An Empirical Study of Representation Learning for Reinforcement Learning in Healthcare

Taylor W. Killian[a,b], Haoran Zhang[a,b], Jayakumar Subramanian[c], Mehdi Fatemi[d], Marzyeh Ghassemi[a,b]

a-University of Toronto, b-Vector Institute, c-Adobe Research India, d-Microsoft Research

Motivation

Sequential observations of patient health are partial in nature.

- Reinforcement Learning has become a popular framing to learn models for sequential decision making within healthcare.
- Prior work applying RL to healthcare neglects learning state representations, choosing instead to construct time-independent states with "raw" observations.

Empirical Hypothesis:

- We can learn better state representations by honoring the nature of the true data generating process (sequential and partially observed).

Data: MIMIC-III Sepsis Cohort

Focusing on sepsis treatment, we build from the patient cohort defined by Komorowski, et al [2018].

- To isolate continuous time-varying features of patient health, we remove all binary or categorical variables. We denote these features as observations $\mathcal{O}$.
- Features such as gender, age, weight, etc. that contribute to defining a patient's demographic context are set aside. These features are denoted by $\mathcal{D}$.
- Acuity scores (SOFA, SAPSII and OASIS) are computed at each time step to be available to constrain the learning process. This collection is denoted as $\mathcal{C}$.

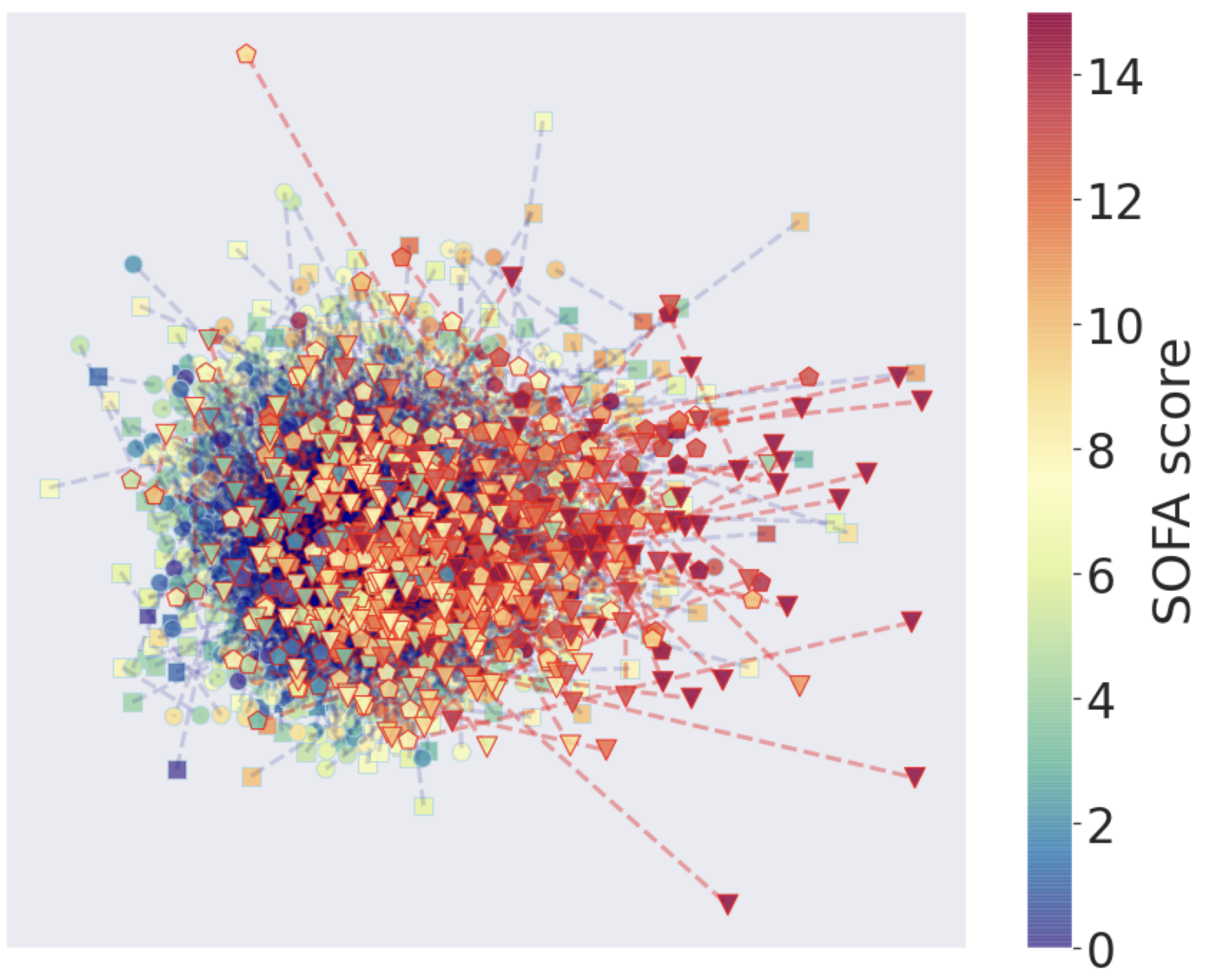


Figure: "Raw" first and final observations in MIMIC-III, colored by SOFA score and embedded with PCA. Blue lines connect observations of patients who survived, red lines signify those that did not.

Learning an Information State

What is the best way to encode sequential data to facilitate the best performing policies?

Given a history $\mathcal{H}_{t,t-1}$ of observations $O_{1:t}$ and actions $A_{1:t-1}$, we want to learn an information state $\hat{S}_t$. We can then use this state representation for policy learning. We learn $\hat{S}_t$ through an auxiliary task of predicting the subsequent observation O_{t+1}:

- Encode $\mathcal{H}_{t,t-1}$ with a function ψ to produce $\hat{S}_t$.
- Using the state representation $\hat{S}_t$, predict O_{t+1} via a decoding function ϕ.

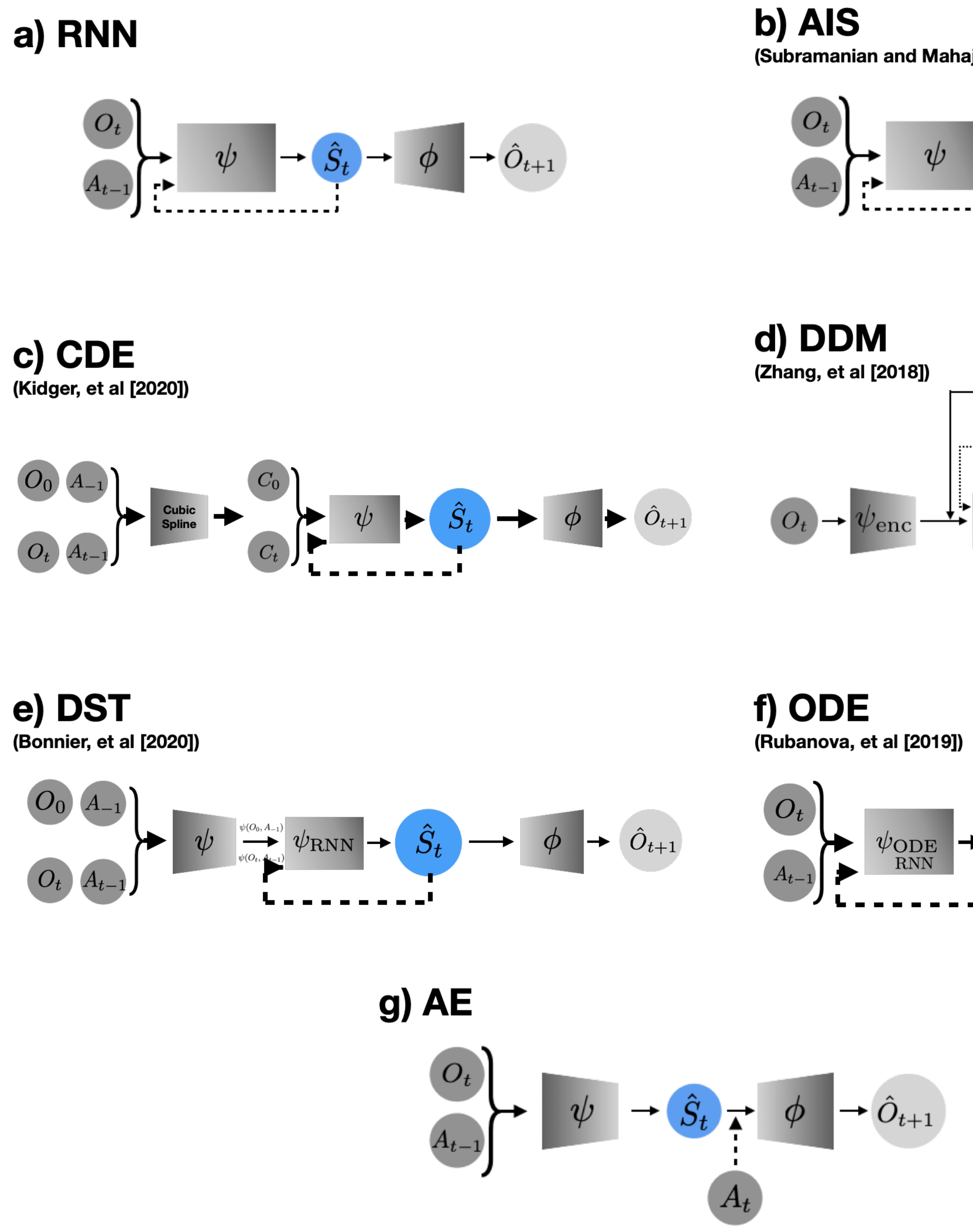
That is, we learn informative state representations using the following general procedure: $\mathcal{H}_{t,t-1} \xrightarrow{\psi} \hat{S}_t \xrightarrow{\phi} \hat{O}_{t+1}$

We can choose to augment this training by:

- Appending demographic context $\mathcal{D}$ to the observations, or
- Constraining the loss by the correlation of $\hat{S}_t$ with the acuity scores $\mathcal{C}$.

Information Encoding Models

We compare the representations learned via several encoding architectures when predicting the subsequent observation.



Predicting the Subsequent Observation

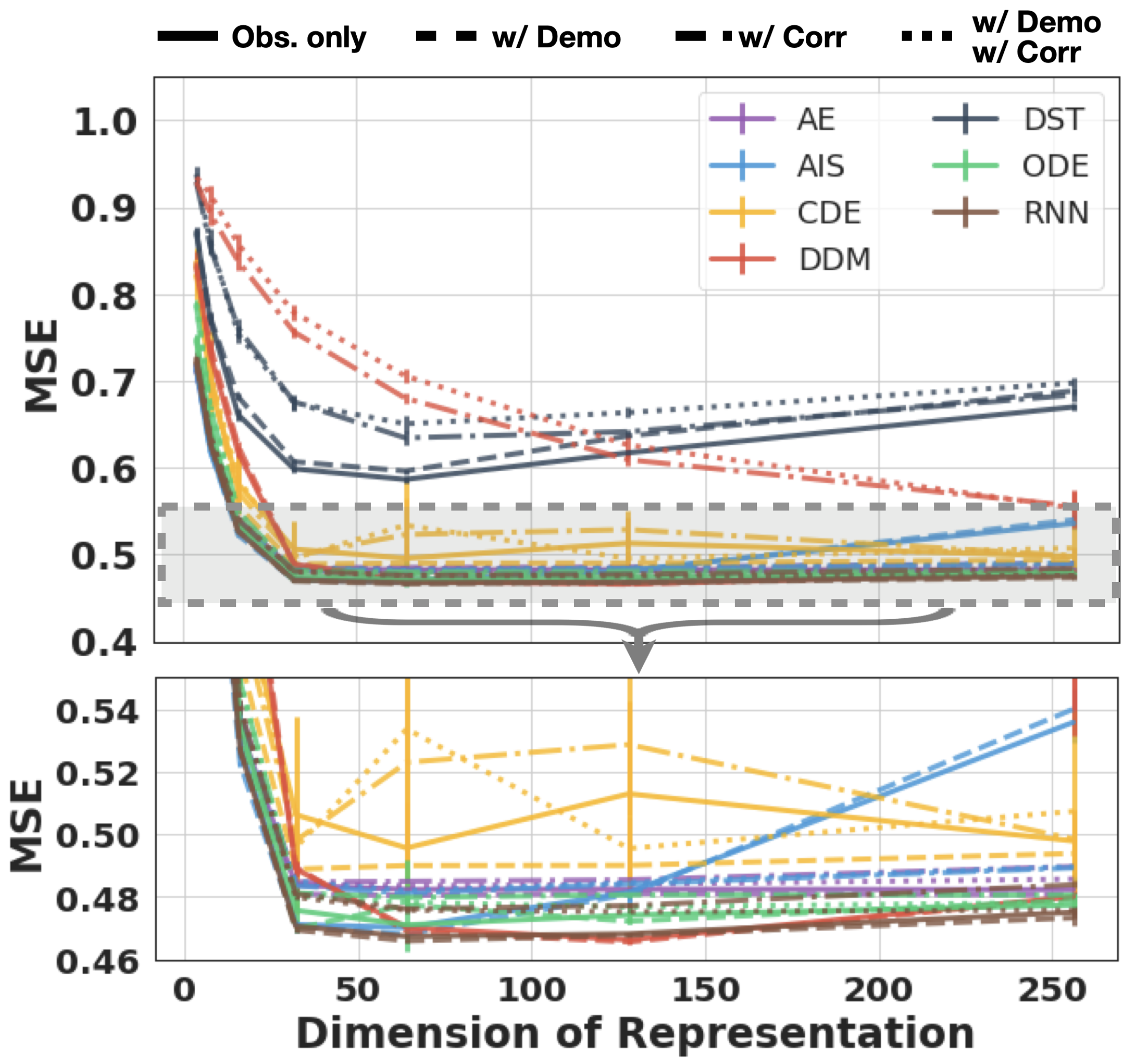


Figure: Mean squared error for SO prediction as a result of varying $\hat{d}_S$, comparing various training settings. Note that augmenting the input with demographic context generally improves prediction performance.

Visualizing the Learned State Representations

Using PCA we visualize the representations learned from the best performing setting of each approach.

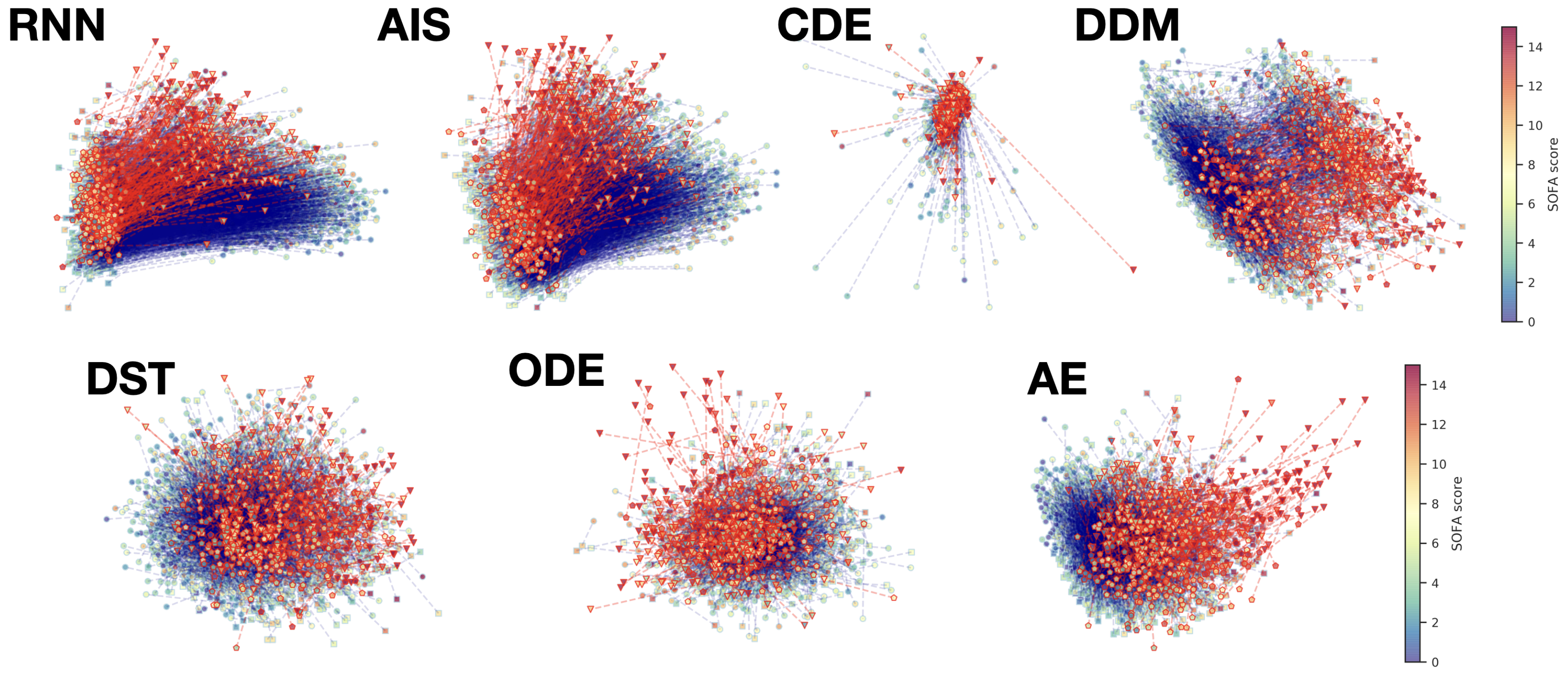


Figure: A composite image of the PCA embedding of the first and final representations learned for each patient trajectory via the encoding models under consideration in this paper. Blue lines connect the representations of patients who survived while red lines signify those that did not.

- Several of these encoding approaches form separable representations between the groups of patients with similar outcomes facilitating better SO prediction performance.
- Generally, these representations also contribute to better policy learning, outperforming the non-recurrent autoencoder baseline (AE).

Learning a Treatment Policy

Using the representations learned through the various encoding models, we train policies using Batch Constrained Q-learning (BCQ).

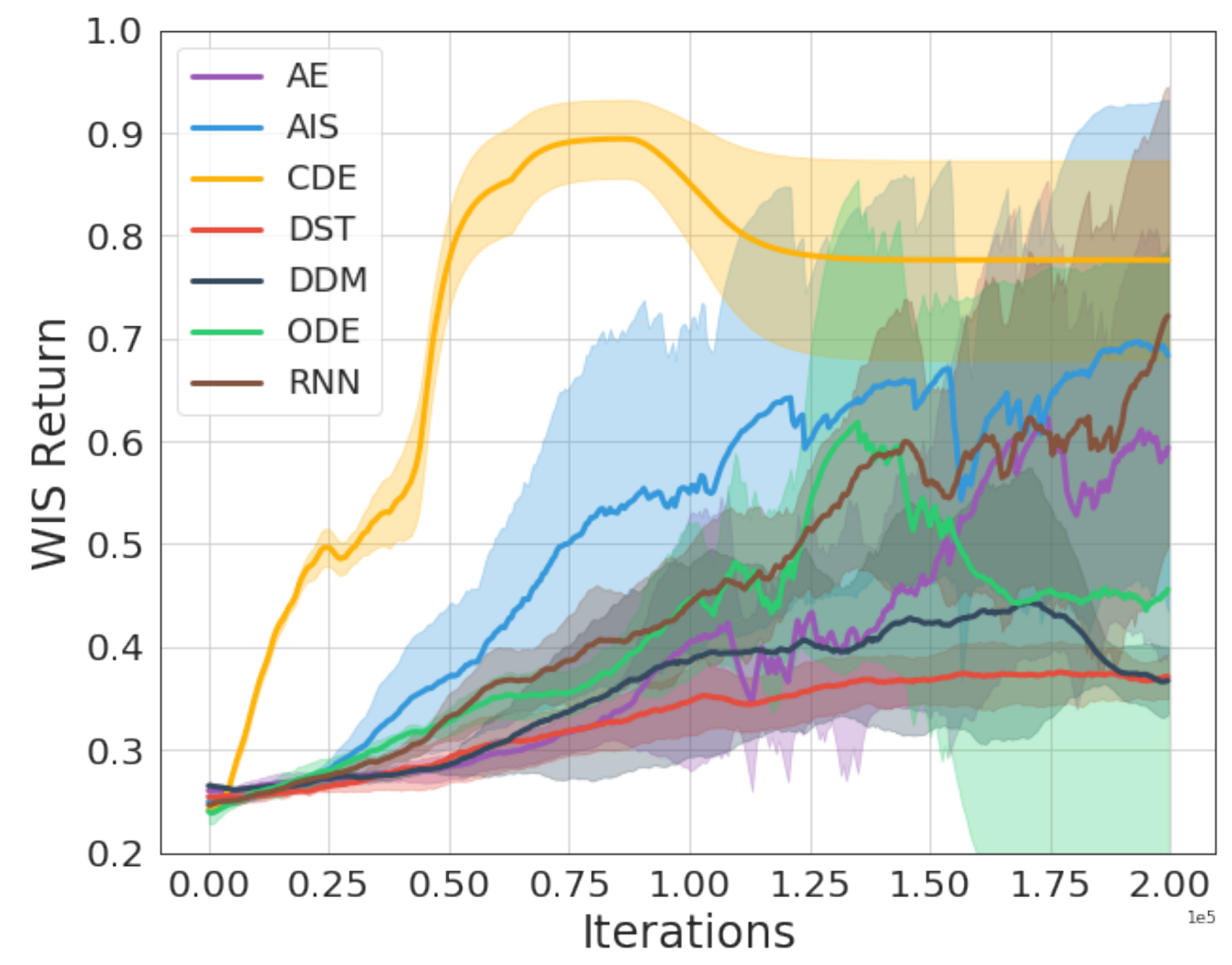


Figure: WIS evaluation of policies trained from the representations encoded by the architectures under consideration, as trained by the discrete form of BCQ (Fujimoto, et al [2019]).

Additional Questions?

Please don't hesitate to reach out via email: `twkillian@cs.toronto.edu`

All code used to define and train the models, including the dBCQ policy and WIS evaluation can be found at: `https://github.com/MLforHealth/rl_representations`.