order N of the signature transform. To create models with comparably sized state representations we selected the output dimension of ψ_{DST} from the set $\{4, 5, 6, 7\}$ and constrained the choices for the order N of the signature transform to be either 1, 2 or 3. These choices produced a range of options for $\hat{d}_s$ between 5 and 400 following $|\text{Sig}^N| = \frac{d^{N+1}-1}{d-1}$.

5.1.2. K-Step Prediction Error

We evaluate the stability of the learned state representation $\hat{S}_t$ using predicted observations from the best model in Task (1) for k-step prediction up to $k = 12$. This corresponds to predicting a patient's condition after 48 hours taking prescribed actions every 4 hours. The errors of the kth prediction are averaged over all k-step sequences from each trajectory in the test dataset $\mathcal{D}_{\text{test}}$.

5.2. Qualitative Evaluations

We develop deeper intuition behind the learned state representations of patient observations in two ways.

5.2.1. Latent-To-Acuity Score Correlation

We evaluate the Pearson correlation between state representations of the test data and the corresponding acuity scores. We average the correlation coefficients of the state representations produced by the best performing model of each approach, ascertained by the quantitative evaluations described above. We report the separate coefficients for each of the three acuity scores used to constrain the learning process, averaged over 5 random initializations.

5.2.2. Visualizing Learned State Representations

We utilize principal component analysis (PCA) to help visualize the learned state representations. We use these projections to derive additional insight into associations between state representations and the acuity scores corresponding to the partially observed features used to generate them. We color these projections with SOFA score corresponding to the observation the state representations derive from to facilitate interpretation. These projections allow us to investigate how well the learned representations are associated and whether the underlying patient acuity may influence that association.

We fit each PCA projection using the entirety of the learned state representations from the test set for each approach, but only plot the first and last observation from each patient trajectory. Additionally, we also differentiate patient trajectories between those who recover from sepsis (and survive) and those that do not during this encounter in the ICU.

5.3. Model Training

We separate the data into a 70/15/15 train/validation/test split using stratified sampling to maintain the same proportions of each terminal outcome (survival or mortality), and no patient was repeated across the subsets of the data.

All models were trained for the same number of epochs, using a variety of learning rates and 5 random initializations (including baselines). The best performing model from each approach and choice of $\hat{d}_s$, evaluated on the validation set, was saved for use in the experiments presented in the following section. The final settings for each model architecture are provided in the Appendix, Section C.

6. Results

Figures 3 and 4 present results using MSE on the left and relative error on the right. The results are also plotted against the mean and normalized variance present in the test data (to compare with MSE and relative error, respectively). The mean errors from the best performing model from each approach are reported in Tables 3 and 4.

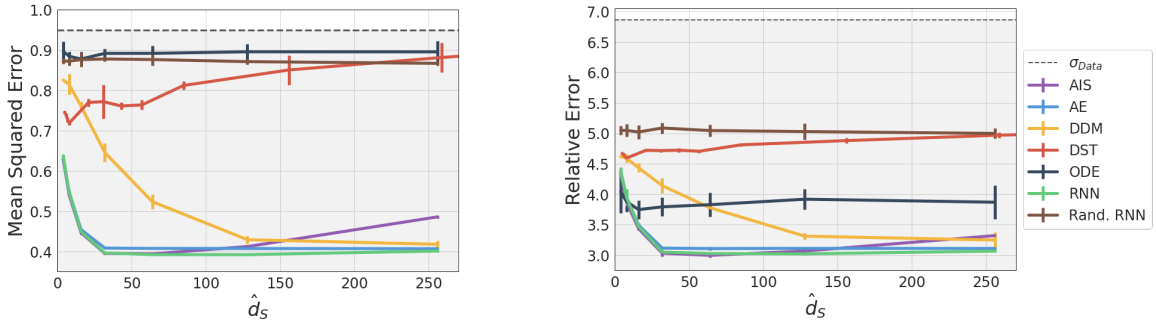

Figure 3: Results for next observation prediction using mean squared error (left) and relative error (Eq. 3, right) as a result of varying the dimension of the learned state representation. Also included is both the MSE and relative error of the test data, both denoted as σ_{Data} in their respective plots. Error bars correspond to twice the standard deviation of each model setting of $\hat{d}_s$ over 5 random initializations.

6.1. Impact of Model Dimension on Next-Step Prediction Error

Results for predicting the next observation, using a trajectory-based state representation are summarized in Figure 3 and Table 3. Model performance for all approaches (excepting DST) improves as $\hat{d}_s$ increases. It is notable that several approaches improve to near the same level of performance, hovering around ~ 0.4 in MSE (AE, AIS, RNN). The best performing models for each approach are reported in Table 3.

It is also demonstrated that each approach performs better than the randomized RNN baseline (again, excepting DST), an unsurprising yet satisfying result demonstrating that each approach learns a state representation that facilitates predicting the next observation rather than pathologically predicting the mean of the training data (an approach that achieves ≈ 0.9 in MSE).

Table 3: Results: Predicting the Next Observation

Approach	**Best MSE**	**Best Rel. Error**	$\hat{d}_s$
Rand. RNN	0.8669± 0.006	5.029 ± 0.130	256
AE	0.4067±0.001	3.11±0.03	256
AIS	0.3937± 0.003	2.99± 0.01	64
DDM	0.4175± 0.009	3.24± 0.13	256
DST	0.7187± 0.005	4.59 ± 0.02	8
ODE	0.8777± 0175.	3.88 ± 0.17	16
RNN	0.3917± 0.002	3.02 ± 0.02	128

Further, in view of the relative error (Fig. 3 (right)), all approaches have error within the natural variation of the test data (the line for σ_{Data} is the normalized sum of standard deviations of the test set, akin to Eqt. 3), signifying that the predictions being made are within the test data distribution.

6.2. Stability of K-Step Prediction Error

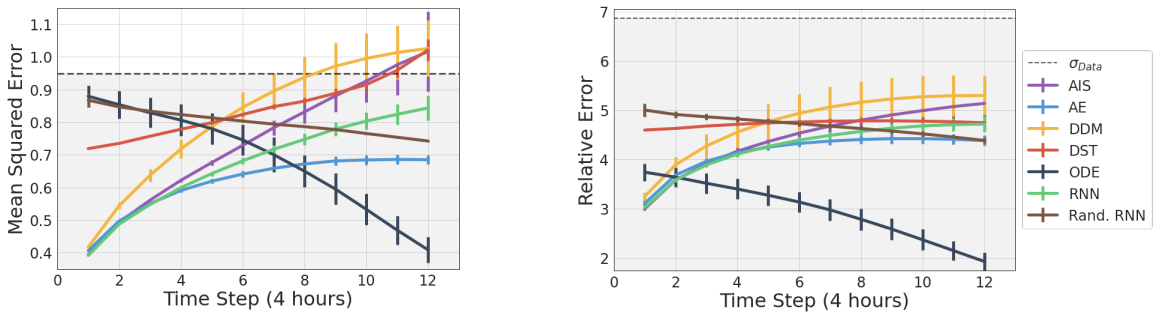

Figure 4: Results for k-step sequential predictions using using mean squared error (left) and relative error (Eq. 3, right). The MSE and relative error of the test data are denoted as σ_{Data} in their respective plots.. Error bars correspond to twice the standard deviation of each model setting of $\hat{d}_s$ over 5 random initializations. The results demonstrate that all approaches, excepting for the Latent ODE, error propagation from repeated predictions greatly reduces their accuracy. Conversely, the ODE approach leverages the inferred continuous dynamics to fit the prediction sequence to the trends found in the data, resulting in increasingly accurate predictions.

Results of a k-step roll out are summarized in Figure 4 and Table 4. This experiment helps to separate the approaches somewhat, helping to differentiate more complex approaches from the basic RNN approach. The primary motivation of this experimental task was to evaluate the stability of the learned state representation when used iteratively to predict several timesteps into the future.

While there is not as uniform of performance across the various approaches, the trend of increasing error as subsequent predictions are made is consistent. A slower rate of growth in prediction error as the sequence length increases provides some evidence that the learned state representation maintains some predictive stability in sequential prediction. Here the most simple models (RNN and the Autoencoder) appear to be outperforming the more complex recurrent methods such as AIS, DDM and DST. However, it is the performance of the Latent ODE approach along these lines that is particularly notable.

Table 4: Results: Prediction of K-step sequence (K=12)

Approach	**Best MSE**	**Best Rel. Error**	$\hat{d}_s$
Rand. RNN	0.7937± 0.002	6.438 ± 0.027	256
AE	0.6847±0.015	4.38±0.06	256
AIS	1.0161± 0.123	5.13± 0.27	64
DDM	1.0254± 0.259	5.29± 0.59	256
DST	1.0211± 0.401	4.74 ± 0.16	8
ODE	0.4087± 0.024	1.927 ± 0.18	16
RNN	0.8433± 0.114	4.715 ± 0.16	128

With repeated predictions, the ODE appears to capture the latent dynamics of the test trajectories, despite the error propagating from successive predictions. This may indicate that sampling from the variational distribution over the learned state representations while training lends to developing a significantly more robust prediction model

while limiting the effect of model error over time. Here the state representations may also be more informative of the underlying physiological processes present in patient response to prescribed treatments.

6.3. Latent-To-Acuity Score Correlation

We report the average correlation coefficient for the state representations learned by each approach, across 5 random initializations of each model, with each acuity score in Figure 5. Reviewing these results alongside Figure 3 and Table 3, shows that those approaches that have the best performance in the next-step prediction task appear (specifically, AE, AIS, DDM and RNN) to have learned state representations that are best correlated with the acuity scores.

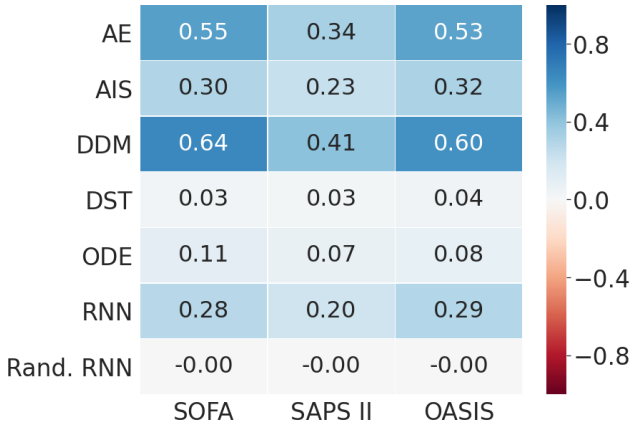

Figure 5: The average Pearson correlation coefficient between the state representations from each approach and the three acuity scores

Of the three acuity scores, each approach produces lowest correlation with SAPS II, likely due to the specific features of the septic patient cohort under investigation in this work not wholly correlating with a prediction of patient mortality as well as the relatively low mortality rate ($\sim 9\%$) in this cohort. Provided that the cohort is comprised of septic patients, it is intuitive that the representations from each approach are, in general, better correlated with SOFA on average.

6.4. Visualizing Learned State Representations

To vizualize the learned state representations we fit a PCA projection using the entirety of the learned state representations from the test set for each approach. However, we only use the first and last observation from each patient trajectory in Figure 6, also taking care to clearly identify representations that come from patients who either survived or died, for ease of interpretation (see figure caption for details).

From the insight gained in the previous section, we only present PCA projections colored by the SOFA score associated with each representation. We show the projections for RNN, AIS and AE in this section with the remaining approaches being included in the Appendix, Section D.

The learned state representations from the RNN and AIS are well separated by severity, with those corresponding to low SOFA scores falling to the left of the figure while more severe SOFA scores falling to the right (see Fig. 6(top, middle), on left); where the most severe representations correspond with the final observation made of patients who die from complications associated with sepsis. This is made more apparent when visually connecting the initial state representation of each patient trajectory to the final state representation. Represented as blue or red lines (corresponding to survival or death, respectively) we show

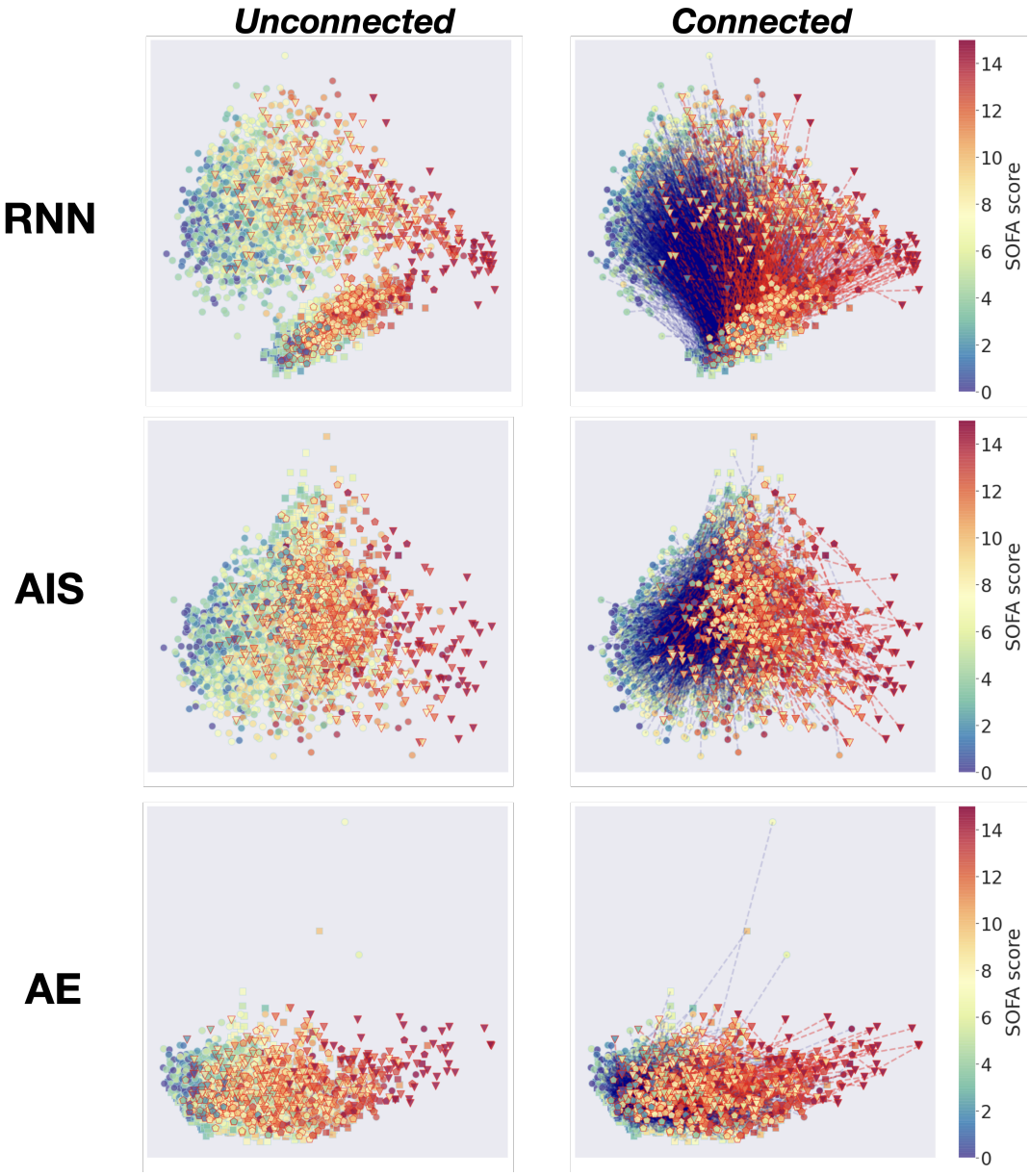

Figure 6: Representations of patient health, learned through an recurrent autoencoder (RNN), Approximate Information State (AIS) and the basic autoencoder (AE) (top-to-bottom, all other approaches are included in the Appendix, Section D), showing the first and final observations made of septic patients in the MIMIC-III dataset, colored by the SOFA score at the time of observation. On the left is a composite of these representations, with the right connecting corresponding beginning and terminal observations; blue lines signify patients who recovered, while red lines signify those that did not and ultimately succumbed to sepsis. Surviving patient trajectories begin with square markers and terminate with circle markers, outlined in white. Trajectories corresponding to patients who die begin with pentagon markers terminating in triangles, outlined in red.

that, in general, the learned state representations of patients in our cohort progress toward regions corresponding to distinct regions within the representation space associated with the outcome they experience.

Figure 6 (bottom) presents the learned state representations, via PCA, for the basic autoencoder (AE). Again, here the learned state representations are appropriately ordered according to severity yet they seem to be more tightly compressed. A consequence of this is that surviving patient representations are not as clearly separable from those representations that correspond to those that suffer death, perhaps signifying the inability of the non-recurrent model architecture to learn important characteristics of the observations. Similar general behavior can be observed in the PCA projected representations learned through DDM are seen in Figure 15 in the Appendix perhaps suggesting that model is perhaps overparametrized for learning state representations for predicting the next state in the specific cohort we used in this paper.

6.5. Discussion

The results presented in this section provide evidence that learned state representations can be helpful when predicting the next observation. By aggregating sequential information in the state representation, it is expected that the learned policies will benefit, as has been shown in the deep RL literature (Jaderberg et al., 2016). Better and more informative state representations, particularly those that embed some estimate of the transition dynamics between observations as well as clinically relevant information (as we have done through constraining the state representations to be correlated with acuity scores) will facilitate better policy learning in RL within healthcare settings.

Recurrent approaches learn appropriate state representations that contain relevant health information As we have demonstrated through the figures and tables presented in this section, it is apparent that recurrent based representation learning approaches present an intriguing option for state representation learning for RL in the context of healthcare. From the analyses we presented above, it is unclear whether there is any single best method for learning state representations. In our experiments, recurrent approaches AIS and RNN appear to learn informative state representations that are jointly associated with patient acuity and final outcome. This is in contrast with the autoencoder (AE) that has near equivalent performance yet the state representations it learns, while well correlated with acuity measures, do not provide identifiable separation between patient outcomes. DST performs the worst among the proposed approaches, possibly due to the loss of information by the encoding function ψ embedding the observation into too few dimensions. This choice appears to have prevented the signature transform from adequately expressing appropriate statistics of the observation sequence. Additionally, by constraining the number of elements of the signature to compare with the dimensions of state representations learned by other methods, it is possible that we may not have adequately represented the capabilities of this approach.

Appropriately parameterized models are critical when learning complex interaction behavior The best performing models for predicting the next observation are those with state representations with an adequate dimension $\hat{d}_s$, without having an over-parametrized model. Two such models we evaluated were the Latent ODE (ODE) and

Decoupled Dynamics Module (DDM), exceptionally expressive model architectures that have divergent performance on the tasks that we investigated. It is possible that our cohort was too narrow to fully leverage these models and appropriately train them. This may signal that—to learn adequate state representations for predicting the next observation—the models need to have enough capacity to encode the observed history and actions sufficient to represent the transition dynamics between observations, yet not too much so as to avoid overfitting.

Generalized insights As we seek to develop algorithmic tools to assist clinicians in complex decision making problems, it is imperative that we discover appropriate representations of patient observations that facilitate better policy development while still maintaining salient health information. This is most apparent in partially observable settings where the complete information about the patient condition is inaccessible. By aggregating observations over time we can detect trends in the features that we do observe while also associating prior observations to fill in for times when features may not be present. We have investigated this intuition in this paper, seeking to confirm that recurrent approaches enable appropriate state representation learning.

7. Conclusion

In this paper we present an analysis of several state representation learning methods for the prediction of future observations in healthcare settings. The proposed approaches were evaluated on prediction tasks designed to identify the best models and setting of learned state representations. We also performed a qualitative evaluation of the learned state representations and their correlation with observed patient acuity.

Following the findings presented in this paper we intend to develop and evaluate treatment policies for the MIMIC-III sepsis cohort briefly described in Section 3 using the learned representations encoded by the recurrent autoencoder and AIS. Further evaluation of this approach needs to be carried out in the development of such treatment policies as well as improving the accuracy of the auxiliary task of predicting the next observation. Additionally, it is necessary to more fully evaluate and interpret what the learned state representations encode in relation to the acuity scores they are constrained to be correlated with. It will be beneficial for the future use of these state representations to determine whether they embed trends in the data following the improving (or degrading) health of the patient beside only encoding features relevant for inferring the next observation.

Finally, for the use of these learned state representations in developing treatment policies, it is critical to evaluate whether they also contain some information about expected outcome when provided a suggested action. Such investigations and state representation learning will provide mechanisms by which we can better understand the cumulative effects of prescribed actions, chosen by following observed or learned policies. State representations and learned value functions used in this manner can enable the identification of suboptimal treatment decisions before they are executed. These inferences can reduce the incidence of malpractice and unneccessary negative outcomes in critical care with the potential to significantly reduce in-hospital mortality. These opportunities for learning optimal state representations for RL in healthcare offer an exciting new area of research that we antici-

pate being fruitful for establishing future advances in clinically relevant sequential decision making problems.

References

David Abel, David Hershkowitz, and Michael Littman. Near optimal behavior via approximate state abstraction. In *International Conference on Machine Learning*, pages 2915–2923, 2016.

Joseph Agor, Osman Y Özaltın, Julie S Ivy, Muge Capan, Ryan Arnold, and Santiago Romero. The value of missing information in severity of illness score development. *Journal of biomedical informatics*, 97:103255, 2019.

Tian Bai, Shanshan Zhang, Brian L Egleston, and Slobodan Vucetic. Interpretable representation learning for healthcare via capturing disease progression through time. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 43–51. ACM, 2018.

Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 37–40. Springer, 2009.

Chun-Hao Chang, Mingjie Mai, and Anna Goldenberg. Dynamic measurement scheduling for event forecasting using deep rl. In *International Conference on Machine Learning*, pages 951–960, 2019.

Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in neural information processing systems*, pages 6571–6583, 2018.

Li-Fang Cheng, Niranjani Prasad, and Barbara E Engelhardt. An optimal policy for patient laboratory tests in intensive care units. In *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, volume 24, pages 320–331. World Scientific, 2019.

Ilya Chevyrev and Andrey Kormilitzin. A primer on the signature method in machine learning. *arXiv preprint arXiv:1603.03788*, 2016.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

Edward Choi, Mohammad Taha Bahadori, Elizabeth Searles, Catherine Coffey, Michael Thompson, James Bost, Javier Tejedor-Sojo, and Jimeng Sun. Multi-layer representation learning for medical concepts. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1495–1504. ACM, 2016.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Scott L Fleming, Kuhan Jeyapragasan, Tony Duan, Daisy Ding, Saurabh Gombar, Nigam Shah, and Emma Brunskill. Missingness as stability: Understanding the structure of missingness in longitudinal ehr data and its impact on reinforcement learning in healthcare. *arXiv preprint arXiv:1911.07084*, 2019.

Joseph Futoma, Sanjay Hariharan, Katherine Heller, Mark Sendak, Nathan Brajer, Meredith Clement, Armando Bedoya, and Cara O'Brien. An improved multi-output gaussian process rnn with real-time validation for early sepsis detection. In *Machine Learning for Healthcare Conference*, pages 243–254, 2017.

Marzyeh Ghassemi, Tristan Naumann, Peter Schulam, Andrew L Beam, Irene Y Chen, and Rajesh Ranganath. Practical guidance on artificial intelligence for health-care data. *The Lancet Digital Health*, 1(4):e157–e159, 2019.

Omer Gottesman, Fredrik Johansson, Matthieu Komorowski, Aldo Faisal, David Sontag, Finale Doshi-Velez, and Leo Anthony Celi. Guidelines for reinforcement learning in healthcare. *Nat Med*, 25(1):16–18, 2019.

Arthur Guez, Robert D Vincent, Massimo Avoli, and Joelle Pineau. Adaptive treatment of epilepsy via batch-mode reinforcement learning. In *AAAI*, 2008.

Katharine E Henry, David N Hager, Peter J Pronovost, and Suchi Saria. A targeted real-time early warning score (trewscore) for septic shock. *Science translational medicine*, 7 (299):299ra122–299ra122, 2015.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Caleb W Hug and Peter Szolovits. Icu acuity: real-time models versus daily models. In *AMIA annual symposium proceedings*, volume 2009, page 260. American Medical Informatics Association, 2009.

Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*, 2016.

Alistair EW Johnson and Roger G Mark. Real-time mortality prediction in the intensive care unit. In *AMIA Annual Symposium Proceedings*, volume 2017, page 994. American Medical Informatics Association, 2017.

Alistair EW Johnson, Andrew A Kramer, and Gari D Clifford. A new severity of illness scale using a subset of acute physiology and chronic health evaluation data elements shows comparable predictive accuracy. *Critical care medicine*, 41(7):1711–1718, 2013.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.

David J. Stone Leo A. Celi Johnson, Alistair EW and Tom J. Pollard. The MIMIC Code Repository: enabling reproducibility in critical care research. *Journal of the American Medical Informatics Association (2017): ocx084*, 2017. Accessed: 2019-08-16.

Rico Jonschkowski and Oliver Brock. State representation learning in robotics: Using prior knowledge about physical interaction. In *Robotics: Science and Systems*, 2014.

Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. An empirical exploration of recurrent network architectures. In *International Conference on Machine Learning*, pages 2342–2350, 2015.

Patrick Kidger, Patric Bonnier, Imanol Perez Arribas, Cristopher Salvi, and Terry Lyons. Deep signature transforms. In *Advances in Neural Information Processing Systems*, pages 3099–3109, 2019.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Matthieu Komorowski. AI Clinician. https://github.com/matthieukomorowski/AI_Clinician, 2018. Accessed: 2019-08-16.

Matthieu Komorowski, Leo A Celi, Omar Badawi, Anthony C Gordon, and A Aldo Faisal. The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care. *Nature Medicine*, 24(11):1716, 2018.

V. Krishnamurthy. *Partially Observed Markov Decision Processes*. Cambridge University Press, 2016. ISBN 9781107134607. URL https://books.google.ca/books?id=j9CxCwAAQBAJ.

Jean-Roger Le Gall, Stanley Lemeshow, and Fabienne Saulnier. A new simplified acute physiology score (saps ii) based on a european/north american multicenter study. *Jama*, 270(24):2957–2963, 1993a.

J.R. Le Gall, S. Lemeshow, and F. Saulnier. A new simplified acute physiology score (saps ii) based on a european/north american multicenter study. *JAMA*, 270(24):2957–2963, 1993b.

Timothée Lesort, Natalia Díaz-Rodríguez, Jean-Franois Goudou, and David Filliat. State representation learning for control: An overview. *Neural Networks*, 2018.

Zachary C Lipton, David Kale, and Randall Wetzel. Directly modeling missing data in sequences with rnns: Improved classification of clinical time series. In *Machine Learning for Healthcare Conference*, pages 253–270, 2016.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

James Morrill, Andrey Kormilitzin, Alejo Nevado-Holgado, Sumanth Swaminathan, Sam Howison, and Terry Lyons. The signature-based model for early detection of sepsis from electronic health records in the intensive care unit. In *2019 Computing in Cardiology Conference (CinC). IEEE*, 2019.

Xuefeng Peng, Yi Ding, David Wihl, Omer Gottesman, Matthieu Komorowski, Li-wei H Lehman, Andrew Ross, Aldo Faisal, and Finale Doshi-Velez. Improving sepsis treatment strategies by combining deep and kernel-based reinforcement learning. In *AMIA Annual Symposium Proceedings*, volume 2018, page 887. American Medical Informatics Association, 2018.

Joelle Pineau, Geoffrey J. Gordon, and Sebastian Thrun. Point-based value iteration: An anytime algorithm for pomdps. In *IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003*, pages 1025–1032, 2003.

Niranjani Prasad, Li-Fang Cheng, Corey Chivers, Michael Draugelis, and Barbara E Engelhardt. A reinforcement learning approach to weaning of mechanical ventilation in intensive care units. *arXiv preprint arXiv:1704.06300*, 2017.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Antonin Raffin, Ashley Hill, Kalifou René Traoré, Timothée Lesort, Natalia Díaz-Rodríguez, and David Filliat. Decoupling feature extraction from policy learning: assessing benefits of state representation learning in goal based robotics. *arXiv preprint arXiv:1901.08651*, 2019.

Aniruddh Raghu, Matthieu Komorowski, Imran Ahmed, Leo Celi, Peter Szolovits, and Marzyeh Ghassemi. Deep reinforcement learning for sepsis treatment. *arXiv preprint arXiv:1711.09602*, 2017a.

Aniruddh Raghu, Matthieu Komorowski, Leo Anthony Celi, Peter Szolovits, and Marzyeh Ghassemi. Continuous state-space models for optimal sepsis treatment-a deep reinforcement learning approach. *arXiv preprint arXiv:1705.08422*, 2017b.

Aniruddh Raghu, Matthieu Komorowski, and Sumeetpal Singh. Model-based reinforcement learning for sepsis treatment. *arXiv preprint arXiv:1811.09602*, 2018.

Eamon P Raith, Andrew A Udy, Michael Bailey, Steven McGloughlin, Christopher MacIsaac, Rinaldo Bellomo, and David V Pilcher. Prognostic accuracy of the sofa score, sirs criteria, and qsofa score for in-hospital mortality among adults with suspected infection admitted to the intensive care unit. *Jama*, 317(3):290–300, 2017.

Yulia Rubanova, Tian Qi Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In *Advances in Neural Information Processing Systems*, pages 5321–5331, 2019.

Najibesadat Sadati, Milad Zafar Nezhad, Ratna Babu Chinnam, and Dongxiao Zhu. Representation learning with autoencoders for electronic health records: A comparative study. *arXiv preprint arXiv:1801.02961*, 2018.

Suchi Saria. Individualized sepsis treatment using reinforcement learning. *Nature medicine*, 24(11):1641–1642, 2018.

Anis Sharafoddini, Joel A Dubin, David M Maslove, and Joon Lee. A new insight into missing data in intensive care unit patient profiles: Observational study. *JMIR medical informatics*, 7(1):e11605, 2019.

Ikaro Silva, George Moody, Daniel J Scott, Leo A Celi, and Roger G Mark. Predicting in-hospital mortality of icu patients: The physionet/computing in cardiology challenge 2012. In *2012 Computing in Cardiology*, pages 245–248. IEEE, 2012.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.

Mervyn Singer, Clifford S. Deutschman, Christopher Warren Seymour, Manu Shankar-Hari, Djillali Annane, Michael Bauer, Rinaldo Bellomo, Gordon R. Bernard, Jean-Daniel Chiche, Craig M. Coopersmith, Richard S. Hotchkiss, Mitchell M. Levy, John C. Marshall, Greg S. Martin, Steven M. Opal, Gordon D. Rubenfeld, Tom van der Poll, Jean-Louis Vincent, and Derek C. Angus. The third international consensus definitions for sepsis and septic shock (sepsis-3). *JAMA*, 315(8):801, feb 2016. doi: 10.1001/jama.2016.0287. URL https://doi.org/10.1001%2Fjama.2016.0287.

Richard D Smallwood and Edward J Sondik. The optimal control of partially observable markov processes over a finite horizon. *Operations research*, 21(5):1071–1088, 1973.

Edward J Sondik. The optimal control of partially observable markov processes over the infinite horizon: Discounted costs. *Operations research*, 26(2):282–304, 1978.

Jayakumar Subramanian and Aditya Mahajan. Approximate information state for partially observed systems. In *Proceedings of the 58th IEEE Conference on Decision and Control*. IEEE, 2019.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112 (1-2):181–211, 1999.

J-L Vincent, Rui Moreno, Jukka Takala, Sheila Willatts, Arnaldo De Mendonça, Hajo Bruining, CK Reinhart, PeterM Suter, and Lambertius G Thijs. The sofa (sepsis-related organ failure assessment) score to describe organ dysfunction/failure, 1996.

Jean-Louis Vincent, Steven M Opal, John C Marshall, and Kevin J Tracey. Sepsis definitions: time for change. *The Lancet*, 381(9868):774 – 775, 2013. ISSN 0140-6736. doi: https://doi.org/10.1016/S0140-6736(12)61815-7. URL http://www.sciencedirect.com/science/article/pii/S0140673612618157.

Wei-Hung Weng and Peter Szolovits. Representation learning for electronic health records. *arXiv preprint arXiv:1909.09248*, 2019.

Chao Yu, Jiming Liu, and Shamim Nemati. Reinforcement learning in healthcare: a survey. *arXiv preprint arXiv:1908.08796*, 2019.

Amy Zhang, Harsh Satija, and Joelle Pineau. Decoupling dynamics and reward for transfer learning. *arXiv preprint arXiv:1804.10689*, 2018.

Appendix

Appendix A. Details about Patient Cohort

A.1. Data extraction and preprocessing

To construct our patient cohort from the MIMIC-III database,we follow the approach described by Komorowski et al. (2018) and the associated code repository given in Komorowski (2018). This includes all adult patients in the intensive care fulfilling the sepsis 3 criteria. A presumed onset of sepsis is defined by temporally related prescription of antibiotics and test results from microbiological cultures. All patient observations are extracted in a 72h span around this presumed onset of sepsis (24h before presumed onset to 48h afterwards). The original cohort extracted by Komorowski et al. (2018) contained a set of 48 variables including demographics, Elixhauser status, vital signs, laboratory values, fluids and vasopressors received and fluid balance. Missing or irregularly sampled data was filled using a time-limited sample-and-hold approach based on clinically relevant periods for each feature. All values that remained missing after this step were imputed using a nearest-neighbor approach. After imputation, all features are z-normalized.

Observed actions (administration of fluids or vasopressors) are categorized by volume and put into 5 discrete bins per action type. The combination of the type of actions leads to 25 possible discrete actions.

A.2. Features used in this paper

As described in Section 3, we only maintain features that correspond to continuous quantities, the evolution of which may result from the selected actions. Those columns we remove from the original extracted cohort by Komorowski et al. are intended to be added to the learned state representations used for developing treatment policies. We include the patient features used in this paper in Table 1.

Table 5: Patient features used for learning state representations for predicting future observations

Glascow Coma Scale	Heart Rate	Sys. Blood Pressure
Dia. Blood Pressure	Mean Blood Pressure	Respiratory Rate
Body Temperature (C)	FiO2	Potassium
Sodium	Chloride	Glucose
Magnesium	Calcium	Hemoglobin
White Blood Cells	Platelets	PTT
PT	Arterial pH	Lactate
PaO2	PaCO2	PaO2 / FiO2
Bicarbonate (HCO3)	SpO2	BUN
Creatinine	SGOT	SGPT
Bilirubin	INR	Base Excess

A.3. Acuity Scores

Here we briefly describe the acuity scores we use to constrain the state representations we learn for predicting future observations. For the particular heuristics used to calculate these scores, we refer the reader to the originating literature sources.

A.3.1. Sepsis-related Organ Failure Assessment - SOFA

The Sepsis-related Organ Failure Assessment score was developed to provide clinicians with an objective measure of organ dysfunction in a patient (Vincent et al., 1996). The score is evaluated for 6 organ systems: pulmonary, renal, hepatic, cardiovascular, haematologic and neurologic. Under the Sepsis-3 criteria, a patient is presumed to be septic if the SOFA score increases by 2 or more points.

A.3.2. Simplified Acute Physiology Score II - SAPS II

The Simplified Acute Physiology Score II (SAPS II) (Le Gall et al., 1993a) was developed to improve issues with SAPS, a simplified score using 13 physiological parameters. These parameters were chosen using univariate feature selection to exclude features uncorrelated with hospital mortality.

A.3.3. Oxford Acute Severity of Illness Score - OASIS

The Oxford Acute Severity of Illness Score (OASIS) is a severity score developed algorithmically which directly optimized for clinical relevance, simultaneously performing multivariate feature selection (Johnson et al., 2013). OASIS requires only 10 features, without depending on laboratory measurements, diagnosis or comorbidity information.

Appendix B. State construction in prior work

See Table 6 for an overview of how prior work has constructed state representations for RL in healthcare settings.

Appendix C. Architecture Details

C.1. RNN

Recurrent Neural Networks (RNNs) are extensions of conventional feed-forward neural networks capable of receiving correlated sequences as input. The RNN also handles variable-length sequences by utilizing a recurrent hidden state, activated by features propagated from the previous timestep. When provided an observation O_t from a sequence, the RNN updates its recurrent hidden state h_t by a nonlinear function that associates the O_t with h_{t-1}. Initially, this hidden state is set to a vector of zeros. This hidden

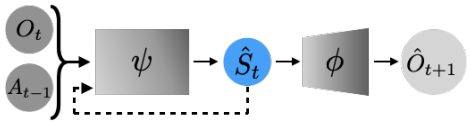

Figure 7: A basic RNN architecture for next observation prediction

Table 6: State construction for RL in healthcare - background

Ref	Domain	State Construction
Guez et al. (2008)	Epilepsy	114 dimensional continuous - summarizing past EEG activity
Raghu et al. (2017a)	Sepsis (MIMIC-III)	Time augmented last observation ($47 + 1 = 48$ dimensional)
Prasad et al. (2017)	Weaning of mechanical ventilation (MIMIC-III)	Last observation (32 dimensional)
Komorowski et al. (2018)	Sepsis (MIMIC-III)	Clustered state with 750 clusters
Raghu et al. (2018)	Sepsis (MIMIC-III)	k-Markov with $k = 4$; $198 = 4 \times 47$ dimensional state space
Peng et al. (2018)	Sepsis (MIMIC-III)	Sequence embedding with RNN (128 dimensional hidden state from 43 features)
Chang et al. (2019)	Sepsis (MIMIC-III)	Last observation (39 dimensional extracted from time-series + 38 static covariates)
Cheng et al. (2019)	Lab testing (MIMIC-III)	Last observation (21 dimensional). Data imputation done using a Multi-output Gaussian Process framework.

state, an embedding of the prior sequence of observations, can then be used to make predictions of various kinds depending on the specific context the model is trained for. See (Chung et al., 2014; Jozefowicz et al., 2015) for a more detailed introduction to such networks.

For predicting the next observation in healthcare settings we make the following adjustments to a basic RNN architecture, shown in Figure 7. The current observation O_t is concatenated with the selected action and passed into the RNN along with the hidden state representation from the previous time step $\hat{S}_{t-1}$. The hidden state representation $\hat{S}_t$ is then passed to a decoder function ϕ that provides the prediction of the next observation $\hat{O}_{t+1}$.

We use a 3-layer Recurrent Neural Network (RNN) for estimating the encoding function ψ, where the first layer is a fully connected layer that maps the current observation and action (58 dimensional input: 33 dimensional observation with a 25 dimensional one-hot encoded action) to 64 neurons with ReLU activation. This is followed by another $(64, 128)$ fully connected layer with ReLU activation which is followed by a GRU layer (Cho et al., 2014) with hidden state size $\hat{d_s}$ chosen from $\{4, 8, 16, 32, 64, 128, 256\}$. For estimating the decoder function ϕ, we use a 3-layer feed-forward neural network with sizes $(\hat{d_s}, 64)$, $(64, 128)$ and $(128, 33)$ with ReLU activation for the first two layers. The last layer outputs a 33-

dimensional vector, which forms the mean-vector of a unit-variance multi-variate Gaussian distribution which is then used to predict the next observation as given in 1.

The best RNN architectures for each choice of $\hat{d}_s$ were trained for 600 epochs with a learning rate of $1e-4$. The λs for constraining the training to correlate with acuity scores are all set to 100.

C.2. AIS

The Approximate Information State (AIS) (Subramanian and Mahajan, 2019) was introduced as an approach to learning the state representation for POMDPs for use in dynamic programming. The learned representation is defined in terms of properties that can be estimated from data, so it lends itself to be used in model pipelines where the state is used for some downstream task. The function ψ is comprised of an encoder followed by a gated recurrent unit (Cho et al., 2014) which outputs the representation $\hat{S}_t$. The input to ψ is the concatenation of the observation O_t and last selected action A_{t-1}. The current action A_t (which is typically induced from the policy, conditioned on $\hat{S}_t$) is concatenated to the state representation $\hat{S}_t$ and then fed through the decoder function ϕ to predict the next observation $\hat{O}_{t+1}$.

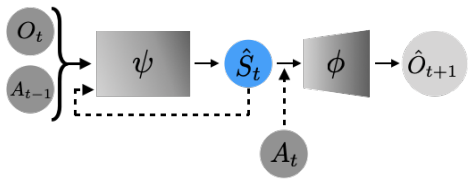

Figure 8: AIS architecture, adapted from Subramanian and Mahajan (2019)

AIS uses the same base architecture as the basic RNN with one adjustment. For the decoder function ϕ we augment the input space by appending the current action A_t to the state representation S_t. Therefore the AIS decoder function ϕ, constitutes a 3-layer feed-forward neural network with sizes $(\hat{d}_s + 25, 64)$, $(64, 128)$ and $(128, 33)$ with ReLU activation for the first two layers. The last layer outputs a 33-dimensional vector, which forms the mean-vector of a unit-variance multi-variate Gaussian distribution which is then used to predict the next observation as given in Equation 1.

The best AIS architectures for each choice of $\hat{d}_s$ were trained for 600 epochs with a learning rate of $5e-4$. The λs for constraining the training to correlate with acuity scores are all set to 100.

C.3. DDM

Zhang et al. (2018), introduced an model-based RL algorithm that decoupled dynamics and reward learning. This decoupling aimed to improve the generalization and stability of RL algorithms operating in environments where perturbations to the observations may occur. The dynamics module utilizes recurrent models to associate sequences of prior observations and their affect on subsequent observations.

We adapt this module, shown in Figure C.3, for the purpose of predicting the next observation in a healthcare setting. The observation O_t is provided to an encoder ψ_{enc} the output of which is concatenated to the selected action A_t and fed into an LSTM (ψ_{LSTM}) (Hochreiter and Schmidhuber, 1997) which provides the state representation $\hat{S}_t$. This state rep-

resentation is then provided to the decoder function ϕ to provide a prediction of the next observation $\hat{O}_{t+1}$. To stabilize the development of this learned state representation, $\hat{S}_t$ is also fed to an inverse dynamics function (denoted by "Inverse" in Fig. C.3) along with the true next observation to predict the action used to generate $\hat{S}_t$.

For specific details about the set-up and training of decoupled dynamics module (DDM), we refer the reader to Zhang et al. (2018)[4].

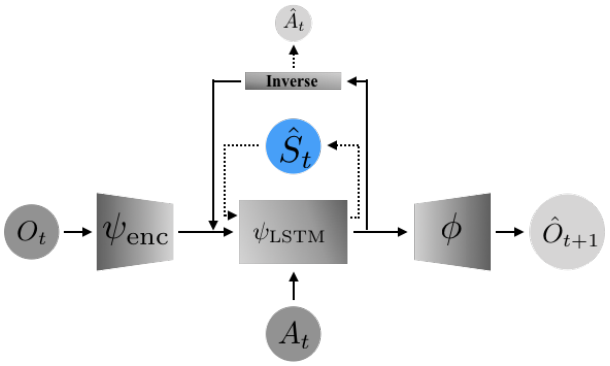

Figure 9: The Decoupled Dynamics Module from Zhang et al. (2018), adapted from its original presentation

The DDM archtecture is made up of three modules, an encoder (ψ_{enc}), a dynamics module (ψ_{LSTM}), and a decoder (ϕ). These three modules combine to both create a latent embedding space for the state representations $\hat{S}$ while also decoding these representations to predict the next observation. The encoding function ψ_{enc} is comprised of a 3-layer feed-forward neural network with sizes $(33, \hat{d}_s)$, $(\hat{d}_s, 288)$, $(288, \hat{d}_s)$. The first two layers are followed by exponential linear unit (ELU) activation functions. The final layer is passed through a `tanh` activation and provided as output to the dynamics model ψ_{LSTM}.

The dynamics module ψ_{LSTM} receives as input the encoded observation and next observation, $z_t = \psi_{\text{enc}}(O_t)$, $z_{t+1} = \psi_{\text{enc}}(O_{t+1})$ respectively the current action A_t and two separate hidden state vectors that describe the distribution of the latent distribution $\hat{Z}$ that the encoder produces estimates of with each observation. The dynamics module ψ_{LSTM} begins with two linear layers of sizes $(25, \hat{d}_s)$ and $(\hat{d}_s, \hat{d}_s)$, the first of which has an ELU activation function. These layers embed the action A_t. This embedding is concatenated with the encoded observation z_t and passed through a linear layer with shape $(2 * \hat{d}_s, \hat{d}_s)$. The output of this embedding is then passed to a LSTM Cell with input dimensions of dimension $\hat{d}_s$ and produces the mean and variance vectors of the latent distribution, each of size $\hat{d}_s$. The mean vector is then passed through a `tanh` activation function and provided as an estimate of the encoded next observation $\hat{z}_{t+1}$. Finally, the dynamics module infers the action A_t that caused the transition between the encoded z_t and z_{t+1}. These encoded representations of the observations are concatenated and passed through a 2-layer fully connected neural network, the first layer with shape $(2 * \hat{d}_s, \hat{d}_s)$ followed by an ELU activation with the second layer having shape $(\hat{d}_s, 25)$.

The decoder function ϕ is a 3-layer fully connected neural network. The first two layers have the shapes $(\hat{d}_s, 288)$, $(288, \hat{d}_s)$ each followed by ELU activation functions. The final layer has the shape $(\hat{d}_s, 33)$. The decoder ϕ takes the predicted next encoded observation ($\hat{z}_{t+1}$, which we use as our learned state representation) as input. The function outputs a 33-dimensional vector which is the prediction for the next observation $\hat{O}_{t+1}$.

4. The author's code can be accessed at https://github.com/facebookresearch/ddr

The best DDM architectures were trained for 600 epochs with the following learning rates for each choice of $\hat{d}_s$;
$\{4 : 1e-3,\ 8 : \ 1e-4,\ 16 : \ 1e-4,\ 32 : \ 5e-4,\ 64 : \ 1e-4,\ 128 : \ 1e-4,\ 256 : \ 1e-4\}$
The λs for constraining the training to correlate with acuity scores are all set to 10.

C.4. DST

As outlined by Kidger et al. (2019), sequentially ordered data can have path-like structure. The statistics of such a path can be represented by the *signature* (Chevyrev and Kormilitzin, 2016). The mapping between a path and its signature is known as the signature transform. Neural network architectures that utilize such transforms may be capable of adequately handling irregularly sampled time-series data from partially observable environments such as those in healthcare.

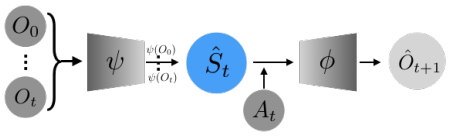

Figure 10: The Deep Signature Transform architecture for next observation prediction

The signature transform Sig^N is defined by an infinite sequence where N roughly corresponds to the order of approximation of matching moments of a distribution. In practice, Sig^N is truncated to include a finite number of elements. The choice of N and the dimension d of the data points of the sequence influence the subsequent number of terms in the truncated signature as $|\text{Sig}^N| = \frac{d^{N+1}-1}{d-1}$. As the dimension d may be large in healthcare settings, for the the signature transform to feasible the sequence of observations $\tau_{j,0:t} = \{O_0, \ldots, O_t\}$ must be passed through a pointwise encoder (i.e. $\psi(\tau_{j,0:t}) = \{\psi(O_0), \ldots, \psi(O_t)\}$) to reduce the dimension d of the input to the signature transform.[5]

We set-up a signature transform for predicting the future observations in a healthcare setting as shown in Figure 10. We pass the sequence of observations $\tau_{j,0:t} = \{O_0, \ldots, O_t\}$ up to the current time through a pointwise encoder ψ. The resulting sequence $\psi(\tau_{j,0:t})$ is processed by the signature transform $\text{Sig}^N(\psi(\tau_{j,0:t}))$ of order N to produce the learned state representation $\hat{S}_t$. This state representation is then concatenated with the current action A_t then decoded with the function ϕ to predict the next observation $\hat{O}_{t+1}$.

Recently, signature transforms have been incorporated into modern neural network architectures and have been shown to have great promise in a variety of learning paradigms (Kidger et al., 2019). Notably, a model architecture utilizing a signature transform for sepsis prediction won the 2019 Physionet challenge (Morrill et al., 2019). The success of such a model demonstrates that such transforms may be capable of adequately handling irregularly sampled time-series data from partially observable environments.

We use a 2-layer fully-connected Neural Network for estimating the encoding function ψ, where the first layer maps the current observation and action (58 dimensional input: 33 dimensional observation with a 25 dimensional one-hot encoded action) to 64 hidden units with ReLU activation. This is followed by another $(64, dim)$ fully connected layer, where

5. Kidger et al. have published a Python library to efficiently calculate the signature transform and integrate it into modern neural network architectures. It can be found at https://github.com/patrick-kidger/signatory

dim is the chosen embedding dimension, which is followed by the signature transform. For estimating the decoder function ϕ, we use a 3-layer feed-forward neural network with sizes $(|Sig^N|, 256)$, $(256, 128)$ and $(128, 33)$ with ReLU activation for the first two layers. The last layer outputs a 33-dimensional vector, which forms the mean-vector of a unit-variance multi-variate Gaussian distribution which is then used to predict the next observation as given in Equation 1

The best DST architectures for each choice of $\hat{d}_s$ were trained for 600 epochs with a learning rate of $1e-4$. The λs for constraining the training to correlate with acuity scores are all set to 100.

C.5. Latent ODE

Rubanova et al. (2019) generalize the latent transitions between observations inside an RNN to a continuous time differential equation using neural networks, building from the Neural ODE (Chen et al., 2018) framework. This generalization produces a model the authors refer to as the ODE-RNN which is used as the recognition network, or encoder, for a variational autoencoder (VAE) with an additional ODE model serving as the decoder. This complete architecture has been named the Latent ODE as it can propagate an initial observation using deterministic dynamics it has learned from complex sequences of data.[6]

We harness the ability of the Latent ODE to probabilistically model sequences of patient observations through latent variables. Here, a differential equation is used to associate prior observations and their underlying dynamics to form a distribution over the latent variables. Conditioned on the input sequence, this distribution can provide a set of latent parameters from which are then used in the decoding ODE to produce the prediction of subsequent observations.

We adapt the Latent ODE model to receive sequences of prior observations $O_{0:t}$ and actions $A_{0:t-1}$, generating the latent distribution from which the state representation $\hat{S}_t$ is sampled from (see Fig. 11). The set of latent parameters are then concatenated with the current action A_t and used to decode the next observation based on the inferred dynamics of the input sequences. We highlight more detail about the specific architectural implementation of the Latent ODE in this paper in Section C.5.

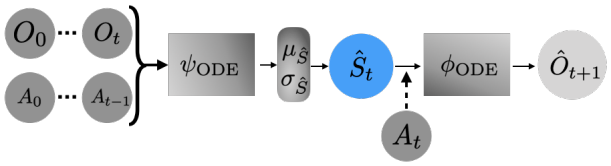

Figure 11: The Latent ODE model adjusted slightly for predicting the next observation

The Latent ODE conceptually derives from the variational autoencoder (VAE) Kingma and Welling (2013) where the encoding function ψ takes the form of an ODE-RNN which infers parameters of the distribution that generates the initial latent state of some sequence of temporally correlated observations provided as input. This encoding function ψ takes as input the sequence of observations and actions concatenated together (such that each observation O_t is attached to the previous action A_{t-1}, the initial observation is concate-

6. The code for the Latent ODE model, utilizing the ODE-RNN can be found at https://github.com/YuliaRubanova/latent_ode

nated with a vector of zeros). This input sequence is provided to a GRU layer that provides output of the size of $\hat{d_s}$. This is then passed through a two layer fully-connected neural network of sizes $(\hat{d_s}, 20)$ and $(20, 2 * \hat{d_s})$, with `tanh` activations. The output of this encoding function ψ is then split to produce the mean and variance of the generating distribution of the initial latent state.

Samples are drawn from this distribution and then associated with the presented data to infer the next latent state through a differential equation solver provided by the `torchdiffeq` package[7]. These latent state sequences are then concatenated with the current action and decoded through a two layer neural network ϕ with shape $(\hat{d_s} + 25, 66)$ $(66, 33)$ with `tanh` activations. We train this model architecture with the same approach as presented in the original paper (Rubanova et al., 2019).

The best Latent ODE architectures for each setting of $\hat{d_s}$ were trained for 600 epochs with a learning rate of $1e-4$. The λs for constraining the training to correlate with acuity scores are all set to 5.

C.6. Autoencoder

To isolate the contribution of the recurrent layer in the RNN (Sec. C.1), we also evaluate a simple autoencoder that replaces that layer in the encoding function ψ with a fully connected layer to produce the state representation $\hat{S_t}$. As is done with AIS (Sec. C.2), we concatenate the current action A_t to $\hat{S_t}$ when predicting the next observation $\hat{O}_{t+1}$ using the decoder function ϕ. The autoencoder architecture shown in Figure 12 was trained using Eqt. 2 as is done with the RNN, AIS, and DST approaches.

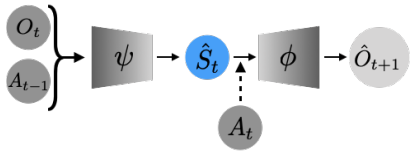

Figure 12: The Autoencoder architecture

The autoencoder's encoding function ψ is comprised of a three layer fully connected neural network with ReLU activations with sizes $(58, 64)$, $(64, 128)$, $(128, \hat{d_s})$ to produce the state representation $\hat{S_t}$. To produce an approximation of the next observation, $\hat{S_t}$ is concatenated with the current action A_t and passed to the decoding function ϕ, another three layer fully connected neural network with ReLU activations. The sizes of the layers comprising ϕ are $(\hat{d_s} + 25, 64)$, $(64, 128)$ and $(128, 33)$. We train this model end-to-end using Equation 1, constrained by the correlation coefficient.

The best autoencoder models for each setting of $\hat{d_s}$ were trained for 600 epochs with a learning rate of $5e-4$. The λs for constraining the training were all set to 100.

Appendix D. Additional PCA Figures

This section contains the nonlinear projection using PCA of the state representations learned from each approach. For simplicity, we only include the representations for the first and final observations of each patient trajectory, colored by the corresponding SOFA score. We also draw lines connecting these points to help infer how the patient's health evolves, as

7. https://github.com/rtqichen/torchdiffeq

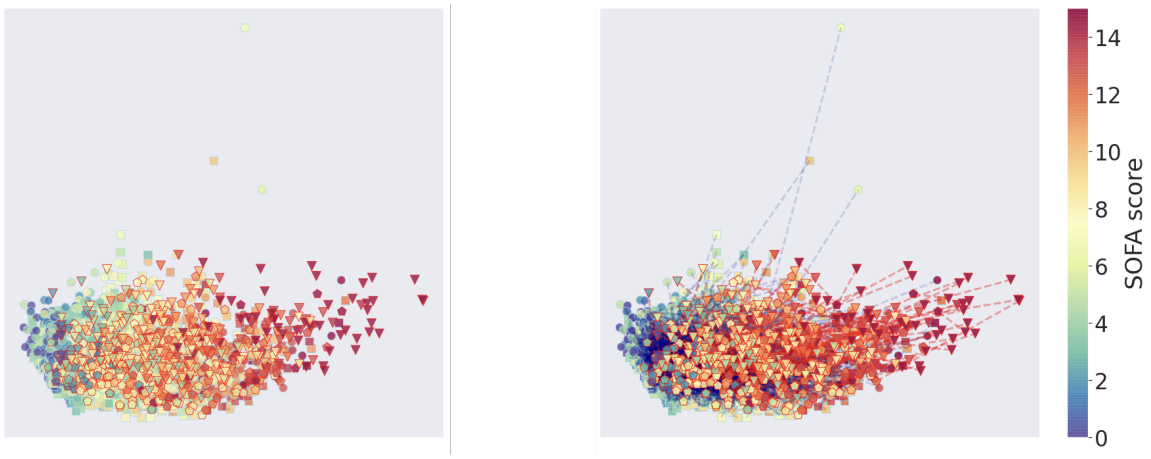

Figure 13: Representations of patient health, learned through an Autoencoder (AE)

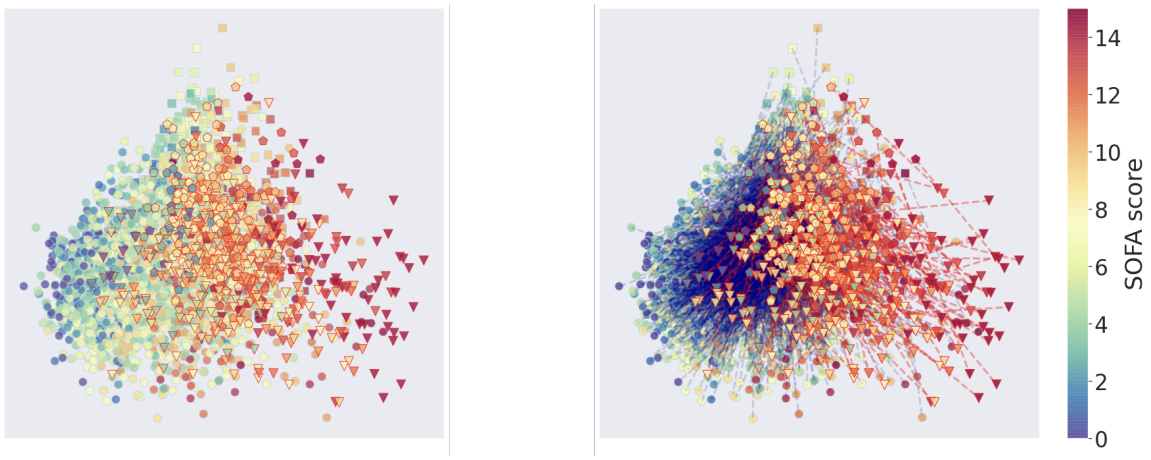

Figure 14: Representations of patient health, learned through Approximate Information State (AIS)

demonstrated in representation space. To aid this inference, we've colored the lines according to patient outcome. Blue lines signify patients who overcame sepsis and survived. Red lines connect the observations of those patients who died following complications associated with their sepsis diagnosis.

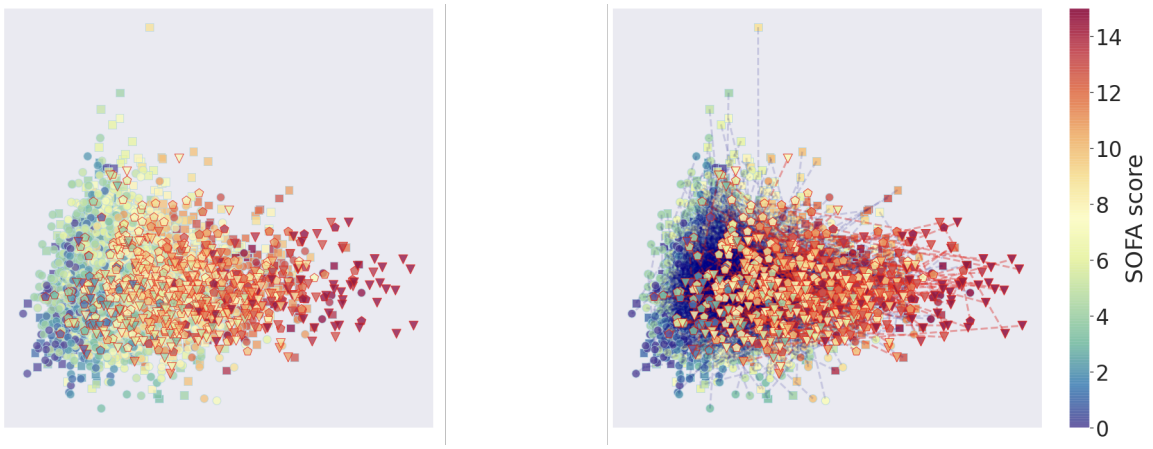

Figure 15: Representations of patient health, learned through the Decoupled Dynamics Module (DDM)

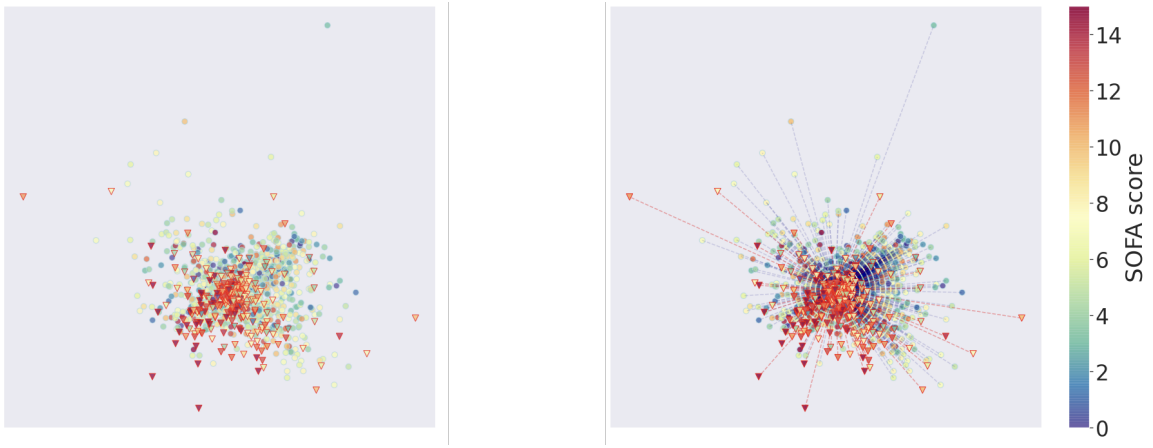

Figure 16: Representations of patient health, learned through the Deep Signature Transform (DST)

Figure 17: Representations of patient health, learned through the Latent ODE (ODE). presents the learned state representations from the Latent ODE approach and serve to provide greater context into the observations made in the previous section as well as reaffirm the model design. The Latent ODE model conceptually descends from the variational autoencoder (VAE) (Kingma and Welling, 2013) in that the encoding function ψ generates a latent distribution, from which samples are drawn to produce predictions through the decoding function ϕ. The Latent ODE model uses the encoding function to infer the initial latent representation of the provided sequence and constructs the generating distribution from these inferred initial representations. This model design is reaffirmed in the learned state representations presented here. The initial observations from each trajectory (regardless of whether the patient survived or not) are all contained in a narrow part of the representation space without regard to their underlying acuity. From this initial representation, the ODE dynamics then push the representations outward as the patient progresses toward discharge or mortality in separate overlapping (at least in the low dimensional projection we have) distributions. This underlying model behavior, compressing state representations into a learned distribution, and represented here helps explain the results in the previous section. Given the close proximity of state representations of widely differing acuity scores, it is not as surprising that the overall Pearson correlation coefficients between them and the acuity scores were low.

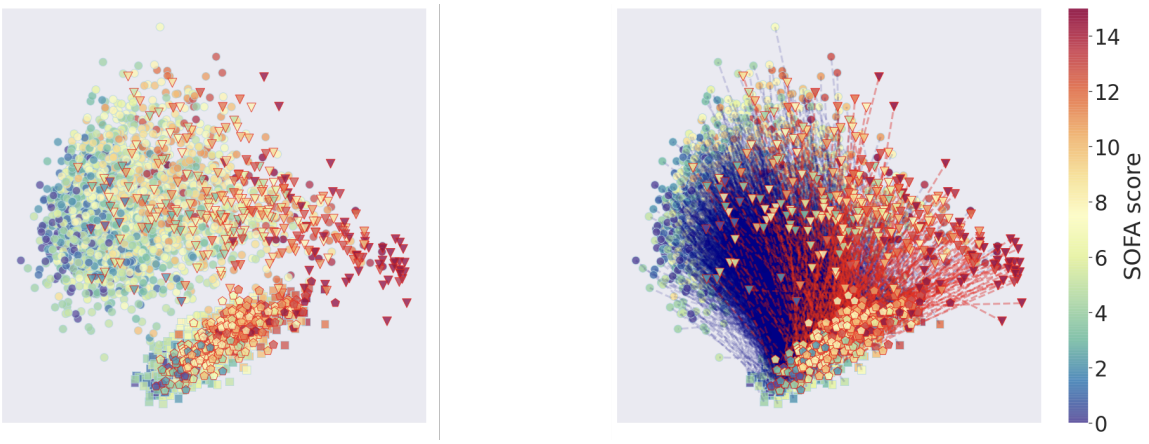

Figure 18: Representations of patient health, learned through a recurrent autoencoder (RNN)

Figure 19: Representations of patient health, learned through an untrained recurrent autoencoder (Rand. RNN)