Interpretable Skill Learning for Dynamic Treatment Regimes through Imitation

Yushan Jiang[1], Wenchao Yu[2], Dongjin Song[1], Wei Cheng[2], Haifeng Chen[2]
[1]Department of Computer Science and Engineering, University of Connecticut, Storrs, CT, USA
[2]Data Science & System Security, NEC Laboratories American, Princeton, NJ, USA
[1]{yushan.jiang, dongjin.song}@uconn.edu
[2]{wyu, weicheng, haifeng}@nec-labs.com

***Abstract*—Imitation learning that mimics experts' skills from their demonstrations has shown great success in discovering dynamic treatment regimes, *i.e.*, the optimal decision rules to treat an individual patient based on related evolving treatment and covariate history. Existing imitation learning methods, however, still lack the capability to interpret the underlying rationales of the learned policy in a faithful way. Moreover, since dynamic treatment regimes for patients often exhibit varying patterns, *i.e.*, symptoms that transit from one to another, the flat policy learned by a vanilla imitation learning method is typically undesired. To this end, we propose an Interpretable Skill Learning (ISL) framework to resolve the aforementioned challenges for dynamic treatment regimes through imitation. The key idea is to model each segment of experts' demonstrations with a prototype layer and integrate it with the imitation learning layer to enhance the interpretation capability. On one hand, the ISL framework is able to provide interpretable explanations by matching the prototype to exemplar segments during the inference stage, which enables doctors to perform reasoning of the learned demonstrations based on human-understandable patient symptoms and lab results. On the other hand, the obtained skill embedding consisting of prototypes serves as conditional information to the imitation learning layer, which implicitly guides the policy network to provide a more accurate demonstration when the patients' state switches from one stage to another. Thoroughly empirical studies demonstrate that our proposed ISL technique can achieve better performance than state-of-the-art methods. Moreover, the proposed ISL framework also exhibits good interpretability which cannot be observed in existing methods.**

***Index Terms*—imitation learning, prototype, interpretable machine learning, dynamic treatment regimes**

I. Introduction

Dynamic treatment regime (DTR) is a set of sequential treatment decision rules that intend to provide individualized and effective treatments for patients [1], [2]. To tackle the complex DTR tasks, one important research direction is based on imitation learning [3]–[7]. Compared to the reinforcement learning methods for DTR [8] where the explicit reward signal from the handcraft rules is sparse and typically not optimal, imitation learning based methods directly learn a mapping between states and actions to replicate expert behavior from demonstrations. Recent studies typically learn the distribution of expert demonstrations in an adversarial manner, where the policy network serves as the generator and is optimized by the reward signals given by one or more discriminators based on the demonstrations [3], [4]. Although these methods have shown their effectiveness in DTR tasks, there are still two key challenges for imitation learning. First, the treatments given by the learned policies are not trustworthy as their underlying rationales are not interpretable. This is extremely critical for real clinical scenarios. Second, DTR tasks often illustrate obvious variability in expert demonstrations, where a flat policy is insufficient to handle the scenarios when patients' symptoms transit from one to another.

To resolve the aforementioned challenges, skill learning methods through hierarchical imitation provides a feasible solution by decomposing a complex decision-making process into multiple lower-level subgoals [6], [7]. By jointly learning a set of low-level policies taking primitive actions in each subgoal and a high-level policy controlling the subgoals, the agent is empowered to capture the varying patterns and recommends more accurate treatments accordingly. In addition to building the hierarchy within a trajectory, researchers also succeeded in modeling the inter-trajectory variation across different demonstrations [5]. In general, the existing methods that model the variation structure of expert demonstrations inherently provide abstract explanations, either by learning a latent variable via the regularization of information-theoretic measures [5], [6], or by learning a subgoal representation with multiple constraints [7]. However, the latent variables or representations cannot provide an explicit explanation for the suggested skills in varying patterns as they do not have clear definitions in terms of clinical. This is undesired as DTR needs a clear and interpretable structure for the reasoning process.

Recent developments in interpretable sequence modeling [9], [10] have revealed the potential to meet the interpretability requirement in DTR. This class of method typically learns the prototypes that are defined as the exemplar sequences or segments in the downstream sequence classification tasks (on natural language data and time series data). The prediction is determined based on the similarity scores between each input and all learned prototypes in the embedding space (obtained via a sequence encoder). In the reasoning process, the model predictions are interpreted by the top similar prototypes. In general, the existing interpretable sequence modeling methods enable faithful explanations of the model predictions with competitive performance compared to other state-of-the-art black box models.

Inspired by their success, we first propose to leverage the interpretable sequence modeling framework in imitation learning for DTR tasks. As the expert's trajectory in DTR is a sequence of patient states with treatments, it is a perfect fit for the aforementioned framework. Note that the aforementioned variation modeling and sequence interpretation modeling methods formulate the task with different granularity, including timestep level [6], segment level [7], [10], and trajectory level [5], [9]. We choose to perform the learning and inference with reasoning at the segment level. Compared to other formulations, the skills learned at the segment level are able to capture the temporal variability of states and are transferable across different trajectories. By aggregating the learned prototypes at each segment, the obtained skill serves as conditional information that implicitly guides the policy network to differentiate varying patterns and provide more accurate treatment recommendations. Meanwhile, the suggested skills can be interpreted at the segment level by tracing back to the exemplar segments. Our main contributions are summarized as follows:

- We propose an interpretable skill learning model, ISL, to learn optimal treatment policies, which exploits the segment-level expert demonstrations and results in representative and more transferable skills across different trajectories.
- The ISL model learns to capture exemplar segments and construct faithful skill embedding for an imitation learning task, which is innovative compared to the existing interpretable sequence modeling methods and imitation learning methods.
- Empirical studies on a benchmark DTR dataset demonstrate that the ISL model provides better performance compared to state-of-the-art models, as well as reasonable explanations for the recommended treatments.

II. Methodology

In this section, we introduce the interpretable skill learning framework, including the basic problem formulation, model architecture, learning objectives, and the training process.

A. Problem formulation

A vanilla imitation learning task for DTR can be described as follows: Given a set of doctors' demonstration trajectories, each of which consists of a sequence of state-action pairs (s_t, a_t), where s_t denotes the patient state and a_t indicates the medication taken by the doctor at timestep t, the goal is to learn a policy that can replicate the doctors' medications. Note that the imitation learning model is built upon step-level demonstrations without consideration of the evolving patient symptoms and corresponding prescriptions.

Our proposed skill learning framework formulates the task at the segment level to exploit the sequential treatment demonstrations: firstly, we split each trajectory into different segments and perform imitation learning on one segment at a time, by hypothesizing that the skill at the segment level is representative and more transferable across different trajectories; secondly, each state in the segment comes along with a fixed number of previous states from the same trajectory, which also forms a input segment to exploit the dynamics of historical patient status up to the current state.

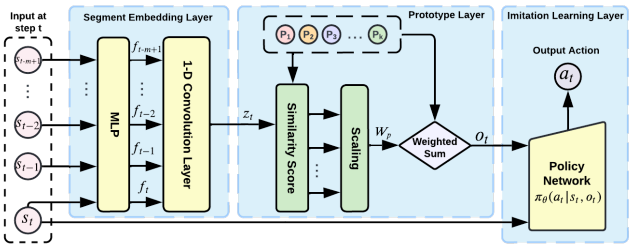

Fig. 1. The architecture of our proposed ISL model

More formally, given a set of disjoint segments split from the original trajectories, $\{[(s_t^{(j)}, a_t^{(j)})]_{t=1}^{m}\}_{j=1}^{n}$, where m is the fixed length of segments, n is the number of segments, we aim to learn a set of prototypes representing exemplar segments in the training data, which can be assembled as a skill embedding to the facilitate the imitation learning of DTR at each step t. Meanwhile, the skill embedding of the given segment can be interpreted based on the most similar prototypes.

B. Interpretable Skill Learning Model

In this section, we introduce the architecture of our interpretable skill learning model that consists of three learning components: a segment embedding layer, a prototype layer, as well as an imitation learning layer, as shown in Figure 1. Note that we omit the superscript j (representing the order of instance) in this subsection for simplicity.

1) Segment Embedding Layer: In each disjoint segment, given the input state at step t with m-1 steps of historical states $[s_{t-m+1}, \cdots, s_{t-2}, s_{t-1}, s_t] \in \mathbb{R}^{m \times d}$, where d denotes the number of dimensions of patient's state, we extract its representation via a segment embedding layer, which can be further used to obtain a skill embedding for imitation learning. The segment embedding layer consists of two components, a step-wise multilayer perceptron (MLP) shared across all steps, and a 1D convolution layer encoding the segment information.

First, the segment input is fed into an MLP layer which encodes the feature at each step in an embedding space. By sharing the same MLP encoder, the state embedding at step t, $\mathbf{f}_t \in \mathbb{R}^{d'}$ can be generated as $\mathbf{f}_t = \mathbf{MLP}(s_t)$. Second, the segment of encoded state in the embedding space, $[\mathbf{f}_{t-m+1}, \cdots, \mathbf{f}_{t-2}, \mathbf{f}_{t-1}, \mathbf{f}_t] \in \mathbb{R}^{m \times d'}$ are fed into a 1D convolution layer to generate the segment embedding $\mathbf{z}_t \in \mathbb{R}^{h \times 1}$:

$$\mathbf{z}_t = \mathbf{CONCAT}_{i=0}^{h-1}(\mathbf{W}_i \star \mathbf{f}_{t-m+1:t} + b_i)$$

where $\mathbf{W}_i \in \mathbb{R}^{m \times d'}$ denotes the i-th convolution kernel (h kernels in total), $b_i \in \mathbb{R}$ denotes the corresponding bias term, $\star$ operator provides the sum of row-wise cross-correlation, $\mathbf{CONCAT}$ provides the concatenation of all convolution results on h kernels.

In general, there are multiple choices for segment embedding layer, such as LSTM/GRU and Transformer [11]–[13].

Long segments are relatively rare in DTR, thus we use convolution neural networks due to its efficiency and effectiveness in extracting salient embedding of short segments.

2) Prototype Layer: As aforementioned, the learning of prototypes gains advantages of interpretability compared to other interpretable imitation learning models as the original data segments on which the prototype vectors are projected, are always available for analysis.

In the prototype layer, there are k prototype vectors $\mathbf{p} = [\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_k] \in \mathbb{R}^{k \times h}$, which are essentially trainable model parameters and each has the same number of dimension as the segment embedding $\mathbf{z_t}$. Each prototype vector represents a class of exemplar segments reflecting the patient's state at one stage. The similarity score between the segment embedding $\mathbf{z_t}$ and each prototype vectors is then computed: $\text{Sim}(\mathbf{z}_t, \mathbf{p}_i) = \exp(-\left\|\mathbf{z}_t - \mathbf{p}_i\right\|_2^2)$, where $\mathbf{p}_i$ denotes the i-th prototype vector, $\left\|\cdot\right\|_2$ denotes the L_2-norm, and the exponential function brings the similarity score to a bounded range for numerical stability.

After that, the similarity scores of all prototype-embedding pairs are scaled between $[0, 1]$ and the resulting scaled score for i-th prototype-embedding pair $\hat{\text{Sim}}(\mathbf{z}_t, \mathbf{p}_i)$ is:

$$\hat{\text{Sim}}(\mathbf{z}_t, \mathbf{p}_i) = \frac{\exp(-\left\|\mathbf{z}_t - \mathbf{p}_i\right\|_2^2)}{\sum_{i=1}^{k} \exp(-\left\|\mathbf{z}_t - \mathbf{p}_i\right\|_2^2)}$$

All scaled scores form a weighting vector $\mathbf{W}_p \in \mathbb{R}^{k \times 1}$ as: $\mathbf{W}_p = [\hat{\text{Sim}}(\mathbf{z}_t, \mathbf{p}_1), \hat{\text{Sim}}(\mathbf{z}_t, \mathbf{p}_2), \cdots, \hat{\text{Sim}}(\mathbf{z}_t, \mathbf{p}_k)]$. Based on $\mathbf{W}_p$, the skill embedding $\mathbf{o}_t \in \mathbb{R}^{1 \times h}$ for the given segment can be constructed by the weighted combination of vectors $\mathbf{p}$:

$$\mathbf{o}_t = \mathbf{W}_p^T \cdot \mathbf{p}$$

where $\cdot$ denotes the inner product operation. Note that instead of the original segment embedding $\mathbf{z}_t$, all prototype vectors $\mathbf{p}$ are involved in the final skill embedding $\mathbf{o}_t$, so that the skill embedding can be interpreted based on the original state segments via prototype-segment association, which will be introduced in the prototype-segment association section.

3) Imitation Learning Layer: We modify a flat policy network $\pi_\theta(a_t|s_t)$ that is parameterized by θ and learns mappings from s_t to a_t, by incorporating the skill embedding $\mathbf{o}_t$ that serves as a high-level indicator guiding the agent to mimic expert demonstrations sampled from the expert policy $\pi_E(a_t|s_t)$. To be specific, the patient state s_t is concatenated with the skill embedding $\mathbf{o}_t$ as conditional information, and fed to the contextual policy $\pi_\theta(a_t|\mathbf{o}_t, s_t)$, from which we get the primitive output action a_t for the DTR task:

$$a_t \leftarrow \pi_\theta(a_t|\mathbf{o}_t, s_t)$$

We build the contextual policy network based on a traditional imitation learning model, behavior cloning (BC), which aims to imitate the doctor's medication at each time step t, by treating it as a supervised learning problem. The actual policy network $\pi_\theta(a_t|\mathbf{o}_t, s_t)$ is implemented by a 3-layer MLP.

Different from the other interpretable imitation learning framework that manipulates the latent codes that is formulated and optimized via information measures [5], our interpretable skill learning method explicitly models the segments of patient's states with prototype vectors that is learned and regularized by the behavior cloning and multiple interpretable learning objectives, based on which the user can always obtain the explanation by tracing back to training segments.

C. Learning Objectives

The learning objective contains the imitation learning objective and multiple regularization components that are designed to enhance the interpretability of learned prototypes on these disjoint segments. In this subsection, we introduce these learning objective terms individually.

1) Imitation Learning: Given a batch of segments with size n, $\{(s_1^{(j)}, a_1^{(j)}), (s_2^{(j)}, a_2^{(j)}), \cdots, (s_m^{(j)}, a_m^{(j)})\}_{j=1}^n$, the contextual policy aims to mimic the doctor's demonstration at the segment level in a supervised manner:

$$\mathcal{L}_{\text{IM}} = \sum_{j=1}^{n} \sum_{t=1}^{m} \pi_E \left(a_t^{(j)} \mid s_t^{(j)}\right) \log \pi_\theta \left(a_t^{(j)} \mid \mathbf{o}_t^{(j)}, s_t^{(j)}\right)$$

where m is the length of a segment, π_E denotes the expert policy where the demonstrations are sampled.

2) Prototype Learning: To improve the interpretability of our skill learning model, similar to what previous work on sequence learning has done [9], [10], we leverage three key components regularizing the learning of prototype vectors regarding the clustering structure of segment embedding, the segment-prototype evidence, and the diversity of prototypes.

The clustering structure regularization enforces the segment embedding $\mathbf{z}_t$ to be as adjacent to its closest prototype as possible via the minimization of the L_2 distance:

$$\mathcal{L}_{\text{Cluster}} = \sum_{j=1}^{n} \min_{i \in [k]} \left\|\mathbf{z}_t^{(j)} - \mathbf{p}_i\right\|_2^2$$

where $[k]$ denotes the integer set with the max element k representing all prototype vectors.

The prototype-segment evidence regularization imposes a dual optimization objective regarding segment embedding and prototype vectors. It encourages each prototype vector to be as similar to a segment embedding as possible by minimizing the L_2 distance between a prototype vector and its closet embedding among the batch of segments:

$$\mathcal{L}_{\text{Evidence}} = \sum_{i=1}^{k} \min_{j \in [n]} \left\|\mathbf{p}_i - \mathbf{z}_t^{(j)}\right\|_2^2$$

The clustering structure and prototype-segment evidence interact with each other and jointly constrain the learning of both the segment embedding layer and prototype layer towards a clear and interpretable structure.

Besides the above regularization terms, it is natural to penalize the similarity between each pair of prototype vectors, as indistinguishable prototype vectors representing very similar

patient states may be redundant. Besides, encouraging the diversity of prototypes would give rise to a better generalization when encountering new segments and trajectories. Therefore, the diversity regularization term is imposed as follows:

$$\mathcal{L}_{\text{diversity}} = \sum_{i=1}^{k} \sum_{i' \neq i}^{k} \max\left(0, d_{\min} - \|\mathbf{p}_i - \mathbf{p}_{i'}\|_2^2\right)$$

where $d_{\min}$ is a proximity threshold determining whether to penalize each prototype pair or not.

3) Full objective function: Given the imitation learning objective and multiple regularization terms, the full objective function can be obtained as:

$$\mathcal{L}_{\text{all}} = \mathcal{L}_{\text{IM}} + \lambda_1 \mathcal{L}_{\text{Cluster}} + \lambda_2 \mathcal{L}_{\text{Evidence}} + \lambda_3 \mathcal{L}_{\text{diversity}}$$

where the corresponding weight λ_1, λ_2, and λ_3 that range from 0 to 1, balance the aforementioned regularization components.

Note that we impose these constraints only at the last step $t = m$, as it contains the entire disjoint segment that we intend to optimize by our hypothesis.

D. Prototype-Segment Association

After the training loss converges, the prototype vectors are optimized to be adequately close to certain segment embedding from the training data. However, these prototype vectors are still not interpretable at this stage as there is no correspondence between them and actual segments. To induce the association between prototypes and segments in the training data, we assign each prototype $\mathbf{p}_i$ to its closet segment in the latent space:

$$\mathbf{p}_i \leftarrow \underset{\mathbf{z}_t^{(j)} \in \mathcal{Z}_{train}}{\arg\min} \left\| \mathbf{p}_i - \mathbf{z}_t^{(j)} \right\|_2^2$$

where $\mathcal{Z}_{train}$ denotes the set of segment embedding generated by feeding all disjoint segments($t = m$) in the training set to the segment embedding layer.

III. Experiments

In this section, we present the experiments that evaluate our proposed ISL model. We first introduce the dataset, evaluation metrics, baseline models, and specific training configurations. Next, we compare the performance of these models in terms of imitation learning. Finally, we demonstrate the effectiveness in terms of interpretation by conducting model analysis.

A. Dataset and Evaluation Metrics

We evaluate our model and baselines based on a public electronic health record dataset MIMIC-III [14], which contains the records of 43,000 unique patients in intensive care units between 2001 and 2012. Following the Sepsis-3 criteria [15], we extracted the Sepsis data among 6,695 distinct diseases and 4,127 drugs. The Sepsis dataset contains the hourly-based trajectories of 11419 patients. Each patient state consist of 43 features including demographics, lab test values, vital signs, the historical treatment and other admission information. Besides, we focus on the medical treatment using intravenous (IV) fluid that is vital in the treatment of septic patients. To construct the segment-level demonstrations, we split each trajectory by the segment length and drop off leftover steps. We set the segment length as 4 because the segment pattern identified in the corresponding ground truth actions is the most obvious. Other pre-processing details follow [4]. We randomly split the final dataset based on the patient IDs for training/validation/testing datasets by a ratio of 60%/20%/20%.

The treatment policies are evaluated on three metrics: the averaged Jaccard coefficient as well as the micro and the macro average of AUC-ROC scores, denoted as Jaccard, MI-AUC, and MA-AUC, respectively. Note that the evaluation is performed on positive trajectories only as the goal is to mimic the demonstrations that lead to the survival of patients.

B. Baselines and Experiment Setup

Our proposed ISL model is compared to baseline and state-of-art imitation learning models for DTR tasks, which are introduced as follows:

- **Behavior Cloning(BC)**: In BC, the trajectories are split into multiple state-action pairings, and the treatment policy is learned from step-by-step expert demonstrations in a supervised manner.
- **GAIL** [3]: GAIL learns a policy in an adversarial fashion, with the policy network generating trajectories and receiving the reward signal provided by the discriminator based on the expert trajectories.
- **ACIL** [4]: By introducing a second cooperative discriminator and a training objective, ACIL utilizes data from both positive and negative trajectories, based on which the distribution of expert and learned demonstrations are optimized to be away from the negative one.

For fair comparisons, the policy networks of all models are set to the same architecture based on a 3-layer MLP with the same neuron size and activation function. Besides, the architecture of the discriminator in GAIL is the same as the two discriminators in ACIL, which is also a 3-layer MLP. We set the batch size as 64 and use Adam [16] as the optimizer. Next, we present the hyperparameters used in ISL: The dimension of segment embedding is 48; the number of prototypes in ISL is $k = 25$; the regularization weights of ISL are set as $\lambda_1 = 0.2$, $\lambda_2 = 0.1$, and $\lambda_3 = 0.1$, with $d_{\min} = 1.0$.

C. Performance Evaluation

Table I summarizes the Jaccard, MI-AUC, and MA-AUC of the baselines and our proposed model. We have several observations and discussions as follows. First, GAIL outperforms BC in terms of Jaccard and also has a minor advantage regarding MI-AUC and MA-AUC. This is because a vanilla BC ignores the sequential information of trajectories and learns from the expert demonstrations step-wisely, which suffers from compounding errors. GAIL models and learns the distribution of expert trajectories in an adversarial manner, where the learned policy is able to receive an effective reward generated by the discriminator. Second, ACIL has

TABLE I
PERFORMANCE COMPARISON ON SEPSIS DATASET

	Jaccard	MI-AUC	MA-AUC
BC	0.5228	0.7848	0.7617
GAIL	0.5286	0.7855	0.7621
ACIL	0.5302	0.7874	0.7645
ISL	**0.5487**	**0.8048**	**0.7837**

TABLE II
ABLATION STUDY

	Jaccard	MI-AUC	MA-AUC
ISL	**0.5487**	**0.8048**	**0.7837**
(a) w/o Prototype	0.5497	0.8092	0.7875
(b) $\lambda_1 = 0$	0.5341	0.7969	0.7774
(c) $\lambda_2 = 0$	0.4870	0.7643	0.7605
(d) $\lambda_3 = 0$	0.5409	0.7992	0.7767

better performance than GAIL as it also leverages the negative demonstrations that guide the learned policy to preserve the positive demonstrations better and to avoid making mistakes. Lastly, ISL outperforms the baselines regarding all evaluation metrics by a clear margin, which demonstrates the effectiveness and advantages of our proposed ideas. This is because ISL explores the temporal structure of the expert trajectories at the segment level, and learns the prototypes to compose the skill embedding in an interpretable and meaningful manner.

D. Model Analysis

To analyze the learning components in our proposed model, we perform an ablation study and visualize the learned prototype on the embedding space. In addition, we provide a case study by analyzing several trajectories with discussions.

1) Ablation Study: We first evaluate the performance of all variants of the ISL model, as shown in Table II. In (a), we discard the prototype layer and directly use the segment embedding as a skill embedding. The performance of this variant is slightly better than our ISL model as the segment information is utilized by the imitation learning layer in the most straightforward manner, where no constraint is imposed. However, this variant is not interpretable as there is no notion of the exemplar segment. Conversely, our ISL model makes use of real data segments as prototypes to construct the skill embedding in an interpretable way, by minorly compromising the performance.

We also evaluate the effect of regularization terms on generating high-quality prototypes by setting each weight to zero in the objective function. It can be observed that removing the clustering and diversity constraints (b,d) on prototypes has a minorly negative effect on the model performance. However, removing the evidence constraint (c) leads to a significant degradation of performance, as the skill embedding relies on an accurate prototype projection so as to provide useful conditional information for imitation learning. We conclude that the regularization terms help the model to learn good prototypes that match the most representative data segment, thus benefiting the result interpretations.

2) Effects of the number of prototypes: Next, we evaluate the performance of the ISL model given different numbers of prototypes, as shown in the middle subplot of Figure 2. It can be observed that the model performance increases quickly and stabilizes after the number of prototypes k exceeds 25. Note that a very small number of prototypes cannot capture all the representative segment information for the downstream task, thus degrading the performance. We select $k = 25$ as it gives the nearly best results with a neat prototype set, which is beneficial for interpretation.

3) Learned Prototypes: In order to assess the effectiveness of ISL regarding interoperability, we analyze the embedding space produced by the segment embedding layer. To be specific, we visualize t-SNE [17] and PCA plots of the prototype vectors and the embedding of segments in the training dataset, as shown in the left subplot of Figure 2. The embedding of each segment is presented as a dot, and each projected prototype vector is presented as a star, where the color represents the mean value of ground truth actions in the segment. Note that we are able to quantify the level of segment treatment as the ground truth action represents the dose of IV fluids, where 0 indicates no usage of IV fluid and 4 indicates that of a full dose. It is clear that the segments form several clusters in the embedding space by different levels of IV fluid usage(treatment). Besides, most prototypes are consistently matched to the nearby segments that use the same dose of IV fluid. Even if some segments with different levels of treatments (indicated by different colors) are mixed up within each cluster and a few prototypes are mismatched, it generally transits to a similar level. Moreover, prototypes are separated by a clear margin in the embedding space, implying a diverse structure for results interpretation. We also visualize the embedding space produced by an ISL variant without the prototype layer. As we discussed before, it cannot interpret the decisions of the learned policy by nature. Although the segment transits from the lowest level of treatment to the highest one, they are not well-structured and distributed loosely. The observations hold for both t-SNE and PCA plots, which indicates the effectiveness and advantages of the prototype layer and corresponding regularization designs.

4) Case Study: We further analyze the effectiveness of our model based on two trajectories with high Jaccard scores in the testing data. Note that one with domain knowledge can perform the reasoning by directly analyzing the prototypes(associated with actual patient states and treatments) and similarity weights that are outputted from the ISL model. Here we provide a simplified reasoning process for demonstration. For visualization purpose, 3 out of 43 features are extracted from the patient states, Total Protein, PaCO2, and PaO2, which are clinically important criteria for judging patient status. We visualize these lab values and highlight each segment with a color indicating the level of treatment suggested by the skill embedding. As the skill embedding is a weighted combination of all prototypes, we select 5 prototypes with the highest

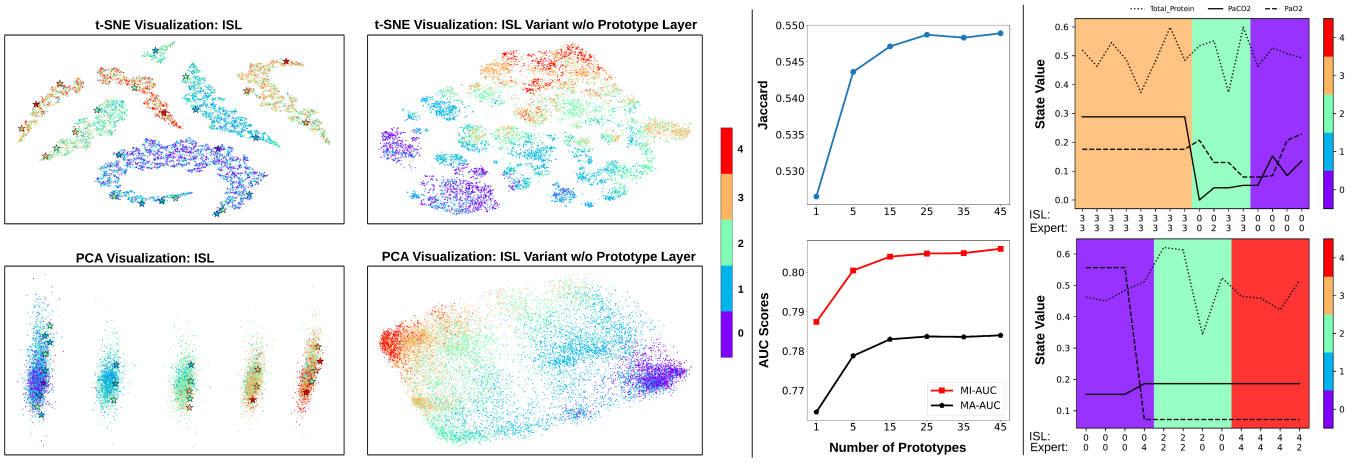

Fig. 2. Left: Learned prototypes and segment visualization on the training dataset. Middle: The effects of the number of prototypes. Right: Visualized trajectories on the testing dataset.

weights for analysis, which on average allocate 64.51% of weights among 25 prototypes in the testing dataset. Therefore, the color that represents the level of treatment suggested by the skill embedding is assigned based on the mean value of the ground truths of these prototypes. Besides, we provide the treatments given by ISL and the ground truth action at each step as a reference. The visualization results are shown in the right subplot of Figure 2. We observe clear segment patterns in each trajectory: (1) The learned skill suggests a different level of treatment as the range of one or more lab values transits. (2) The treatments in a segment given by the policy network are consistent with that implied by the skill embedding. We conclude that in these cases, the ISL model is able to capture the most representative segments and infer a faithful skill embedding in order to provide accurate treatments.

IV. Conclusion

In this paper, we present an interpretable skill learning model (ISL) to tackle a complex DTR task. Unlike existing methods, ISL learns the prototypes that capture the most representative segment-level demonstrations and composes trustful skill embedding for the decision-making and reasoning of treatments. Empirical results on a real-world DTR dataset demonstrate the advantage of ISL in providing more accurate treatments compared to state-of-the-art methods. The proposed ISL model also presents good interpretability according to the visualized patterns and the analysis of trajectories.

References

[1] B. Chakraborty and S. A. Murphy, "Dynamic treatment regimes," *Annual review of statistics and its application*, vol. 1, p. 447, 2014.

[2] F. J. Diaz, M. R. Cogollo, E. Spina, V. Santoro, D. M. Rendon, and J. de Leon, "Drug dosage individualization based on a random-effects linear model," *Journal of biopharmaceutical statistics*, vol. 22, no. 3, pp. 463–484, 2012.

[3] J. Ho and S. Ermon, "Generative adversarial imitation learning," *Advances in neural information processing systems*, vol. 29, 2016.

[4] L. Wang, W. Yu, X. He, W. Cheng, M. R. Ren, W. Wang, B. Zong, H. Chen, and H. Zha, "Adversarial cooperative imitation learning for dynamic treatment regimes," in *Proceedings of The Web Conference 2020*, 2020, pp. 1785–1795.

[5] Y. Li, J. Song, and S. Ermon, "Infogail: Interpretable imitation learning from visual demonstrations," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[6] A. Sharma, M. Sharma, N. Rhinehart, and K. M. Kitani, "Directed-info gail: Learning hierarchical policies from unsegmented demonstrations using directed information," *arXiv preprint arXiv:1810.01266*, 2018.

[7] L. Wang, R. Tang, X. He, and X. He, "Hierarchical imitation learning via subgoal representation learning for dynamic treatment recommendation," in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 2022, pp. 1081–1089.

[8] A. Raghu, M. Komorowski, L. A. Celi, P. Szolovits, and M. Ghassemi, "Continuous state-space models for optimal sepsis treatment: a deep reinforcement learning approach," in *Machine Learning for Healthcare Conference*. PMLR, 2017, pp. 147–163.

[9] Y. Ming, P. Xu, H. Qu, and L. Ren, "Interpretable and steerable sequence learning via prototypes," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 903–913.

[10] J. Ni, Z. Chen, W. Cheng, B. Zong, D. Song, Y. Liu, X. Zhang, and H. Chen, "Interpreting convolutional sequence model by learning local prototypes with adaptation regularization," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 1366–1375.

[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[12] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[14] A. E. W. Johnson, T. J. Pollard, L. Shen, L.-W. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, "Mimic-iii, a freely accessible critical care database," *Scientific data*, vol. 3, p. 160035, May 2016. [Online]. Available: https://europepmc.org/articles/PMC4878278

[15] M. Singer, C. S. Deutschman, C. W. Seymour, M. Shankar-Hari, D. Annane, M. Bauer, R. Bellomo, G. R. Bernard, J.-D. Chiche, C. M. Coopersmith, R. S. Hotchkiss, M. M. Levy, J. C. Marshall, G. S. Martin, S. M. Opal, G. D. Rubenfeld, T. van der Poll, J.-L. Vincent, and D. C. Angus, "The Third International Consensus Definitions for Sepsis and Septic Shock (Sepsis-3)," *JAMA*, vol. 315, no. 8, pp. 801–810, 02 2016. [Online]. Available: https://doi.org/10.1001/jama.2016.0287

[16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR (Poster)*, 2015.

[17] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.