Proceedings of Machine Learning Research 136:139–160, 2020 Machine Learning for Health (ML4H) 2020

An Empirical Study of Representation Learning for Reinforcement Learning in Healthcare

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

Jayakumar Subramanian JAYAKUMAR.SUBRAMANIAN@GMAIL.COM
Media and Data Science Research Lab, Adobe India

Mehdi Fatemi MEHDI.FATEMI@MICROSOFT.COM
Microsoft Research

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

Editors: Emily Alsentzer[*], Matthew B. A. McDermott[*], Fabian Falck, Suproteem K. Sarkar, Subhrajit Roy[†], Stephanie L. Hyland[†]

Abstract

Reinforcement Learning (RL) has recently been applied to sequential estimation and prediction problems identifying and developing hypothetical treatment strategies for septic patients, with a particular focus on *offline* learning with observational data. In practice, successful RL relies on informative latent states derived from sequential observations to develop optimal treatment strategies. To date, how best to construct such states in a healthcare setting is an open question. In this paper, we perform an empirical study of several information encoding architectures using data from septic patients in the MIMIC-III dataset to form representations of a patient state. We evaluate the impact of representation dimension, correlations with established acuity scores, and the treatment policies derived from them. We find that sequentially formed state representations facilitate effective policy learning in batch settings, validating a more thoughtful approach to representation learning that remains faithful to the sequential and partial nature of healthcare data.

Keywords: representation learning, reinforcement learning, partial observability, sequential autoencoding

1. Introduction

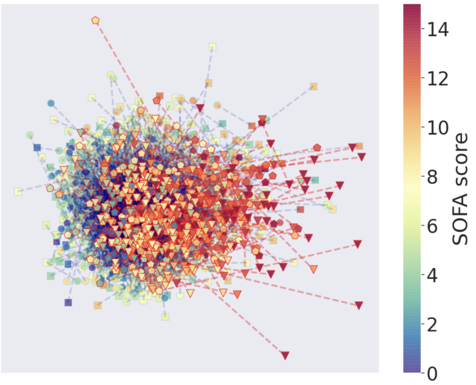


Figure 1: The first and final observations of septic patients in MIMIC-III, colored by SOFA score, visualized via Principal Component Analysis. Blue lines connect observations of patients who recovered, while red lines signify those that did not. Notably, these raw observations of severe health are not directly separable.

Many problems in healthcare are a form of sequential decision making, e.g., clinical staff

[*] Equal Contribution
[†] Equal Contribution

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 (Sox et al., 2007). Recently, reinforcement learning (RL) has been proposed as a promising approach for finding an optimal policy for such processes from data (Gottesman et al., 2019). However, the development of a successful policy rests on the ability to derive informative states from observations. In healthcare these observations are noisy, irregular, and may not convey the entirety of a patient's condition (Obermeyer and Emanuel, 2016). While there are many proposed state construction approaches to handle these challenges (Li et al., 2019; Chang et al., 2019; Peng et al., 2018; Prasad et al., 2017; Raghu et al., 2017a,b), few consider the sequential nature of observations, choosing instead to isolate the features from a single time step to construct the state.

Critical care is one specific setting where sequential data is crucial for predictive modelling. Raw physiological observations may not be clearly separable with respect to patient acuity or outcome, complicating downstream prediction and treatment models (Ibrahim et al., 2020) (see Figure 1). Complex model architectures have shown improved performance on such tasks, due in part to their improved ability to generate high-quality representations (Choi et al., 2016; Sadati et al., 2018; Weng and Szolovits, 2019). Yet within healthcare, the design and learning of patient representations for RL is an open problem (Yu et al., 2019).

In this work, we provide a controlled investigation of sequentially encoded state representations for use within RL applied to healthcare. We focus on the problem of treating septic patients (Liu et al., 2020), using a patient cohort defined by Komorowski et al. (2018) from the MIMIC-III dataset (Johnson et al., 2017). We compare seven encoding architectures, and evaluate representations learned from sequential patient observations through three experiments.

First, we examine the effect of representation dimension when training models to predict **s**ubsequent physiological **o**bservations (SO) through autoencoding (Baldi, 2012) prior physiological observations. We position this as an auxiliary task to the development of a treatment policy (Jaderberg et al., 2017).

Second, we investigate the impact of including contextual information as well as regularization when training these models. Context is added by augmenting the physiological observations with 5 demographic features. When regularizing model training, the learned representations are regularized to correlate with three clinical patient acuity scores – OASIS (Jones et al., 2009), SAPS II (Le Gall et al., 1993a) and SOFA (Johnson et al., 2013). We then qualitatively evaluate representations to determine their correlation with these scores, and embed them into a lower dimensional visualization to demonstrate their separability in contrast to the raw data.

Finally, we learn treatment policies from the encoded patient state representations using a state of the art off-policy RL algorithm, the discretized form of Batch Constrained Q-learning (dBCQ) (Fujimoto et al., 2019a). Policies are evaluated using weighted importance sampling (Mahmood et al., 2014).

To our knowledge, we present the first rigorous empirical evaluation of learned patient state representations that facilitate policy learning. A summary of our contributions are:

- We show that, keeping all other hyperparameters constant, increasing the latent dimensionality could reduce prediction accuracy, indicating that high capacity representations are not always most informative.

- We find that including demographic context when learning the state representation generally improves the performance of predicting SO.
- We demonstrate that sequentially formed state representations can facilitate effective policy learning in batch settings. In particular, we find that representations learned through the recent Neural CDE (Kidger et al., 2020) facilitate an especially effective policy.

2. Background and 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). Recent research has also separated feature extraction from policy learning (Raffin et al., 2019), where the goal is to isolate relevant features of the recorded observations in the representation, and provide more salient information to the policy learning algorithm.

Problems modeled as POMDPs often require a state representation to be specified, typically deriving from prior observations and actions (Kaelbling et al., 1998). Past work in state construction has ranged from concatenation of a finite number of consecutive observations (Mnih et al., 2013) to using the final layer of a recurrent neural network (RNN) to collectively embed a sequence of inputs (Hausknecht and Stone, 2015).

Most prior work in the context of RL and healthcare has constructed states from unprocessed observations, framing the problem as a fully observable MDP [1]. This approach naively abstracts the true nature of the data generating process which is inherently partially observable. Missingness as well as an incomplete understanding of biological and physiological processes contribute to the partial nature of healthcare observations. There is a growing set of RL literature in this applied space that accounts for partial observability explicitly. The literature specific to sepsis treatment (Tsoukalas et al., 2015; Li et al., 2018; Peng et al., 2018; Li et al., 2019; Lu et al., 2020) often learns state representations by utilizing recurrent methods, encoding sequentially observed features of the patient's condition into a hidden state.

To date, none of these works provide any analysis or justification of specific state representation choices. In this paper we address this empirical gap by rigorously evaluating multiple recurrent state representation learning approaches for use in healthcare. With this study we hope to provide a foundation for further research into representation learning for sequential decision problems within healthcare.

3. Data

We consider the treatment of septic patients using data from the Medical Information Mart for Intensive Care (MIMIC-III) dataset (v1.4) Johnson et al. (2016). We follow Komorowski et al. (2018) to extract and preprocess[2] relevant vital and lab measurements to build a cohort of 19,418 patients among which there is an observed mortality rate just above 9% (determined by death within 48h of the final observation).

To evaluate the formation of sequential representations of a patient's condition, we focus on patient vital signs and lab measurements that change over time, whether in response to se-

[1] For reference, Table 4 summarizes these approaches, found in Appendix C

[2] Code available at https://github.com/matthieukomorowski/AI_Clinician

lected treatments or as a consequence of their acute condition. This creates a dataset of 33 features $\mathcal{O}$ with a discrete categorical action space with 25 possible choices of combination between fluid and vasopressor amounts. We also experiment with including 5 additional demographic features $\mathfrak{D}$.

We include a list of features in Table 2 with additional details included in Section A of the Appendix.

4. Methods

In this section, we provide a general overview of state representation learning via autoencoding architectures. We focus on the context of our first experimental analysis, where representations are used to predict the subsequent observation (SO).

4.1. General overview

With a batch of observed patient trajectories—comprised of transitions between subsequent observations O_t and O_{t+1} following treatment action A_t—we seek to learn an encoding function $\psi : \mathcal{H}_{t,t-1} \to \hat{S}_t$ as well as a decoding prediction function $\phi : \hat{S}_t \times A_t \to \hat{O}_{t+1}$. Here the history $\mathcal{H}_{t,t-1}$ contains all observations $O_{0:t}$ and actions $A_{0:t-1}$ preceding the target observation O_{t+1}. Together, the encoding and decoding functions form a prediction $\hat{O}_{t+1}$ using the learned state representation $\hat{S}_t$. That is, $\hat{O}_{t+1} = \phi\left(\psi\left(\mathcal{H}_{t,t-1}\right), A_t\right) = \phi(\hat{S}_t, A_t)$. To facilitate sequentially stable predictions for the state representation $\hat{S}_t$ we choose encoding functions ψ with a recurrent structure. Thus, $\hat{S}_t$ implicitly embeds the history $\mathcal{H}_{t,t-1}$, which has been shown to improve sepsis treatment policies (Li et al., 2019).

We jointly train the encoding function ψ and decoding function ϕ via a loss function $\mathcal{L}(O_{t+1}, \hat{O}_{t+1})$, which in general computes the mean squared error between the predicted and true SO, as specified by the particular encoding approach.

4.2. Information encoding models

We target six recurrent modeling approaches, largely motivated by their development to learn dynamics models:

- Basic RNN Autoencoder (RNN) (Chung et al., 2014)
- Approximate Information State (AIS, Subramanian and Mahajan (2019))
- Neural Controlled Differential Equations (CDE, Kidger et al. (2020))
- Decoupled Dynamics Module (DDM, Zhang et al. (2018))
- Deep Signature Transforms (DST, Bonnier et al. (2019))
- And the ODE-RNN (ODE, Rubanova et al. (2019))

These approaches are depicted in Figure 2. We also compare these approaches to a simple non-recurrent Autoencoder (AE). A comparative overview of the features that differentiate each approach as well as specific details about how each are trained are presented in Sec. B of the Appendix.

The unifying feature among these approaches is the development of a latent representation space $\hat{S}$ that encodes information about the patient observations made over time. The formation of $\hat{S}$ is meant to develop informative representations to facilitate better downstream policy learning, by implicitly accounting for the history $\mathcal{H}_{t,t-1}$. That is, we seek to develop a strategy to select treatments based on the encoded history via the learned state representation: $A_t \sim \pi(\hat{S}_t|\mathcal{H}_{t,t-1})$. Through the remainder of this work, we evaluate the characteristics of the representations embedded in $\hat{S}$.

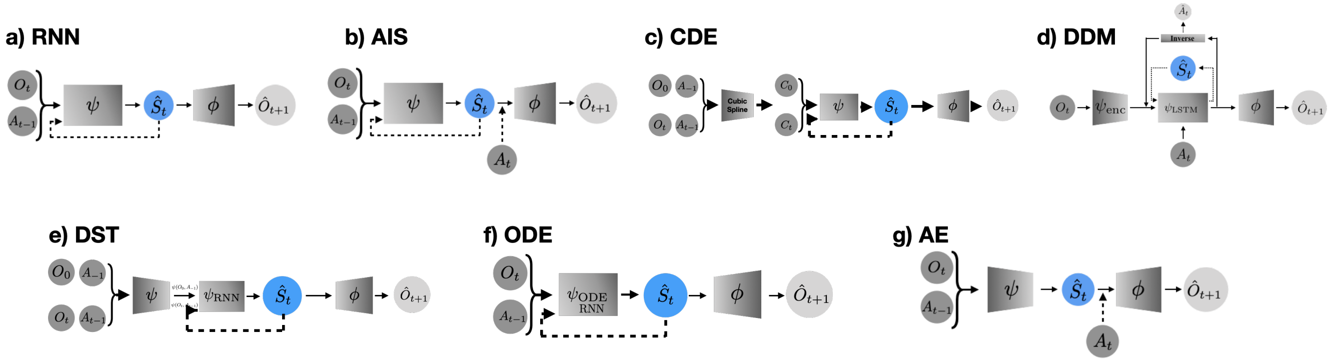


Figure 2: The architectures used to construct state representations via predicting future observations. **a)** basic RNN autoencoder **b)** Approximate Information State (Subramanian and Mahajan, 2019) **c)** Neural CDE (Kidger et al., 2020) **d)** Decoupled Dynamics Module (Zhang et al., 2018) **e)** Deep Signature Transform (Bonnier et al., 2019) **f)** ODE-RNN (Rubanova et al., 2019) **g)** a non-recurrent Autoencoder. See Table 3 in Appendix, Section B for a summary.

Model training: We separate the data into a 70/15/15 train/validation/test split using stratified sampling. This maintains the same proportions of each terminal outcome (survival or mortality), and ensures that no patients are repeated across splits. All models were trained for the same number of epochs, using a variety of learning rates and 5 random initializations. The final settings for each model architecture are provided in the Appendix, Section B.

4.3. Augmenting the learning process

In hopes of ensuring that the intermediate state representations $\hat{S}_t$ retain clinically relevant features, we investigate augmenting the training of the representation space S through a combination of two options: (1) Include the demographic context features $\mathfrak{D}$ (e.g. age, gender, etc.) as input to the encoder function ψ. When training with this option the history $\mathcal{H}_{t,t-1}$ contains observations $O_i^+ = [O_i, \mathfrak{D}_i]$. (2) Regularize the loss function by 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 — through a linear combination of the correlation coefficients to subtract from the loss. The complete objective function when using this form of regularization is then,

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

where $\lambda\ \rho(\hat{S}_t) = \lambda_1\ \rho^{\text{SOFA}}(\hat{S}_t) + \lambda_2\ \rho^{\text{SAPS II}}(\hat{S}_t) + \lambda_3\ \rho^{\text{OASIS}}(\hat{S}_t)$. We choose the hyperparameter λ so that the final prediction loss of the regularized model is not inordinately larger than its unregularized counterpart. Additionally, we set $\lambda_1 = \lambda_2 = \lambda_3$ for simplicity in this paper yet these hyperparameters could be chosen independently of one another.

4.4. Policy development

We train policies on each of the learned state representations outlined in Section 4.2. As we do not have the ability to generate more data through an exploration of novel treatment strategies, we develop a policy using offline, batch reinforcement learning. In this setting, it is critical that the estimated value function not extrapolate to actions that are absent from the provided data (Gottesman et al., 2019). To avoid this extrapolation error Fujimoto et al. (2019b) developed an algorithm that truncates any Q-function estimate corresponding to actions that fall outside the support of the dataset. This algorithm, Batch Constrained Q-Learning (BCQ), originally designed for continuous control problems was later adapted for use in discrete action settings (Fujimoto et al., 2019a).

We use this discretized BCQ algorithm to learn treatment policies from state representations $\hat{S}$. We train the policies using the encoded training subset of our data and validate the performance with the testing subset using weighted importance sampling (WIS), following Li et al. (2019). The WIS return for each policy throughout training is computed by: $R^{\text{WIS}} = \frac{\sum_n^N w_n R_n}{\sum_n^N w_n}$, where the w_n are the per-trajectory IS weights and R_n is the observed outcome of the trajectory. All further details regarding policy training and intermediate results are provided in Section D.1.

5. Empirical Study

We evaluate the representations $\hat{S}_t$ learned from patient data following the three experiments outlined at the conclusion of Section 1. All analyses and results reported through the remainder of this section are provided using only the test set of the patient cohort. All code used to extract and preprocess the data, train and evaluate the encoding models as well as the policies can be found at https://github.com/MLforHealth/rl_representations.

5.1. Representation dimension in SO prediction

We evaluate the accuracy of predicting the SO O_{t+1} from O_t and A_t. Our primary investigation considers the effect of varying the dimension $\hat{d}_s$ of the learned state representation $\hat{S}_t$ from the set $\hat{d}_s \in \{4, 8, 16, 32, 64, 128, 256\}$. Other than varying the latent dimension in each run, we keep all other model and optimization hyperparameters constant. This experiment evaluates the information capacity needed in the state representation $\hat{S}_t$ to adequately predict the SO.

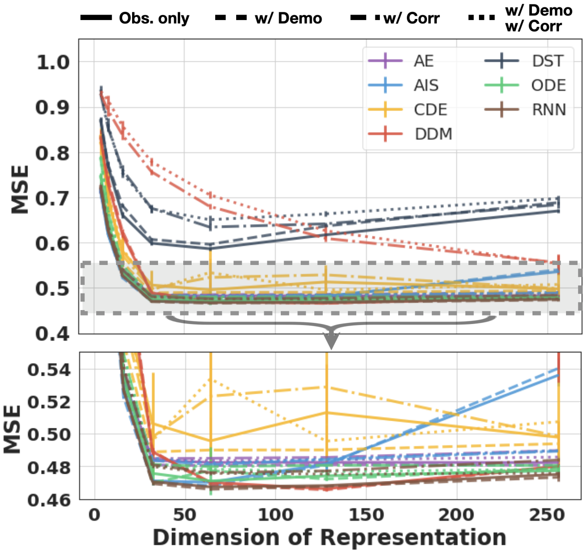


Figure 3: Mean squared error for SO prediction as a result of varying $\hat{d}_s$, comparing various training settings. Error bars are twice the std. dev. of each model over 5 random seeds. We note that augmenting the input to the encoding function ψ with demographic context generally improves prediction performance. See Table 1 for the best performing settings.

Results from these models, learned through the described training settings, are presented in Figure 3 and Table 1. We see that the prediction performance of these models saturates as the dimension increases beyond 64, with the test loss increasing with larger representations. Aside from DST, the best performing settings of all other approaches converge between a loss of 0.46 and 0.48. This indicates that the highest capacity representations may not be the most informative for this prediction task.

5.2. Augmenting learning in SO prediction

We evaluate the two proposed training augmentations (see Sec. 4.3) — adding demographic features $\mathfrak{D}$ during training, and regularizing $\hat{S}$ to be correlated with SOFA, SAPS II and OASIS — via the accuracy of predicting the SO O_{t+1} from O_t and A_t.

When augmenting the input to the encoding function ψ with demographic context $\mathfrak{D}$ the prediction performance is generally improved (see the dashed curves in Figure 3). In contrast, the performance slightly degrades when the learned representations are regularized to be correlated with acuity scores (see the dotted and dot-dash lines in Figure 3), except for the DDM and DST models where there is a noticeable negative effect on model performance.

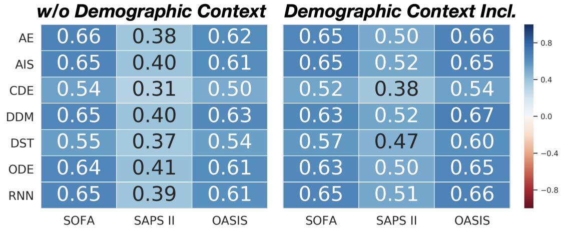
Table 1: Optimal model settings for each approach when predicting the SO. Models are trained with observations $\mathcal{O}$ and can be augmented with demographic context $\mathcal{D}$ or by the correlation regularization $\mathcal{C}$.

Approach	Best MSE	$\hat{d}_s$	Training Setting
AE	0.4804±0.001	64	w/ $\mathcal{O} + \mathfrak{D}$
AIS	0.4679± 0.004	64	$\mathcal{O} + \mathfrak{D}$
CDE	0.4887± 0.019	32	$\mathcal{O} + \mathfrak{D}$
DDM	0.4654± 0.002	128	$\mathcal{O} + \mathfrak{D}$
DST	0.5863± 0.013	64	$\mathcal{O}$
ODE	0.4698± 0.003	32	$\mathcal{O} + \mathfrak{D}$
RNN	0.4658± 0.002	64	$\mathcal{O} + \mathfrak{D}$

5.3. Qualitative analysis

The following analyses investigate the qualitative impact that the separate training strategies have on learned representations.

Representation-to-acuity score correlation We first evaluate the average correlation coefficient between the representations and derived acuity scores (see Section A.3 for more information). This is to demonstrate the capacity of the representations $\hat{S}_t$ to maintain clinically relevant information. We perform this analysis with and without the correlation regularization described in the previous subsection. The intention of this regularization is that the more positively correlated the representation is to the acuity scores, the more clinically informative the learned representation is. This was designed in hopes to improve SO prediction and policy learning yet there was no demonstrated advantage in doing so as shown in Figure 3 and Figure 14.



	w/o Demographic Context			*Demographic Context Incl.*		
	SOFA	SAPS II	OASIS	SOFA	SAPS II	OASIS
AE	0.66	0.38	0.62	0.65	0.50	0.66
AIS	0.65	0.40	0.61	0.65	0.52	0.65
CDE	0.54	0.31	0.50	0.52	0.38	0.54
DDM	0.65	0.40	0.63	0.63	0.52	0.67
DST	0.55	0.37	0.54	0.57	0.47	0.60
ODE	0.64	0.41	0.61	0.63	0.50	0.65
RNN	0.65	0.39	0.61	0.65	0.51	0.66

Figure 4: The average Pearson correlation coefficient between the state representations from each encoding approach and acuity scores. Shown here are the average coefficients when regularizing the learning process and demographic features are omitted (left) or are included (right) as input. For SAPS II and OASIS, the inclusion of demographic features when constructing the state representations results in higher correlation.

We show the average correlation coefficients of the learned state representations with acuity scores in Figure 4 for the two training settings where regularization is included (with and without demographic context). Unregularized representations fail to encode information that is correlated with the acuity scores (see Figure 15 in the Appendix). Between the two settings, representations are better correlated with the acuity scores when a patient's demographic context is included. This suggests that clinical acuity scores are strongly entangled with demographics features. Further investigation into the effects of this entanglement, including questions of fairness, is outside the scope of this study and is therefore a suggested element of future work.

Visualizing learned state representations Next, we use principal component analysis (PCA) to project the learned representations into a lower dimensional space. PCA embeddings are fit using the encoded representations for the entire test set but only the first and final representation from a patient trajectory are vizualized. To aid in connecting these two points, we have drawn a line between them colored by the patient outcome, survival (blue) vs. death (red).

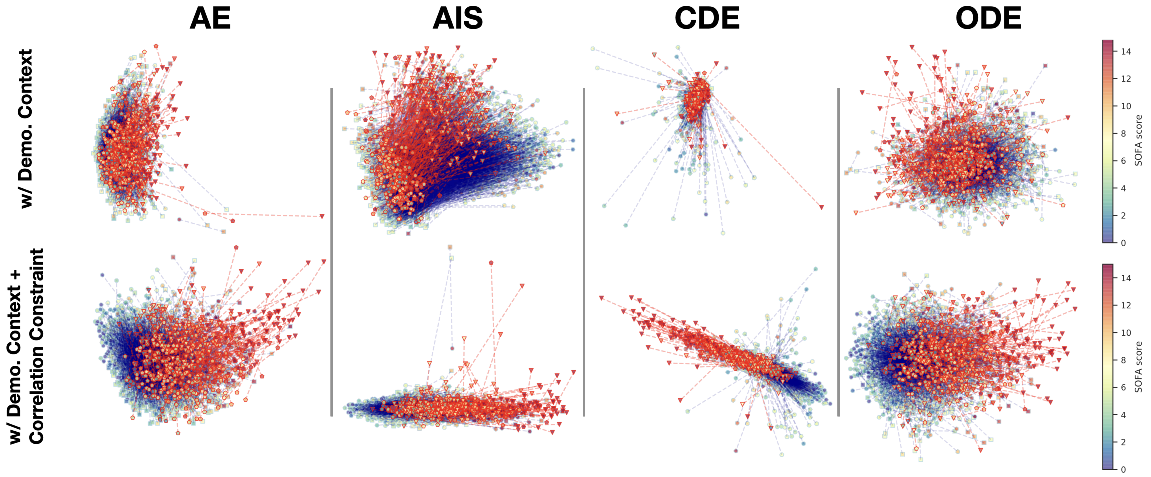


Figure 5: Representations of patient health, learned through a non-recurrent autoencoder (AE), Approximate Information State (AIS), a Neural CDE (CDE) and an ODE-RNN (ODE) (left-to-right, all other approaches are included in the Appendix, Section D.3) for two training settings. We show the first and final observations made of septic patients in the MIMIC-III dataset, colored by the SOFA score. Blue lines represent the trajectory of patients who recovered, while red lines connect observations of those those that did not.

As shown in Figure 1, PCA projections of raw observations are not separable. Separability is desirable because a representation that separates patients who are most at risk of death could be used to more easily facilitate prediction models. In Figure 5 we show PCA projections for AE, AIS, CDE and ODE in two training settings (remaining approaches and training settings in the Appendix, Section D.3). We focus on the role of including demographics without acuity regularization (top), and when it is included (bottom). With exception of AIS, regularization provides better separation between the patients that survive their sepsis infection and those that do not. Additionally, the regularization compresses the feature space of some encoding approaches. In combination with findings in Section 5.1, this compression suggests that the information prioritized via acuity regularization does not contribute to an improved state representations despite improved separability in representation space. Further analysis of the information content stored in the representations as a consequence of being regularized to correlate with acuity scores is a subject of future work.

5.4. Policy training and evaluation

We investigate the quality of treatment policies learned from the state representations via the approaches outlined in Section 4, following the procedure outlined in 4.4. We train policies using discretized BCQ, and evaluate with weighted importance sampling (WIS).

In Figure 6 we present the best performing policies learned from state representations. For each approach, excepting ODE, the top policies were learned from representations trained with the demographic context included as input to the encoding function ψ. The best ODE policy was developed from representations learned from the observations alone (see Figure 14 in the Appendix).

Among the various approaches, policies learned from representations encoded by the Neural CDE (CDE) far outperform the others. Simpler recurrent based architectures such as AIS and RNN also obtain higher performance

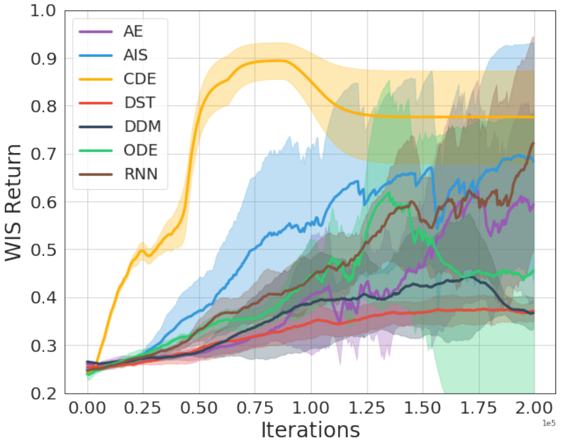


Figure 6: WIS evaluation of policies trained from the representations encoded by the architectures outlined in Section 4. Policies are trained from an experience replay buffer comprised of the training batch of patient trajectories for 200k iterations, evaluating the trained policy every 500 iterations. Results presented here are averaged over 5 random seeds, the shaded region measures a single standard deviation across seeds.

than the non-recurrent autoencoding baseline (AE). These results contribute toward the validation of our empirical hypothesis, that recurrent architectures provide better state representations in sequential partially observed settings. However, the AE based policy still learns a better policy than those based from far more complex methods (DST, DDM and ODE) signifying that the representations from these methods did not adequately encode sufficient information to learn a policy from in the batch setting, possibly due to dataset limitations.

6. Discussion

In this paper we have empirically evaluated seven information encoding approaches to develop sequential state representations of patient health, useful for learning effective treatment policies. We performed several experiments to determine characteristics useful for training state representations from noisy patient data that is inherently partially observed. To support the formation of informative representations we designed a supervised task where the representation implicitly encodes a history $\mathcal{H}_{t,t-1}$ of previous observations and actions to predict the next SO. This auxiliary task allowed us to investigate several properties of the representation space $\hat{S}$ based on decisions of how to execute the training.

In Section 5.1 we showed that higher dimensional representations reduce prediction accuracy, indicating the high capacity representations are not the most informative. In tandem we demonstrated that the inclusion of demographic context improves the learned state representations. This was verified (see Section D.1) when learning treatment policies. The best performing policies for each information encoding approaches presented in this paper were trained from representations learned with demographic context.

Future work In future work we intend to explore the use of multi-task learning (McDermott et al., 2020; Lin et al., 2019) to jointly train the representation space. Additionally, we plan to investigate methods that incorporate indicators of feature missingness and other underlying contextual variables (Agor et al., 2019; Fleming et al., 2019; Sharafoddini et al., 2019; Che et al., 2018; Lipton et al., 2016). We intend to study the effect these approaches have on the representation space, including a quantification of any performance reductions that may arise through use of demographic information encoding bias in the representations (Chen et al., 2018a).

The class of Neural Differential Equation methods (Chen et al., 2018b; Rubanova et al., 2019; Kidger et al., 2020) were developed to account for irregular time series with missing values and have demonstrated high performance in prediction tasks when provided feature sets with varying rates of missingness. Following the analyses performed in this paper, the Neural CDE appears promising for constructing state representations in the midst of the missingness and other irregularities inherent in healthcare data.

The conceptual separation between representation learning and policy learning in this paper was motivated by prior literature on state representation learning (Raffin et al., 2019). This choice allowed us to focus on the formation and analysis of the representation space $\hat{S}$. Another reason for this design choice was to enable straightforward use of current state of the art batch RL training algorithms. However, this decoupling is not necessary for developing off-policy sepsis treatment policies as discussed and demonstrated by Li et al. (2020). Another line of future work utilizing the findings of this paper is to similarly develop an end-to-end policy development approach that combines the objectives of the auxiliary tasks and RL algorithm, explicitly accounting for the state representation space as it encodes features of the expected outcome via the RL objective.

Additionally, it is necessary to more fully evaluate and interpret what the learned state representations encode and whether clinically relevant relationships are preserved (Bai et al., 2018). 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 SO.

Conclusion 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 reliable treatment policies, developed following a learning process that acknowledges the sequential and partial nature of the observations that are made.

This paper recommends possible ways of thinking of representation learning as a form of auxiliary task within policy development. Among the various research directions that are natural extensions from this work, we affirm the necessity of thoughtfully designing the representation learning process to honor the partial and sequential nature of the data generating process. These opportunities for learning optimal state representations for RL in healthcare offer an exciting new area of research that we anticipate being fruitful for establishing future advances in clinically relevant sequential decision making problems.

Acknowledgments

We thank our many colleagues who contributed to thoughtful discussions and provided timely advice to improve this work. We specifically appreciate the feedback provided by Nathan Ng, Vinith Suriyakumar and Karsten Roth.

This research was supported in part by Microsoft Research, a CIFAR AI Chair at the Vector Institute, a Canada Research Council Chair, and an NSERC Discovery Grant.

Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute www.vectorinstitute.ai/#partners.

References

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.

Pierre Baldi. Autoencoders, unsupervised learning, and deep architectures. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pages 37–49, 2012.

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

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.

Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):1–12, 2018.

Irene Chen, Fredrik D Johansson, and David Sontag. Why is my classifier discriminatory? In *Advances in Neural Information Processing Systems*, pages 3539–3550, 2018a.

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, 2018b.

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.

Scott Fujimoto, Edoardo Conti, Mohammad Ghavamzadeh, and Joelle Pineau. Benchmarking batch deep reinforcement learning algorithms. *arXiv preprint arXiv:1910.01708*, 2019a.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, pages 2052–2062, 2019b.

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.

Matthew Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps. In *2015 AAAI Fall Symposium Series*, 2015.

Milos Hauskrecht and Hamish Fraser. Planning treatment of ischemic heart disease with partially observable markov decision processes. *Artificial Intelligence in Medicine*, 18(3):221–244, 2000.

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.

Zina M Ibrahim, Honghan Wu, Ahmed Hamoud, Lukas Stappen, Richard JB Dobson, and Andrea Agarossi. On classifying sepsis heterogeneity in the icu: insight using machine learning. *Journal of the American Medical Informatics Association*, 27(3):437–443, 2020.

Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. In *International Conference on Learning Representations*, 2017.

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(1):1–9, 2016.

Alistair EW Johnson, David J. Stone, Leo A. Celi, 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.

Alan E Jones, Stephen Trzeciak, and Jeffrey A Kline. The sequential organ failure assessment score for predicting outcome in patients with severe sepsis and evidence of hypoperfusion at the time of emergency department presentation. *Critical care medicine*, 37(5):1649, 2009.

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

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. In *Advances in Neural Information Processing Systems*, 2020.

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.

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.

Luchen Li, Matthieu Komorowski, and Aldo A Faisal. The Actor Search Tree Critic (ASTC) for Off-Policy POMDP Learning in Medical Decision Making. *arXiv preprint arXiv:1805.11548*, 2018.

Luchen Li, Matthieu Komorowski, and Aldo A Faisal. Optimizing sequential medical treatments with auto-encoding heuristic search in POMDPs. *arXiv preprint arXiv:1905.07465*, 2019.

Luchen Li, Ignacio Albert-Smet, and Aldo A Faisal. Optimizing medical treatment for sepsis in intensive care: from reinforcement learning to pre-trial evaluation. *arXiv preprint arXiv:2003.06474*, 2020.

Xingyu Lin, Harjatin Baweja, George Kantor, and David Held. Adaptive auxiliary task weighting for reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 4772–4783, 2019.

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.

Siqi Liu, Kay Choong See, Kee Yuan Ngiam, Leo Anthony Celi, Xingzhi Sun, and Mengling Feng. Reinforcement learning for clinical decision support in critical care: Comprehensive review. *Journal of medical Internet research*, 22(7):e18477, 2020.

MingYu Lu, Zachary Shahn, Daby Sow, Finale Doshi-Velez, and Li-wei H Lehman. Is deep reinforcement learning ready for practical applications in healthcare? a sensitivity analysis of duel-ddqn for sepsis treatment. *arXiv preprint arXiv:2005.04301*, 2020.

A Rupam Mahmood, Hado P van Hasselt, and Richard S Sutton. Weighted importance sampling for off-policy learning with linear function approximation. In *Advances in Neural Information Processing Systems*, pages 3014–3022, 2014.

Matthew McDermott, Bret Nestor, Evan Kim, Wancong Zhang, Anna Goldenberg, Peter Szolovits, and Marzyeh Ghassemi. A comprehensive evaluation of multi-task learning and multi-task pre-training on ehr time-series data. *arXiv preprint arXiv:2007.10185*, 2020.

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.

Shamim Nemati, Mohammad M Ghassemi, and Gari D Clifford. Optimal medication dosing from suboptimal clinical examples: A deep reinforcement learning approach. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2978–2981. IEEE, 2016.

Ziad Obermeyer and Ezekiel J Emanuel. Predicting the future—big data, machine learning, and clinical medicine. *The New England journal of medicine*, 375(13):1216, 2016.

Sonali Parbhoo, Jasmina Bogojeska, Maurizio Zazzi, Volker Roth, and Finale Doshi-Velez. Combining kernel and model based learning for hiv therapy selection. *AMIA Summits on Translational Science Proceedings*, 2017:239, 2017.

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.

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.

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.

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.

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.

Susan M Shortreed, Eric Laber, Daniel J Lizotte, T Scott Stroup, Joelle Pineau, and Susan A Murphy. Informing sequential clinical decision-making through reinforcement learning: an empirical study. *Machine learning*, 84(1-2):109–136, 2011.

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.

Harold C Sox, Marshall A Blatt, Keith I Marton, and Michael C Higgins. *Medical decision making.* ACP Press, 2007.

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, 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.

Athanasios Tsoukalas, Timothy Albertson, and Ilias Tagkopoulos. From data to optimal decision making: a data-driven, probabilistic machine learning approach to decision support for patients with sepsis. *JMIR medical informatics*, 3(1):e11, 2015.

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.

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 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 (aged 18 years and older) 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 2.

A.3. 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). 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. 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) 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. OASIS requires only 10 features, without depending on laboratory measurements, diagnosis or comorbidity information.

Table 2: Observed features used for learning state representations

Time-varying continuous features

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

Demographic and contextual features

Age	Gender	Weight
Ventilation Status	Re-admission status	

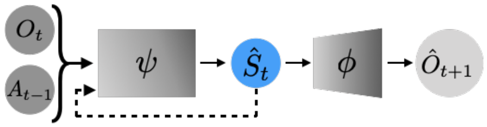


Figure 7: A basic RNN architecture for SO prediction

Appendix B. Architecture Details

We provide a comparative overview of the features that differentiate each approach in Table 3. Specific details about each architecture and how they are trained is included in the following subsections.

B.1. RNN

Recurrent Neural Networks (RNNs) are extensions of conventional feed-forward neural networks capable of receiving correlated sequences as input. The RNN 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 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 SO 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 SO $\hat{O}_{t+1}$.

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 SO.

Table 3: Overview of approaches for state representation learning under evaluation

Approach	Recurrent	Sequence as input	Num. Parameters
AE			$27k - 76k$
AIS	✗		$28k - 339k$
CDE	✗	✗	$78.9k - 1.78m$
DDM	✗		$6k - 1.25m$
DST	✗	✗	$47k - 256k$
ODE	✗		$48.3k - 329k$
RNN	✗		$26k - 337k$

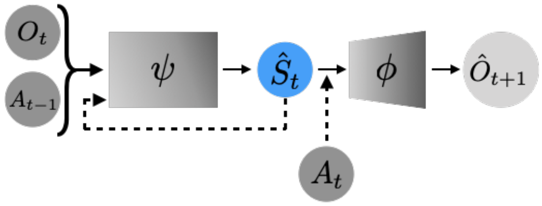


Figure 8: AIS architecture, adapted from Subramanian and Mahajan (2019)

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 regularizing the training to correlate with acuity scores are all set to 100.

B.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 SO $\hat{O}_{t+1}$.

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 multivariate Gaussian distribution which is then used to predict the SO.

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 regularizing the training to correlate with acuity scores are all set to 100.

B.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 9, for the purpose of predicting the SO 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 $\hat{S}_t$.

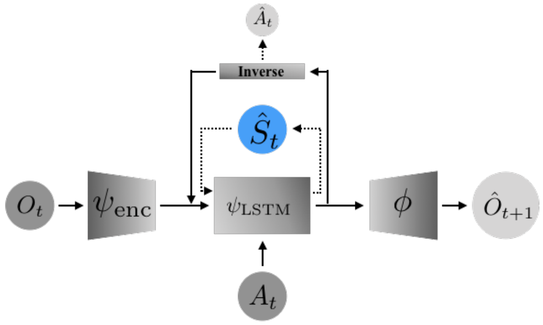


Figure 9: The Decoupled Dynamics Module from Zhang et al. (2018), adapted from its original presentation

This state representation is then provided to the decoder function ϕ to provide a prediction of the SO $\hat{O}_{t+1}$. 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. 9) along with the true SO 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)[3].

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 SO. 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 SO, $z_t = \psi_{\text{enc}}(O_t)$, $z_{t+1} = \psi_{\text{enc}}(O_{t+1})$ 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 SO $\hat{z}_{t+1}$. 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 subsequent 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 SO $\hat{O}_{t+1}$.

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 regularizing the training to correlate with acuity scores are all set to 0.25.

B.4. DST

As outlined by Bonnier 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.

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

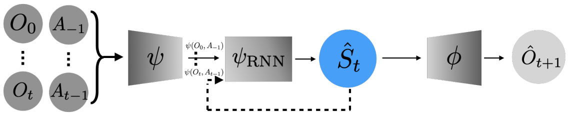


Figure 10: The Deep Signature Transform architecture for SO prediction

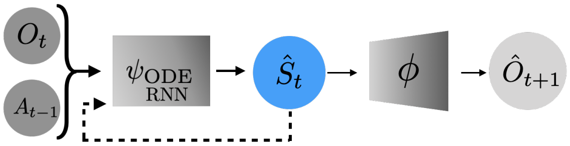


Figure 11: The ODE-RNN architecture for SO prediction

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 $|\text{Sig}^N|= \frac{d^{N+1}-1}{d-1}$.

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. This sequence is then passed through a recurrent neural network to produce the learned state representation $\hat{S}_{0:t}$. This state representation is then passed through the decoder ϕ to predict the SO $\hat{O}_{t+1}$.

Implemented using the Signatory library[4], this is essentially equivalent to using the signature transformation as a stream preserving non-linear transformation layer in a neural network.

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 (Bonnier 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.

In the encoder, we start with two pointwise one-dimensional convolutional layers (with a kernel size of 1) to add 8 augmented features to the 63 dimensional input vector. We then apply a stream preserving signature transformation with a depth of 2. The latent states are obtained by passing the output of the signature transform through a 2 layer GRU with *dim* hidden units, where *dim* is the chosen embedding dimension.

For estimating the decoder function ϕ, we again use two pointwise one-dimensional convolutional layers (with a kernel size of 1) with filter sizes of 64 and 32 respectively. Then, we apply a stream preserving signature transformation of depth 2. Finally, We use a pointwise 2-layer feed-forward neural network with sizes $(|Sig^N|, 64)$, and $(64, 33)$ with ReLU activation.

The best DST architectures for each choice of $\hat{d}_s$ were trained for 50 epochs with a learning rate of 10^{-3}. The λs for regularizing the training to correlate with acuity scores are set to 1.

B.5. ODE-RNN

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., 2018b) framework. An ODE-RNN is a recurrent neural network where the hidden states between observations evolve according to a parameterized ODE[5].

Although ODE-RNNs are natively able to handle missing values and irregularly sampled time series, for the purposes of this paper, we still use imputed, time-binned data for this model.

For the encoder, we use a GRU with 50 units, where the hidden states between observations are

[4] https://github.com/patrick-kidger/signatory

[5] To implement the ODE-RNN, we use the code available at https://github.com/YuliaRubanova/latent_ode

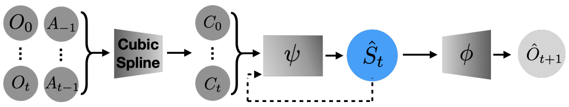


Figure 12: The Neural CDE architecture for SO prediction

modelled by a Neural ODE parameterized by a 2-layer MLP with 50 hidden units. We use the adaptive stepsize `dopri5` solver. For the decoder, we use an MLP applied at each time step, consisting of 3 layers, with sizes $(dim, 100)$, $(100, 100)$, $(100, 33)$ and ReLU activations.

The best ODE-RNN architectures for each setting of $\hat{d}_s$ were trained for 100 epochs with a learning rate of 10^{-3}. The λs for regularizing the training to correlate with acuity scores are all set to 1.

Note that though latent ODE representations have shown better performance in representing time series data, we do not believe that a latent ODE is appropriate for this task. This is because the encoder of a latent ODE involves an ODE-RNN running backwards in time over the inputs to obtain a probability distribution over z_0. Thus, the initial latent state would contain information about all subsequent observations and actions (similar to if a bidirectional RNN were used). A sampled value of z_0 is then used as the initial condition in a Neural ODE to solve for $z_{1:T}$. This information leakage would result in an unrealistic estimate of the SO prediction error.

B.6. CDE

Similar to ODE-RNNs (Rubanova et al., 2019), Neural Control Differential Equations (CDEs) (Kidger et al., 2020) model temporal dynamics by parameterizing the time derivative of the hidden states by a neural network. Unlike ODE-RNNs, the hidden states in CDEs evolve smoothly as a function of time, even at time points when data is observed. To accomplish continual dependence on the data throughout the latent trajectory, cubic spline interpolation is used, and the network operates on pre-computed cubic spline coefficients instead of the actual observations. The initial value for the latent space is calculated by a linear map on the inputs at $t = 0$. It has been shown that CDEs are universal approximators from sequences in $\mathbb{R}^D$ to real valued targets.

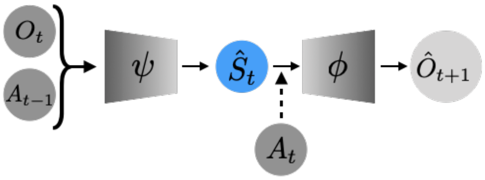


Figure 13: The Autoencoder architecture

For the encoder, we use a Neural CDE parameterized by an MLP with four hidden layers, each of which has 100 hidden units. We use ReLU activation for the hidden layers, and a tanh activation for the final layer. For the decoder, we use an MLP applied at each time step, consisting of 3 layers, with sizes $(dim, 100)$, $(100, 100)$, $(100, 33)$ and ReLU activations.

The best CDE architectures for each setting of $\hat{d}_s$ were trained for 200 epochs with a learning rate of 2×10^{-4}. The λs for regularizing the training to correlate with acuity scores are all set to 1.

B.7. Autoencoder

To isolate the contribution of the recurrent layer in the RNN (Sec. B.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. B.2), we concatenate the current action A_t to $\hat{S}_t$ when predicting the SO $\hat{O}_{t+1}$ using the decoder function ϕ. The autoencoder architecture shown in Figure 13 was trained using same loss function as the RNN, AIS, and DST approaches.

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 SO, $\hat{S}_t$ 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 the loss function, with the option to be regularized 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 regularizing the training were all set to 100.

Appendix C. State construction in prior work

See Table 4 for an overview of how prior work has constructed state representations for RL in healthcare settings.

Appendix D. Additional experimental results

In this section we include a more exhaustive accounting of the experimental results that did not fit within the space constraints of the main body of the paper.

D.1. Policy Training

We train policies on each of the learned state representations outlined in Section 4.2. As we do not have the ability to generate more data through an exploration of novel treatment strategies, we develop a policy using offline, batch reinforcement learning. In this setting, it is critical that the estimated value function not extrapolate to actions that are not present in the provided data (Gottesman et al., 2019). To counter this error caused by extrapolation, Fujimoto et al. (2019b) developed an algorithm for continuous control settings that truncates any Q-function estimate corresponding to actions that fall outside the support of the dataset. This algorithm, Batch Constrained Q-Learning (BCQ) was then adapted and simplified by the authors for use in discrete action settings (Fujimoto et al., 2019a).

As the patient cohort that we have to learn policies from is defined with discrete actions, we use the simplified Batch Constrained Q-Learning (BCQ) for discrete action settings (Fujimoto et al., 2019a) to learn treatment policies from state representations $\hat{S}$. We train the policies using the encoded training subset of our data, validating the performance of the policy using the testing subset via weighted importance sampling (WIS), following Li et al. (2019). The WIS return for each policy throughout training is computed by: $R^{\text{WIS}} = \frac{\sum_n^N w_n R_n}{\sum_n^N w_n}$, where the w_n are the per-trajectory IS weights and R_n is the observed outcome of the trajectory.

WIS evaluation of policies trained from the representations encoded by the architectures outlined in Section 4. The Q-network used in our implementation of BCQ was comprised of 3 fully connected layers, using 64 nodes per layer (excepting for the DDM architecture where we used 128 nodes per layer). The learning rate was empirically tuned for each training approach in a log-uniform range of $\{1e-5, 1e-2\}$. The best policies for all approaches, excepting CDE, used a learning rate of $1e-3$. CDE used a learning rate of $1e-5$. The BCQ action eliminiation threshold τ was set to 0.3 for all experiments.

All policies were trained[6] from a uniformly sampled experience replay buffer comprised of the training batch of patient trajectories for 200k iterations, evaluating the trained policy every 500 iterations using the testing subset of the patient data.

The behavior policy used in WIS was derived via behavior cloning using a two layer fully connected neural network trained with a supervised cross entropy loss using the stored actions with corresponding actions. WIS evaluation was performed by using the observation drawn from the test set of the patient data, predicting the observed action using the approximated behavior policy and then comparing with the inferred action provided by the current policy trained with BCQ using the corresponding state representation encoded by the user's choice of information encoding approach.

In Figure 14, we present the evaluations of policies learned from the representations learned through each information encoding approach. Each subfigure features the policy performance based on the training strategy used to learn the state representations.

[6] We adpated Fujimoto et al. (2019a)'s code which can be accessed at: `https://github.com/sfujim/BCQ/tree/master/discrete_BCQ`

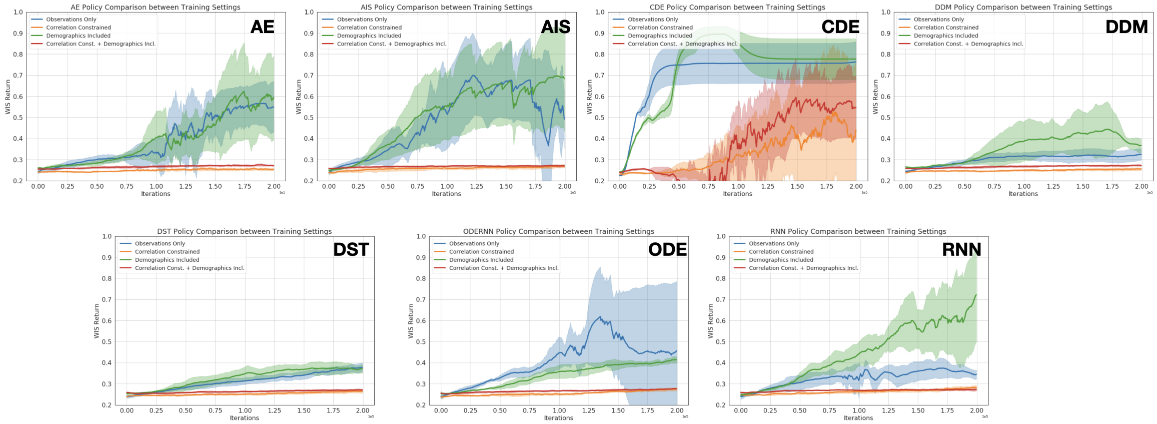


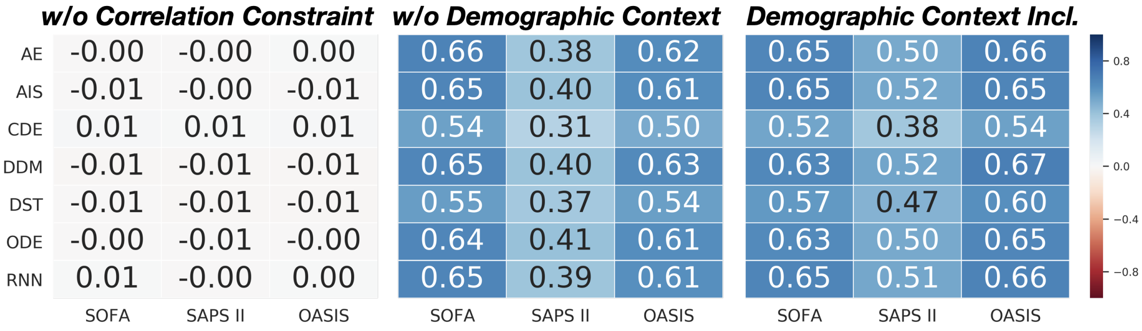
Figure 14: A compilation of the policy learning curves for each representation training setting for all encoding approaches investigated in this paper.

D.2. Analysis of correlation coefficient between representations and acuity scores

Here we present in Figure 15 the average correlation coefficients between the acuity scores and learned state representations from the various information encoding approaches. What is compared here is the effect of representation learning process on the subsequent correlation coefficients.

D.3. 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 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.



	w/o Correlation Constraint			w/o Demographic Context			Demographic Context Incl.		
	SOFA	SAPS II	OASIS	SOFA	SAPS II	OASIS	SOFA	SAPS II	OASIS
AE	-0.00	-0.00	0.00	0.66	0.38	0.62	0.65	0.50	0.66
AIS	-0.01	-0.00	-0.01	0.65	0.40	0.61	0.65	0.52	0.65
CDE	0.01	0.01	0.01	0.54	0.31	0.50	0.52	0.38	0.54
DDM	-0.01	-0.01	-0.01	0.65	0.40	0.63	0.63	0.52	0.67
DST	-0.01	-0.01	-0.01	0.55	0.37	0.54	0.57	0.47	0.60
ODE	-0.00	-0.01	-0.00	0.64	0.41	0.61	0.63	0.50	0.65
RNN	0.01	-0.00	0.00	0.65	0.39	0.61	0.65	0.51	0.66

Figure 15: The average Pearson correlation coefficient between the state representations from each encoding approach and acuity scores. Shown here are the average coefficients when the representation learning process is unregularized (left), when demographic features are omitted (center) or are included (right). The inclusion of demographic features when constructing the state representations causes them to be more correlated. When the state representations are uncoorelated. They fail to embed information directly correlated with the derived acuity scores.

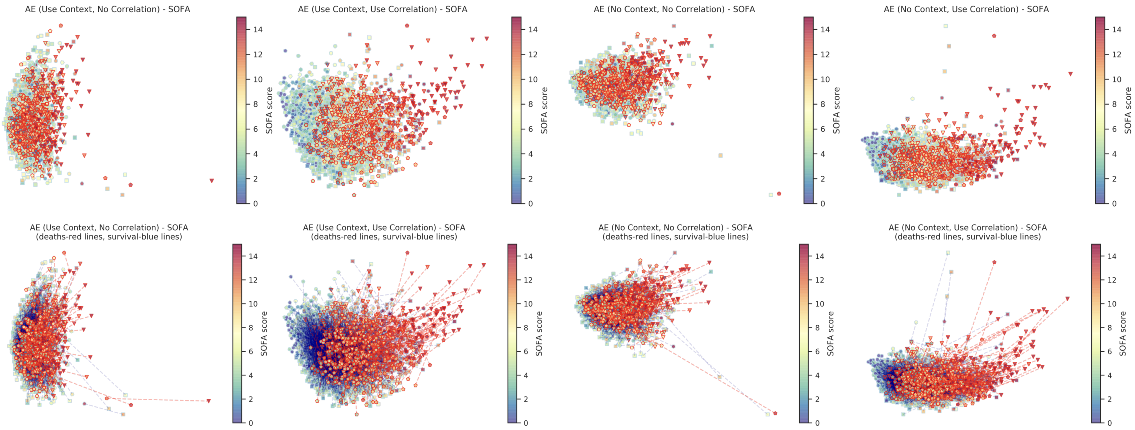


Figure 16: Representations of patient health, learned through an Autoencoder (AE)

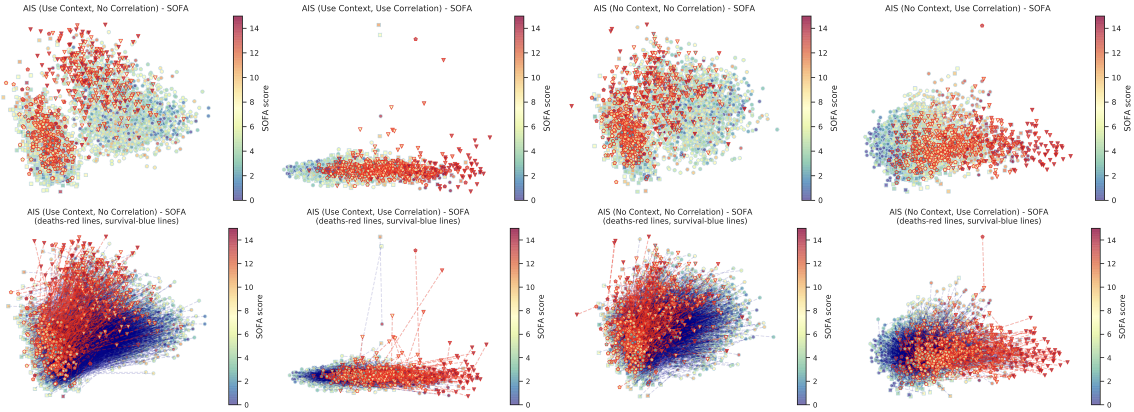


Figure 17: Representations of patient health, learned through Approximate Information State (AIS)

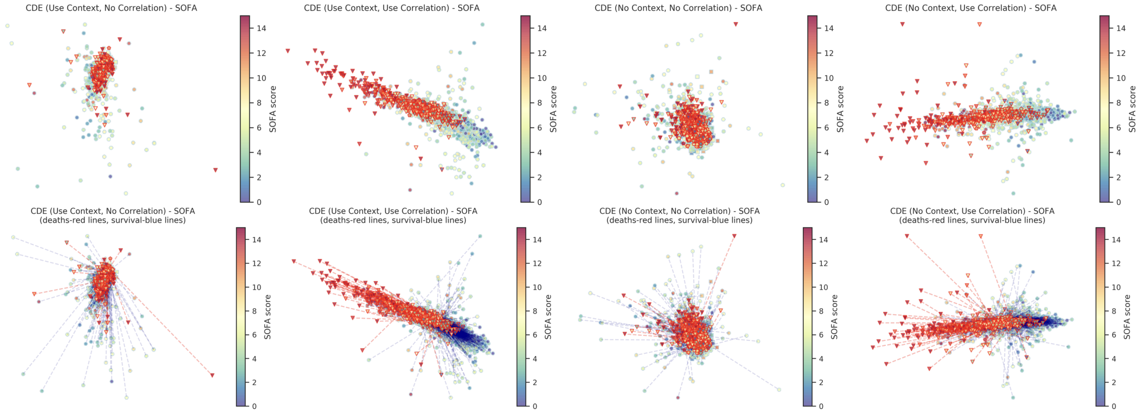


Figure 18: Representations of patient health, learned through the Neural CDE (CDE)

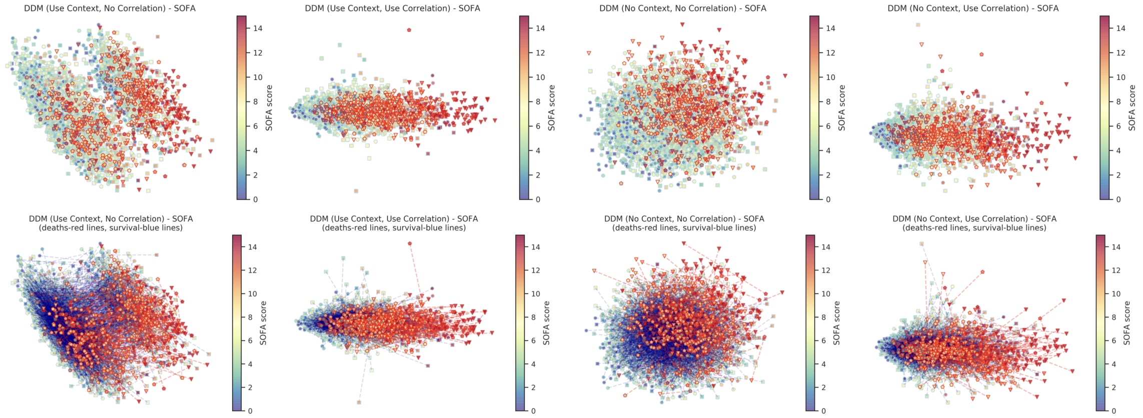


Figure 19: Representations of patient health, learned through the Decoupled Dynamics Module (DDM)

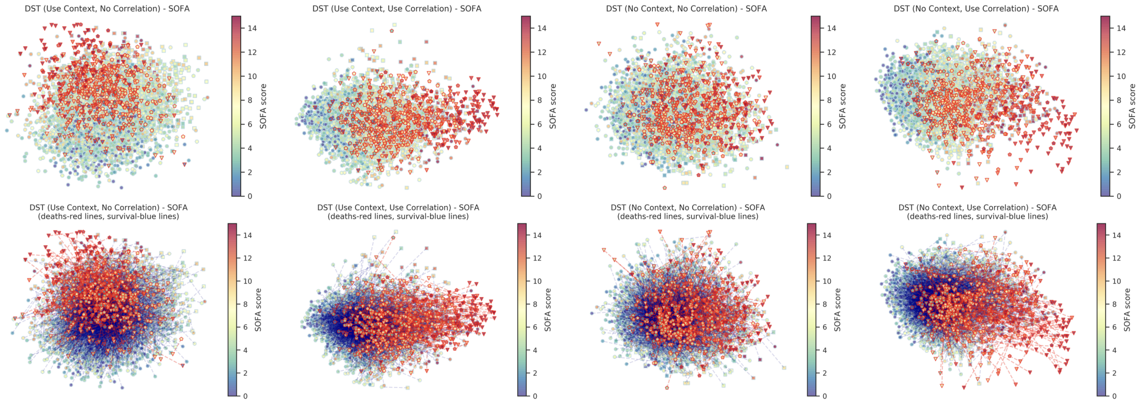


Figure 20: Representations of patient health, learned through the Deep Signature Transform (DST)

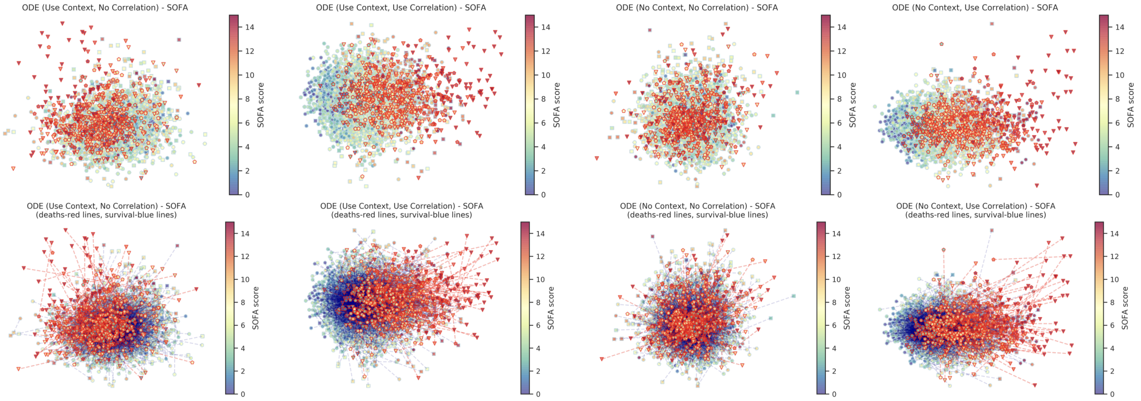


Figure 21: Representations of patient health, learned through the ODE-RNN (ODE).

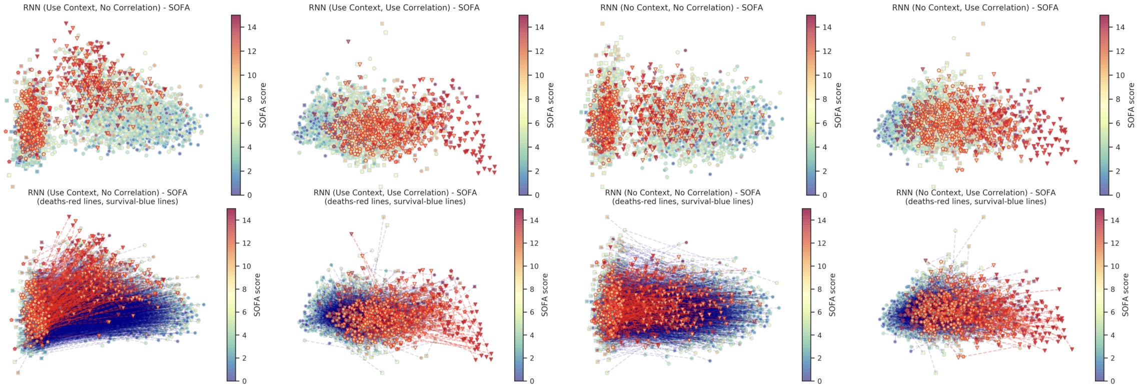


Figure 22: Representations of patient health, learned through a recurrent autoencoder (RNN)

Table 4: State construction for RL in healthcare - background

Ref	Domain	State Construction
Hauskrecht and Fraser (2000)	Heart Disease Management	10 categorical variables + alive/dead; constructed hierarchically
Guez et al. (2008)	Epilepsy	114 dimensional continuous - summarizing past EEG activity
Shortreed et al. (2011)	Schizophrenia Treatment	20 demographic + 30 time varying; imputed using fully conditional specification
Tsoukalas et al. (2015)	Sepsis	9 states constructed from vitals based on medical criteria
Nemati et al. (2016)	Medication dosing	Estimated using discriminative hidden Markov model on 21 continuous vitals + 6 binary demographics
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)
Parbhoo et al. (2017)	HIV Treatment	7 hidden discrete physiological states from Bayesian model-based RL over 80 observations
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
Li et al. (2018)	Sepsis (MIMIC-III)	5 demographic + 46 time varying ; modelled as Gaussian mixture
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.
Li et al. (2019)	Sepsis (MIMIC-III)	Auto-encoding SMC over 48 patient variables