RESEARCH-ARTICLE

FedSkill: Privacy Preserved Interpretable Skill Learning via Imitation

YUSHAN JIANG, University of Connecticut, Storrs, CT, United States

WENCHAO YU, NEC Laboratories America, Inc., Princeton, NJ, United States

DONGJIN SONG, University of Connecticut, Storrs, CT, United States

LU WANG, East China Normal University, Shanghai, China

WEI CHENG, NEC Laboratories America, Inc., Princeton, NJ, United States

HAIFENG CHEN, NEC Laboratories America, Inc., Princeton, NJ, United States

Open Access Support provided by:

NEC Laboratories America, Inc.

East China Normal University

University of Connecticut

Published: 06 August 2023

Citation in BibTeX format

KDD '23: The 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining
August 6 - 10, 2023
CA, Long Beach, USA

Conference Sponsors:
SIGMOD
SIGKDD

KDD '23: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (August 2023)
https://doi.org/10.1145/3580305.3599349
ISBN: 9798400701030

FedSkill: Privacy Preserved Interpretable Skill Learning via Imitation

Yushan Jiang
University of Connecticut
Storrs, CT, USA
yushan.jiang@uconn.edu

Wenchao Yu*
NEC Labs America
Princeton, NJ, USA
wyu@nec-labs.com

Dongjin Song*
University of Connecticut
Storrs, CT, USA
dongjin.song@uconn.edu

Lu Wang
East China Normal University
Shanghai, China
luwang@stu.ecnu.edu.cn

Wei Cheng
NEC Labs America
Princeton, NJ, USA
weicheng@nec-labs.com

Haifeng Chen
NEC Labs America
Princeton, NJ, USA
haifeng@nec-labs.com

ABSTRACT

Imitation learning that replicates experts' skills via their demonstrations has shown significant success in various decision-making tasks. However, two critical challenges still hinder the deployment of imitation learning techniques in real-world application scenarios. First, existing methods lack the intrinsic interpretability to explicitly explain the underlying rationale of the learned skill and thus making learned policy untrustworthy. Second, due to the scarcity of expert demonstrations from each end user (client), learning a policy based on different data silos is necessary but challenging in privacy-sensitive applications such as finance and healthcare. To this end, we present a privacy-preserved interpretable skill learning framework (FedSkill) that enables global policy learning to incorporate data from different sources and provides explainable interpretations to each local user without violating privacy and data sovereignty. Specifically, our proposed interpretable skill learning model can capture the varying patterns in the trajectories of expert demonstrations, and extract prototypical information as skills that provide implicit guidance for policy learning and explicit explanations in the reasoning process. Moreover, we design a novel aggregation mechanism coupled with the based skill learning model to preserve global information utilization and maintain local interpretability under the federated framework. Thoroughly experiments on three datasets and empirical studies demonstrate that our proposed FedSkill framework not only outperforms state-of-the-art imitation learning methods but also exhibits good interpretability under a federated setting. Our proposed FedSkill framework is the first attempt to bridge the gaps among federated learning, interpretable machine learning, and imitation learning.

CCS CONCEPTS

• **Computing methodologies → Learning paradigms**.

*Wenchao Yu and Dongjin Song are corresponding authors.

KDD '23, August 6–10, 2023, Long Beach, CA, USA

ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599349

KEYWORDS

Federated Learning, Imitation Learning, Prototype, Interpretable Machine Learning

ACM Reference Format:
Yushan Jiang, Wenchao Yu, Dongjin Song, Lu Wang, Wei Cheng, and Haifeng Chen. 2023. FedSkill: Privacy Preserved Interpretable Skill Learning via Imitation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3580305.3599349

1 INTRODUCTION

Imitation learning that mimics experts' skillful behaviors has demonstrated its success in a wide range of real-world decision-making tasks, e.g., autonomous driving, robotics, and healthcare [7, 11, 13, 37, 43]. Without accessing and optimizing the explicitly predefined reward that can be sparse and inappropriate in complex tasks, imitation learning exhibits advantages over reinforcement learning by directly exploiting the experts' demonstrations for policy learning. However, building an effective imitation learning system is challenging since it needs to exploit a sufficient amount of expert demonstrations to accurately represent the needed skill in task environments. This introduces two critical challenges that hinder the model deployment in practical application scenarios. First, expert demonstrations can be extremely sparse from an individual client or organization. Meanwhile, it is infeasible to perform centralized model training by collecting example demonstrations from multiple sources (e.g., clients or organizations) because of privacy concerns or enforced regulations. Second, even the demonstrations from a single source can be multi-mode and highly variable as human experts may have different skills. It is inherently difficult to capture different skill structures within the set of demonstrations, let alone explicitly discover the underlying rationales of the learned policy.

A typical solution to resolve the first challenge is to leverage federated learning and perform imitation learning in a privacy-preserving manner. Existing studies have shown that federated imitation learning is desired in various applications, e.g., urban traffic forecasting, autonomous driving, robotic system, and UAV swarm coordination [22, 38, 40, 41]. Specifically, by leveraging expert demonstrations from multiple sources (e.g., clients or organizations), agents can benefit from the federated framework and yield better performance on downstream tasks.

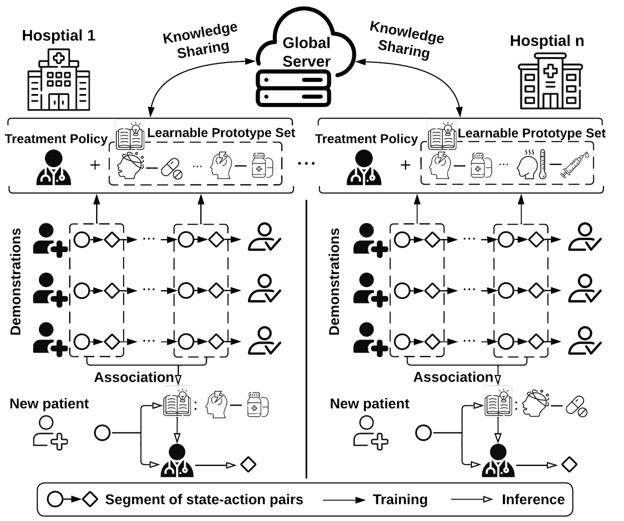

Figure 1: The overview of the proposed FedSkill framework in dynamic treatment recommendation.

To facilitate the deployment of federated imitation learning for practical applications, a critical step is to build a federated imitation learning framework with a desired interpretation architecture. Recent developments in interpretable sequence modeling that resort to case-based reasoning have brought an end-to-end interpretable solution on downstream classification results [25, 26]. These methods have achieved competitive performance compared to the state-of-the-art black box models, and more importantly, enabled faithful explanations for the model's output based on the known representative data. As the expert demonstration is often a sequence of state and action pairs, similar techniques could be developed to interpret the learned skills within the varying patterns.

In this paper, we propose a self-explainable federated skill learning framework with built-in interpretability, i.e., FedSkill, to better exploit the experts' demonstrations based on imitation learning. Specifically, we first present an interpretable skill learning model to capture the varying patterns in the trajectories of expert demonstrations, and extract prototypical information as skills provide implicit guidance for the agent to take actions, as well as explicit explanations in the reasoning process. Note that the skill is learned at the segment level since it is more flexible and transferable across different experts compared to a trajectory-level formulation, and is less complex and laboring compared to a step-level formulation. Next, we develop a novel aggregation mechanism coupled with the based skill learning model to preserve global information utilization and maintain local interpretability under the federated framework.

We provide an example in dynamic treatment recommendation (DTR) to illustrate the proposed FedSkill framework, as shown in Figure 1. Suppose that n hospitals collaborate to cure a disease via imitation learning. The goal for each hospital is to learn a model that discovers and explains the underlying skills to cope with the variability of patients' symptoms, in a privacy-preserved manner. In this model, a contextual treatment policy is coupled with a learnable prototype set containing parameterized vectors that will represent prototypical symptoms and treatments (prototypes) after training. As such, the skill is formulated by combining multiple prototypes representing different treatment plans that correspond to different symptoms. During the local training at each hospital, we extract the representations of segmented trajectories as candidates for skill learning, where the parameterized vectors are optimized to have desired interpretation properties via multiple objectives based on these candidates. On the server side, an intuitive choice is to apply federated averaging to aggregate the treatment policy networks and learnable prototype sets. However, the treatment demonstrations can be heterogeneous across hospitals, leading to different disease treatment skills. To enable better knowledge sharing under federated learning, we follow the fashion in [10] and perform clustering to identify the memberships of parameterized vectors from all hospitals. As such, similar prototypical knowledge is aligned on the server side, which enhances local skill learning. After training, each hospital owns a common treatment policy and a unique prototype set, where we can easily associate each parameterized vector with a meaningful segment in local training data, thus yielding a prototype that is readily interpretable. Note that local data privacy is preserved as the global server does not access the data or its representation during the whole process. Now, each hospital can detect a new patient's symptoms, and provide the underlying skill with interpretations by analyzing the data that it is familiar with, where the contextual treatment policy can recommend proper medications based on the patient's status and inferred skill. Our contributions are highlighted as follows:

- We propose a federated interpretable skill learning framework via imitation, i.e., FedSkill, which exploits the segment-level expert demonstrations from multiple clients and produces representative and transferable skills.
- The skill learning model learns to construct faithful skills to explain the underlying rationale of varying demonstrations for an imitation learning task, which is innovative compared to the existing imitation learning techniques.
- To the best of our knowledge, this is the first work that equips the federated imitation learning framework with a self-explainable structure for downstream tasks. The proposed FedSkill framework provides an alignment of global skillful information and preserves local interpretability in a privacy-preserving manner.
- Experiments and analysis on a synthetic dataset and two benchmark DTR datasets demonstrate that FedSkill provides better performance than state-of-the-art imitation models under federated learning setting, together with reasonable explanations towards the recommended treatments.

The remainder of this paper is organized as follows. Section II reviews the existing literature related to our federated interpretable skill learning framework. Section III presents the methodology of our framework. The datasets, experimental setup, evaluation and analysis are described in Section IV. Section V concludes our paper.

2 RELATED WORK

2.1 Imitation Learning

Existing imitation learning methods can be generally partitioned into three categories, behavior cloning (BC) [1, 9], inverse reinforcement learning (IRL) [13, 23, 27, 46], and adversarial imitation learning (AIL) [11, 31, 37, 42, 47]. Recent studies have heightened the importance of AIL in alleviating the issues in high dimensional state-action space [44]. It exploits the expert's demonstration by learning its distribution to give the implicit reward signal based on an adversarial structure, which is scalable and typically boosts the performance of learned policy in large and complex environments. Nevertheless, the experts' behaviors in real-world tasks often illustrate strong variability among multiple modes, and most existing methods that provide a flat policy are insufficient. These methods don't account for the variability of expert trajectories and fail to present an interpretable structure of multi-mode demonstrations.

To resolve this issue, some methods construct a contextual policy and learn the context information to represent the variability of expert demonstrations at the step [29], segment [36] and trajectory level [20], respectively. One branch of existing methods models the variation of the underlying demonstration trajectory as a latent skill variable in graphical models where the association between state and meaningful representation is enforced by information-theoretic regularization [20, 29]. The other branch models the hierarchy of imitation learning with the formulation of high-level representations as skills and low-level policies, so that the variability is automatically addressed and explained by the interactions between high-level guidance and low-level behaviors [36]. However, these methods provide ad-hoc or limited explanations by manipulating the latent contextual information, which is insufficient for reasoning skills and actions existing in the varying patterns. Our proposed method also uses a contextual policy for imitation learning, but the learned context is explicitly explainable as one can always trace back to the associated original training segments for explaination.

2.2 Prototype Learning for Sequence Data

As aforementioned, our work naturally relates to interpretable modeling methods for sequence data. Recent studies in this field have switched the focus from post-hoc interpretation methods [16] that learn to fit explanations from inferred results, to self-explainable methods that provide end-to-end built-in interpretations for their own outputs. Besides the well-known attention-based interpretation mechanism [6], the prototype learning methods also achieve satisfactory performance and gives explicit interpretations for sequence data classification [25, 26]. These methods learn a set of parameters to represent exemplar sequences in the representation space via specific optimization objectives designed for interpretation. After training converges, these parameters are associated with the original data as prototypes. As such, the prediction can be inferred and explained by the similarity between the associated prototypes and input data in the representation space, which is closely related to the case-based reasoning process. Instead of providing interpretations for the low-dimension sequences (e.g., univariate time series, protein sequences), our method focuses on capturing and interpreting the skills existing in varying patterns of shorter segments but with more complex behaviors.

2.3 Federated Learning

In recent years, federated learning has attracted significant attention in the AI research community as it enables collaborative learning from multiple data sources in a privacy-preserved fashion. Federated Averaging (FedAvg) [24] is the most commonly used aggregation mechanism among the existing methods because of its simplicity and effectiveness in a wide range of federated environments. In order to tackle the data heterogeneity issues, another widely-used method, Federated Proximal(FedProx) [19] improves the local updates in FedAvg by regularizing the L_2 distance between local and global models. Moreover, performing global clustering provides another solution to tackle the heterogeneous data distribution [4, 10, 28, 32, 45], facilitating various downstream tasks. Enlightened by their success, we also adopt federated clustering for a better alignment of skillful information toward the inference and reasoning of imitation learning tasks.

Note that the most recent works have started exploring the idea of federated prototype learning [8, 33]. There are two main differences between our method and their works. Firstly, the prototype in these studies is defined as certain forms (e.g., mean) of the data representation generated by the feature encoding networks, while the so-called prototype in our method is just model parameters during training. Therefore, our method has much fewer privacy concerns regarding data leakage than the existing methods. Secondly, these works' research focus is to address the heterogeneous data distribution and improve the performance of the global model. While our method can demonstrate robustness in heterogeneous data environments, our motivation for using prototypes is to provide interpretation without violating privacy requirements.

3 METHODOLOGY

In this section, we present the FedSkill framework that performs skill learning by imitating expert demonstrations from multiple clients. We first state the problem we aim to study. Then, we give the detailed architecture design of the interpretable skill learning model over a single client. Finally, we develop the federated interpretable skill learning and introduce the training procedures over different local clients and the global server, which renders local interpretability while preserving data privacy.

3.1 Problem Formulation

3.1.1 Learning Intrepretable Skills under Federation. We brief the federated interpretable skill learning framework (as shown in Figure 3) as follows: Suppose there is a set of clients C with $|C|$ clients, each client aims to learn an interpretable skill learning model including a client-specific prototype layer (referring to the learnable prototype set in Figure 1) connecting the other learning components that generate segment representation and perform imitation learning task, respectively. A set of parameterized vectors in the prototype layer learn from segment representations so as to establish the prototypes and construct skill embedding that provides contextual information for imitation learning.

On the server side, all client models are aggregated based on our designed mechanism so that similar prototypical information is aligned. After global and local training, parameterized vectors of each client will be translated to interpretable prototypes via the association in the representation space of local training data. As such, each client can explain the skills based on the most similar prototypes in the reasoning process.

3.1.2 Segment-level Trajectory Exploitation. As aforementioned, we exploit segment-level expert demonstrations to perform imitation learning tasks based on the intuition that the obtained skills are

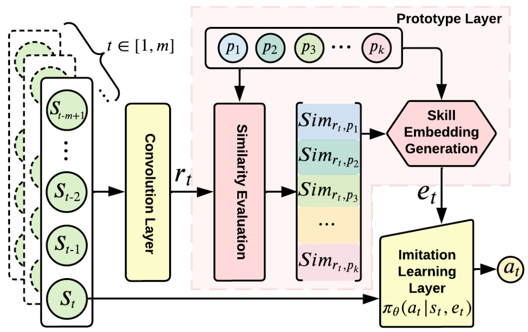

Figure 2: The illustration of the proposed interpretable skill learning model via imitation.

representative and more transferable. For this purpose, we divide each trajectory into multiple non-overlapped segments of length m, resulting in a set of segments $\{[(s_t^{(i)}, a_t^{(i)})]_{t=1}^m\}_{i=1}^N$, where N is the number of segments from all trajectories. In the segment, each state comes with the previous $m-1$ steps of states from the same trajectory, which encodes temporal dynamics up to the current step. As such, the skill transitions across consecutive segments are also captured in the model learning. Padding of the initial state in the input is performed in the exploitation process when necessary.

3.2 Interpretable Skill Learning Model

The proposed interpretable skill learning model consists of a convolution layer to encode the temporal dynamics from the input segment, a prototype layer to generate skill embedding, and a contextual imitation learning layer to perform primitive actions, as shown in Figure 2.

3.2.1 Convolution Layer for Segment Representation. Given the input segment to the model at the step t, $[s_{t-m+1}, \cdots, s_{t-2}, s_{t-1}, s_t] \in \mathbb{R}^{m\times d}$, where d denotes is the feature dimension of each state, we extract its representation $\mathbf{r}_t$ via a 2-D convolution layer $\mathbf{Conv}\,(\cdot)$, which simultaneously encodes the feature and temporal dynamics:

$$\mathbf{r}_t = \mathbf{Conv}\,(\mathbf{s}_{t-m+1:t}) = \mathbf{tanh}(\mathbf{CAT}_{i=0}^{h-1}(\mathbf{W}_i \star \mathbf{s}_{t-m+1:t} + b_i))$$

where $\mathbf{r}_t \in \mathbb{R}^{h\times 1}$, h is the number of kernels; $\mathbf{W}_i \in \mathbb{R}^{m\times d}$ and $b_i \in \mathbb{R}$ are the weight and the bias term for i-th 2D convolution kernel, respectively; $\star$ is a 2D cross-correlation operator, $\mathbf{CAT}$ denotes the concatenation operator.

We use a convolution-based encoder to generate segment representation, rather than recurrent-based [5, 12] and transformer-based [35] encoders as it is empirically more effective and efficient in extracting salient information for segments that typically have shorter lengths. Note that graph structure learning methods[18, 39] can also explicitly model the structured feature interactions at the segment level, which will be explored in our future work.

3.2.2 Prototype Layer for Skill Embedding Generation. As aforementioned, there is a set of k parameterized vectors in the prototype layer, $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_k] \in \mathbb{R}^{k\times h}$, whose dimensionality is identical to $\mathbf{z_t}$. Each vector is optimized to be representative and close to a set of similar segments in the representation space, where the optimization objectives will be introduced later.

During the forward pass, the similarity between the segment representation $\mathbf{r}_t$ and each parameterized vector is first evaluated by an exponential function based on their L_2 distance: $\exp(-\|\mathbf{r}_t - \mathbf{p}_\cdot\|_2^2)$, which bounds the similarity value to a unit range. To measure the relative similarity, all evaluated scores are re-scaled by their sum, and the final pairwise similarity score for the segment representation and i-th vector, $sim(\mathbf{r}_t, \mathbf{p}_i) \in \mathbb{R}$, is given as:

$$sim_{\mathbf{r}_t,\mathbf{p}_i} = \frac{\exp(-\|\mathbf{r}_t - \mathbf{p}_i\|_2^2)}{\sum_{i=1}^k \exp(-\|\mathbf{r}_t - \mathbf{p}_i\|_2^2)}.$$

All relative similarity scores are concatenated to a similarity vector $\mathbf{W}_{sim} \in \mathbb{R}^{k\times 1}$ as $\mathbf{W}_{sim} = [sim_{\mathbf{r}_t,\mathbf{p}_1}, sim_{\mathbf{r}_t,\mathbf{p}_2}, \cdots, sim_{\mathbf{r}_t,\mathbf{p}_k}]$, which imply the importance of the corresponding vectors in the skill embedding generation. Thus, the skill embedding $e_t \in \mathbb{R}^{1\times h}$ is generated by the weighted combination of all parameterized vectors:

$$\mathbf{e}_t = \mathbf{W}_{sim}^T \cdot \mathbf{P}$$

Note that the segment representation $\mathbf{r}_t$ is not directly used in the final form of a skill embedding. Instead, we use the parameterized vectors to reconstruct the skill information preserved by the segment representation, which renders a flexible interpretation structure. In the reasoning process, all parameterized vectors $\mathbf{p}$ are projected to prototypes that are the segment representations of local training data, thus the underlying skill of a segment is explained by similar prototypes with high weights. It is also worth noting that the above process follows a fashion of soft-skill combination, which can be altered to a hard-skill selection by adopting a Gumbel-Softmax mechanism[14]. We find that the soft-skill combination provides better flexibility and generalizability when encountering a new pattern in a complex task environment.

3.2.3 Imitation Learning Layer. In the imitation learning layer, we build a contextual policy based on the behavior cloning (BC) method that learns a mapping from state s_t to action a_t in a supervised manner. The contextual policy is parameterized by θ and denoted as $\pi_\theta(a_t|\mathbf{e}_t, s_t)$, which takes the concatenation of skill embedding and the state as the input. As aforementioned, the skill embedding captures the varying patterns of expert trajectories and guides the agent to perform primitive action more accurately:

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

Note that our model is similar to existing imitation learning methods that contextualize the policy with a latent code or a formulated sub-goal to address the variability of expert trajectories. However, only our method is capable of providing an explicit interpretation of the varying patterns in the reasoning process.

3.2.4 Learning Objectives for Better Interpretability. The learning objectives consist of two aspects, a segment-level imitation learning objective and three objectives that reinforce the interpretability of the final prototypes and skills in these non-overlapped segments. Next, we introduce each component based on a batch of segments in the training data: $\{[(s_t^{(i)}, a_t^{(i)})]_{t=1}^m\}_{i=1}^n$, with batchsize n and segment length m.

In this paper, we focus on the setting of replicating the experts' discrete actions, thus the first objective is to minimize the below

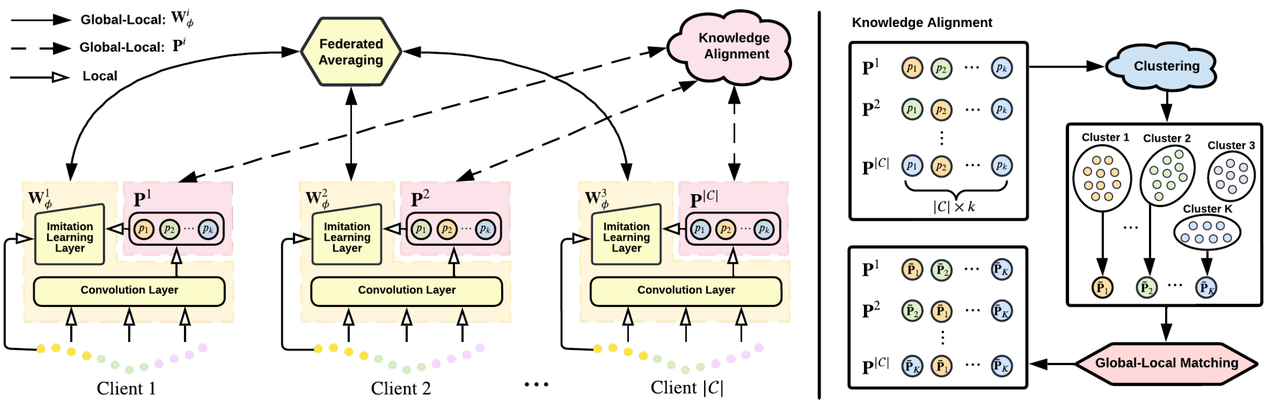

Figure 3: An illustration of the proposed FedSkill. The server performs federated averaging on the parameters in convolution and imitation layers. The client-specific parameterized vectors are aggregated by clustering and matching to the centroids of the corresponding clusters.

loss of imitation learning with BC:

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

where π_E denotes the expert policy that generates demonstrations.

The second objective regularizes the non-overlapped segment representation to be adjacent to its closest prototype, which enforces a clustering structure of segments in the representation space. This is achieved by minimizing the smallest L_2 distance between each non-overlapped $\mathbf{r}_{t=m}$ and all parameterized vectors in $\mathbf{P}$:

$$\mathcal{L}_c = \sum_{i=1}^{n} \min_{\mathbf{p}_j \in \mathbf{P}} \left\|\mathbf{r}_{t=m}^{(i)} - \mathbf{p}_j\right\|_2^2.$$

The third objective reversely regularizes each vector to be similar to a segment representation by minimizing the smallest L_2 distance between each $\mathbf{p}_i$ and a batch of non-overlapped segment representations $\mathbf{r}_{t=m} \in \mathcal{R}_n \overset{\text{def}}{=} [[\mathbf{Conv}\,(\mathbf{s}^{(j)}_{t-m+1:t})]_{t=m}]_{j=1}^{n}$, which helps the downstream projection to evidence segments.

$$\mathcal{L}_e = \sum_{i=1}^{k} \min_{\mathbf{r}_{t=m} \in \mathcal{R}_n} \left\|\mathbf{p}_i - \mathbf{r}_{t=m}^{(j)}\right\|_2^2.$$

The second and third objectives impose dual regularizations on the learning of the convolution layer and the prototype layer towards a clearer representation structure for interpretation.

The last objective enforces a diverse structure of parameterized vectors to avoid redundancy and improve the generalizability of resulting prototypes, where the L_2 distance between each pair of vectors is penalized, with a threshold $d_{\min}$:

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

As such, the full objective to be minimized can be written as: $\mathcal{L} = \mathcal{L}_{im} + \lambda_1\mathcal{L}_c + \lambda_2\mathcal{L}_e + \lambda_3\mathcal{L}_d$, with non-negative weights λ_1, λ_2, and λ_3 being used to balance components towards an optimal solution.

3.3 Federated Interpretable Skill Learning

In this subsection, we introduce the details of the FedSkill framework as shown in Figure 3. On the server side, the global knowledge alignment is conducted during the training stage; on the client side, a local prototype projection is applied at the model deployment stage, which yields interpretability without privacy concerns.

Algorithm 1: Training Algorithm of FedSkill

Input : local datasets D^c, client set C, number of clusters K, number of global epochs T, number of local epochs E, learning rate η

Server executes:

Initialize the global interpretable skill learning model $\mathbf{W}^g$ that consists of convolution and imitation learning layers $\mathbf{W}^g_\phi$, and K parameterized vectors $\mathbf{P}^g$, $\mathbf{W}^g := \mathbf{W}^g_\phi \cup \mathbf{P}^g$

for round $t = 0$ to T **do**
  Distribute $\mathbf{W}^g_\phi$ and client-specific $\mathbf{P}^C$
  for $c \in C$ **do in parallel**
    $\mathbf{W}^c_{\phi,t+1}, \mathbf{P}^c_{t+1} \leftarrow \text{LocalUpdate}(c, \mathbf{W}^c_t)$
  end
  $\mathbf{W}^g_{\phi,t+1} \leftarrow \frac{1}{|C|}\sum_{c=1}^{|C|} \mathbf{W}^c_{\phi,t+1}$
  $\hat{\mathbf{P}}^1, \hat{\mathbf{P}}^2, \cdots, \hat{\mathbf{P}}^K \leftarrow \text{Clustering}(\bigcup_{c\in C} \mathbf{P}^c_{t+1})$
  $\bar{\mathbf{P}}^1, \bar{\mathbf{P}}^2, \cdots, \bar{\mathbf{P}}^K \leftarrow \text{ClusterMean}(\hat{\mathbf{P}}^1, \hat{\mathbf{P}}^2, \cdots, \hat{\mathbf{P}}^K)$
  $\mathbf{P}^C_{t+1} := \{\mathbf{P}^1_{t+1}, \cdots, \mathbf{P}^{|C|}_{t+1}\} \leftarrow \text{Match}(\bar{\mathbf{P}}^1, \cdots, \bar{\mathbf{P}}^K)$
end
return $\mathbf{W}^g_\phi, \mathbf{P}^C$

Client executes:

$\text{LocalUpdate}(c, \mathbf{W}^c)$:
for each local epoch $e = 1$ to E **do**
  for each batch $\mathbf{b}$ of D^c **do**
    $\mathbf{W}^c \leftarrow \mathbf{W}^c - \eta\nabla\mathcal{L}(\mathbf{W}^c; \mathbf{b})$
  end
end
return $\mathbf{W}^c := \mathbf{W}^c_\phi \cup \mathbf{P}^c$

3.3.1 Training Mechanisms of FedSkill. The proposed *FedSkill* framework is illustrated in the left subplot of Figure 3. The global server

Algorithm 2: Local Prototype Projection of FedSkill

Input : local datasets D^c, local vector sets $\mathbf{P}^c$, client set C

for $c \in C$ **do in parallel**

  Get all non-overlapped segment representations of D^c:

  $\mathcal{R}_{train} = [[\textbf{Conv}\,(\mathbf{s}^{(j)}_{t-m+1:t})]_{t=m}]^{|D^c|}_{j=1}$

  for each parameterized vector $\mathbf{p}^c_i \in \mathbf{P}^c$ **do**

    $\mathbf{p}_i \leftarrow \underset{\mathbf{r}^{(j)}_{t=m} \in \mathcal{R}_{train}}{\arg\min} \left\| \mathbf{p}_i - \mathbf{r}^{(j)}_{t=m} \right\|^2_2$

  end

end

return $\mathbf{P}^c, \forall c \in C$

initializes and distributes the global interpretable skill learning model to local clients at the beginning of global training. As aforementioned, the model is partitioned into two sets of parameters, one set is denoted as $\mathbf{P}^g$, containing the parameterized vectors in the prototype layer. The other set is denoted as $\mathbf{W}^g$, containing the weights of the convolution layer and imitation learning layer.

For each global epoch, the server aggregates the local models from clients after local training. We use existing mechanisms (FedAvg[24] and FedProx[19]) to aggregate $\mathbf{W}^g$ that generates segment representation and the final action. However, the parameterized vector sets can differ a lot due to the heterogeneity of expert demonstrations across different clients. This discrepancy will lead to a misalignment between local and global skills and thus aggregating $\mathbf{P}^g$ with FedAvg[24] or FedProx[19] may yield sub-optimal prototypes for inference and reasoning.

To this end, we propose a knowledge alignment mechanism on the server side, as shown in the right subplot of Figure 3. After receiving $k \times |C|$ parameterized vectors, the server performs clustering (e.g., K-means and Gaussian Mixture Models with K clusters/components, where we adopt $K = k$) to identify the membership of each vector, and the vectors containing similar prototypical/skillful representations are aligned to the same cluster. After that, the centroid vector that represents a mode of skill is obtained by the mean of all vectors that belongs to the same cluster. Finally, each local vector is matched by the global centroid vector based on the identified membership, and distributed back to local clients. As a result, each client owns a specific vector set that best represents the prototypical information from its own data through the entire training process, where the skillful knowledge is shared and aligned across different clients. The detailed training algorithm of FedSkill is presented in Algorithm 1.

3.3.2 Local Prototype Projection. Even if the training is completed and the local parameterized vector is well-regularized to be skillful, the returned local client model is not readily interpretable. This is because these vectors are just close approximations (due to $\mathcal{L}_e$ in Section 3.2.4) and do not associate with any actual data that represents explicit prototypes. To enable the interpretability of each client's skill learning model, we perform local prototype projection to each parameterized vector, by assigning it to the training segment that has the smallest L_2-distance in the representation space. The detailed process is depicted in Algorithm 2. Only after this stage are the vectors translated to explicit prototypes the local model can capture the varying patterns and construct a meaningful skill embedding based on these prototypes. We emphasize that this step is crucial but privacy-sensitive. That is why we only perform this step once for each client during the inference and reasoning stage (unlike [25] that performs it every few training epochs). This preserves the privacy of expert demonstrations, as no data or representations are leaked to the global server and other clients.

Table 1: Statistics of Datasets

Dataset	#Trajectories	#Features	#Actions
Comorbidity	8447	47	23
Sepsis	11419	43	5
Grid-world	14195	4	4

4 EXPERIMENTS

In this section, we evaluate the effectiveness of the proposed FedSkill framework. Specifically, we first introduce the datasets, evaluation metrics, and experiment setups including training details and model specifications of FedSkill and related baseline methods. Then, we compare FedSkill with 4 state-of-the-art baselines over 3 datasets to demonstrate its effectiveness and interpretation capability.

4.1 Datasets and Evaluation Metrics

4.1.1 Datasets and Evaluation Metrics. We evaluate our method on two benchmark datasets for dynamic treatment recommendations (DTR) and a synthetic grid-world dataset. The DTR datasets are two subsets of a public Electronic Health Record database MIMIC-III [15]. It contains the records of 43,000 unique patients in intensive care units between 2001 and 2012, which covers 6,695 diseases and 4,127 medications. A sequence of the patient states and the doctor's medications form a demonstration trajectory. The patient state consists of static features of demographics and admission information, and dynamic features of historical treatment, lab test values as well as vital signs. We follow the sepsis-3 criteria[30] to extract the demonstration trajectories of the sepsis patients and follow the procedures in [2] to extract the demonstration trajectories of patients with comorbidity. Note that the Sepsis data contains the trajectories of survival and deceased patients, while only the survival trajectories are used to evaluate the methods. The synthetic data is sampled from an expert agent in the 13×13 grid-world[3] environment, which has four rooms separated by walls and connected through four hallways. The expert agent can choose among four directions: up, down, left, and right. We define the task that the agent at the starting state performs exploration to reach the target state, where the staring states and target states are initialized in two different rooms along the left diagonal. Table 1 summarizes the three datasets' statistics. The segment length for Comorbidity, Sepsis, and Grid-world datasets are set as 2,4,8, respectively. The selection of segment length is based on the empirical pattern analysis of three datasets with aligned performance evaluations.

To evaluate the learned model regarding the consistency between the predicted actions and those from expert demonstrations, we adopt three commonly used metrics: the Jaccard coefficient, the micro and macro average of AUC-ROC scores, denoted as Jaccard, MI-AUC, and MA-AUC, respectively. Note that we follow [37] and

Table 2: Hyperparameters of Interpretable Skill Learning Model

Dataset	#Prototypes	#Dimension	λ_1	λ_2	λ_3	d_{min}
Comorbidity	28	48	0.15	0.15	0.05	0.8
Sepsis	30	64	0.20	0.15	0.05	1.0
Grid-world	20	4	0.25	0.25	0.15	1.0

evaluate the models only on positive trajectories where the expert succeeds in fulfilling the task, when negative ones are available.

4.2 Experiment Setup

4.2.1 Settings of Imitation Learning Models. We compared our interpretable skill learning model with the following 4 baseline imitation learning methods:

- **Behavior Cloning (BC)** [1]: It divides the trajectories into state-action pairs and learns a policy from the step-wise expert demonstrations via supervised learning.
- **GAIL** [11]: GAIL learns a policy in an adversarial manner, where the policy network serves as a generator to mimic expert trajectories and receives the reward signal provided by the discriminator.
- **ACIL** [37]: ACIL exploits the information from both positive and negative trajectories by introducing another cooperative discriminator and training objective based on GAIL, so that the distributions of the experts' and learned demonstrations are far from that of negative ones.
- **ConvBC**: We implement a contextual policy network conditioned on the segment representation, where the representation is generated by convolution neural networks that capture the varying patterns in trajectories.

For fair comparisons, the policy networks of the above methods are the same 3-layer Multilayer Perceptron (MLP) with the same neuron size and activation function at each layer. The discriminators in GAIL and ACIL are also the same 3-layer MLP. The three datasets are split for training/validation/testing sets by a ratio of 6/2/2, based on the IDs of trajectories. The batch sizes for Comorbidity, Sepsis, and Grid-world datasets are 64, 64, and 256, respectively. The detailed hyperparameter configurations of our interpretable skill learning model for three datasets are presented in Table 2 (selected based on the best performance on the validation set).

4.2.2 Settings of Federated Frameworks. Based on the above imitation learning models, we implement several federated imitation learning baselines under FedAvg [24] and FedProx [19] frameworks. We also implement two variants based on the FedSkill, $\text{FedSkill}_{\text{Avg}}$ and $\text{FedSkill}_{\text{Prox}}$ for the comparison of performance under different federated mechanisms. K-means clustering is used in knowledge alignment. The experiments are performed under three clients with equal data partition, which is also performed at the granularity of trajectory. The number of local training epochs is 3, where the Adam optimizer[17] is used for the local model update.

4.3 Performance Evaluation

The performance evaluation results of FedSkill and baselines are shown in Table 3. As there are no negative trajectories in the extracted Comorbidity and Grid-world datasets, the performance of ACIL is not reported. Moreover, for the variants of FedSkill, we report the final performance of the global model that doesn't have the interpretation capability (i.e., $\text{FedSkill}_{\text{avg}}$ (Global) and $\text{FedSkill}_{\text{pro}}$ (Global)), as well as the averaged performance of client models that perform local prototype projection (i.e., $\text{FedSkill}_{\text{avg}}$ (Local) and $\text{FedSkill}_{\text{pro}}$ (Local)). As the FedSkill contains a client-specific parameter set for prototype learning, the notion of the global model does not exist. Thus we only report the clients' average performance, i.e., FedSkill.

According to the results on three datasets, several observations can be made. First, The GAIL outperforms BC on three datasets, which demonstrates the advantage of adversarial imitation learning. ACIL has slightly better performance than GAIL, as it refines the policy by preventing it from taking actions that lead to undesired results in negative trajectories. Second, the Conv-BC outperforms other baselines by a clear margin as it captures the variability of different trajectories in the representation space, which is helpful for downstream imitation learning tasks. Third, our proposed method and variants constantly achieve the best or second-best results on two DTR datasets, which shows the advantage of our proposed FedSkill technique in complex tasks. Besides, the FedSkill generally results in the best performance of client models compared to other variants, suggesting the importance of skill alignment in global information utilization. Compared to our method, ConvBC sometimes gains an advantage as it does not impose constraints on representation space. Meanwhile, it is not able to render an interpretable structure to explain the underlying skills in the demonstration trajectories. Our method also demonstrates its effectiveness in simple tasks from simulated environments, even if it slightly compromises the performance to obtain interpretability. To summarize, our method achieves state-of-the-art performance while maintaining the capability of skill discovery and interpretation.

4.4 Interpretable Prototypes under Federation

Next, we present how our method renders interpretable prototypes that establish the later reasoning process. Our demonstration is based on the DTR data.

4.4.1 Representative Prototype Learning under Federated Learning. We compare the representation space of local training data from the same client, which is generated by our method and ConvBC, respectively. The visualization of segment representations via t-SNE [34] and PCA plots are shown in Figure 4. The dots and stars stand for segments and prototypes in the representation space, where the color indicates the statistics of ground truth action segments. We use the mode value for the analysis of Comorbidity data, and the mean value for that of Sepsis data as different actions in Sepsis indicate different treatment levels based on the same medication (intravenous(IV) fluid). We can observe that Federated ConvBC somewhat separates segment representations that contain different expert behaviors, but these segments are loosely structured and distributed in the dispersed representation space. As aforementioned, this method does not have the notion of prototypes and thus fails to interpret the varying patterns.

In contrast, the segment representations generated by our model form clear clusters for different skills. Moreover, most prototypes are correctly projected and consistently matched to the nearby

Table 3: Performance comparison of different methods. The results are reported based on 5 rounds of experiments. Note that the performance of ACIL is not reported ("—") on Comorbidity and Grid-world datasets since they do not have negative trajectories.

Method	Aggregation	Comorbidity			Sepsis			Grid-world		
		Jaccard	MI-AUC	MA-AUC	Jaccard	MI-AUC	MA-AUC	Jaccard	MI-AUC	MA-AUC
BC	FedAvg	0.6161±0.0105	0.9415±0.0014	0.9287±0.0015	0.5263±0.0011	0.7862±0.0004	0.7640±0.0002	0.9170±0.0021	0.9648±0.0002	0.9643±0.0002
	FedProx	0.6084±0.0072	0.9403±0.0010	0.9278±0.0004	0.5275±0.0012	0.7870±0.0009	0.7643±0.0012	0.9188±0.0040	0.9645±0.0005	0.9641±0.0004
GAIL	FedAvg	0.6271±0.0059	0.9417±0.0025	0.9280± 0.0026	0.5303±0.0012	0.7887±0.0008	0.7660±0.0005	0.9233±0.0011	0.9650±0.0003	0.9647±0.0003
	FedProx	0.6211±0.0082	0.9416±0.0017	0.9282±0.0014	0.5305±0.0015	0.7879±0.0007	0.7651±0.0006	0.9245±0.0028	0.9651±0.0003	0.9647±0.0003
ACIL	FedAvg	—	—	—	0.5310±0.0018	0.7883±0.0007	0.7656±0.0005	—	—	—
	FedProx	—	—	—	0.5312±0.0008	0.7889±0.0008	0.7660±0.0005	—	—	—
ConvBC	FedAvg	0.6780±0.0083	0.9551±0.0014	0.9422±0.0016	0.5517±0.0014	0.8106±0.0014	0.7856±0.0006	0.9255±0.0031	0.9659±0.0007	0.9655±0.0009
	FedProx	0.6738±0.0115	0.9540±0.0028	0.9398±0.0024	0.5514±0.0022	0.8125±0.0014	0.7875±0.0019	0.9287±0.0019	0.9664±0.0015	0.9661±0.0017
FedSkill$_{\mathbf{Avg}}$ (Global)		0.6771±0.0060	0.9531±0.0013	0.9401±0.0012	0.5513±0.0013	0.8068±0.0014	0.7855±0.0016	0.9218±0.0053	0.9652±0.0015	0.9651±0.0016
FedSkill$_{\mathbf{Avg}}$ (Local)		0.6742±0.0082	0.9527±0.0018	0.9390±0.0015	0.5508±0.0011	0.8060±0.0021	0.7849±0.0012	0.9198±0.0026	0.9648±0.0012	0.9641±0.0015
FedSkill$_{\mathbf{Prox}}$ (Global)		0.6658±0.0142	0.9514±0.0028	0.9377±0.0036	0.5515±0.0016	0.8071±0.0024	0.7846±0.0011	0.9216±0.0040	0.9651±0.0010	0.9648±0.0014
FedSkill$_{\mathbf{Prox}}$ (Local)		0.6654±0.0138	0.9510±0.0033	0.9373±0.0041	0.5491±0.0005	0.8052±0.0042	0.7828±0.0038	0.9203±0.0021	0.9649±0.0012	0.9641±0.0009
FedSkill		0.6803±0.0083	0.9543±0.0009	0.9404±0.0011	0.5511±0.0013	0.8083±0.0015	0.7858±0.0014	0.9201±0.0066	0.9650±0.0017	0.9644±0.0018

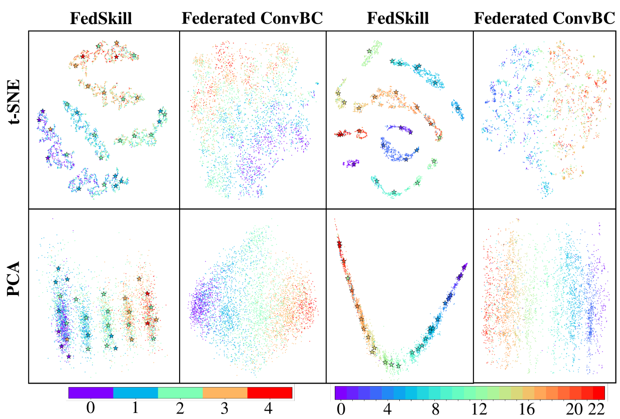

Figure 4: Visualizations of segment representations and prototypes from the same client (Leftmost two columns: Sepsis dataset; Rightmost two columns: Comorbidity dataset)

segments for each cluster. Although some mixed-up segments and mismatched prototypes with different treatment levels exist in Sepsis data, the deviation is within a similar level. The visualization results demonstrate that our method is able to exploit the variability of expert behaviors and render representative prototypes that contain skillful information.

4.4.2 Effective Prototype-Segment Association. To further demonstrate the effectiveness of projection and interpretability of learned prototypes under federated learning, we use the Sepsis data as the context and provide the cross-client visualizations of the raw segments that are associated with prototypes, as shown in Figure 5 (a-b). Each segment is presented as a heat map describing how the state feature in the vertical axis changes across different steps in the horizontal axis, where the color indicates the magnitude of min-max scaled features. In addition to the prototypes, their neighboring segments in the representation space are also presented. We have two main observations based on the visualizations. First, the patterns of prototype segments that represent different treatment levels can be generally distinguished, and some similarities between prototype and neighboring segments can be observed. This observation enhances our previous conclusion that our method learns representative prototypes. Second, we find that some prototypes indicating the same treatment level also have similar patterns across different clients. This demonstrates the effectiveness of FedSkill in terms of global information exploitation from multiple data silos.

4.5 Case Study: Trajectory Analysis in a Heterogeneous Environment

In previous discussions, we have shown the capability of FedSkill regarding the generation of interpretable prototypes that capture different skills. Now we demonstrate the reasoning process by analyzing the constructed skill embedding from prototypes. Note that we simulate a Non-IID data partition scenario on Sepsis data with a case study to show the effectiveness of FedSkill in a heterogeneous environment, and compare it with a single client model trained on its own data. To this end, we follow [8, 21] and use Dirichlet distribution with concentration parameter $(0.4, 0.4, 0.4)$ under 3 clients to perform data partition based on the mean value of ground truth actions of trajectories. It results in different action distributions and different numbers of trajectories across clients simultaneously. We explore the most skewed client with 6.89%(288) training trajectories which are dominated by treatment levels 2(106) and 3(167) with a few of levels 0(7) and 4(8).

We visualize the skills and agent actions on a selected testing trajectory with 12 steps (3 segments) that has simple clinical patterns for analysis with ease, as shown in the left subplot of Figure 5 (c). Three lab values from the state features are extracted, Total Protein, PaCO2, and PaO2, which are clinically important criteria to evaluate patient status. The background of each segment is colored by the treatment level suggested by the learned skills, which is referred to the same type of prototypes allocating the most weights in the skill embedding, as shown in Figure 5 (d). In segment 1, the prototypes of treatment level 0 allocate the most weights (35.11%), thus the obtained skill suggests no usage of IV fluid in the left subplot, and so on so forth. Next, we analyze the trajectory in detail. In general, an increased Total Protein/PaO2, as well as a decreased PaCO2 suggest alleviated symptoms. In this case, the patient is first

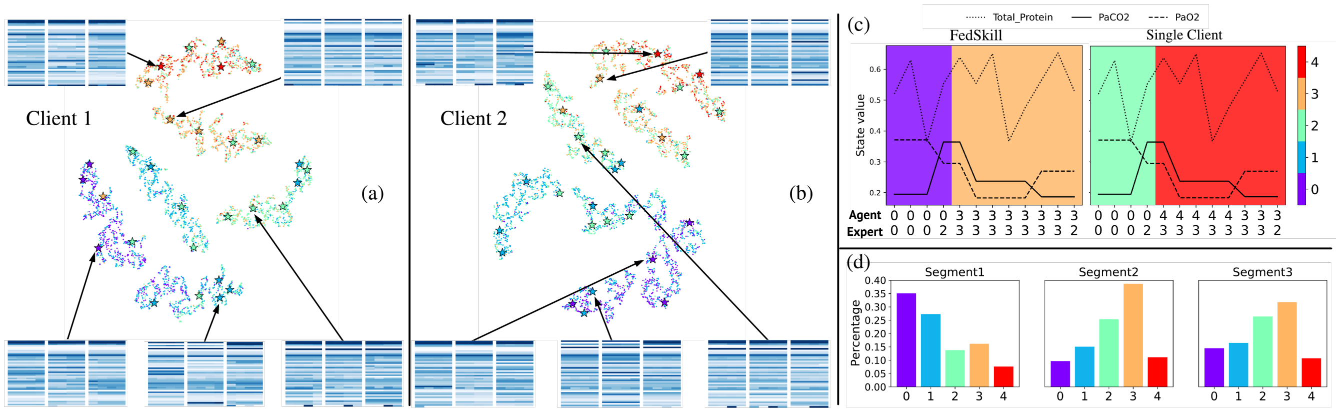

Figure 5: (a-b) Visualizations of prototypes and associated segments across different clients on Sepsis data: For each segment block, the leftmost segment represents the prototype and others are its neighbouring segments; (c) Illustration of trajectory analysis with Non-IID; (d) The allocated weights for skill embedding generation in FedSkill.

treated with a small dose of IV fluid in the first segment and the symptom worsens. Then the patient's status improves and is finally cured after constantly using higher doses of IV fluid in the second and third segments. It is clear that the skill embedding of FedSkill can capture the transition of lab values and suggest a different dose of IV fluid for treatment. Besides, the contextual policy network is able to recommend consistent treatments suggested by the skill embedding. In general, the actions and underlying skills are consistent to the expert's demonstrations in this trajectory.

We also visualize the skills and actions rendered by the single client model for comparison, as shown in the right subplot of Figure 5 (c). The single client model fails to suggest a proper dose of IV fluid in the first segment as it learns on a skewed data distribution with very few cases that represent the patient's status, which also leads to deviated treatment recommendations in the second and third segments. Meanwhile, FedSkill is still able to infer a faithful skill embedding so as to recommend appropriate treatments in this case, which suggests its effectiveness under data heterogeneity.

4.6 Base Model Study

We present a base model study on Sepsis dataset to further demonstrate the importance of learning components in our base interpretable skill learning model.

4.6.1 Hyperparameter study. We first perform an ablation study on the regularization terms that are designed to enhance interpretability, as shown in the (a-c) of Table 4. It can be observed that discarding the clustering and diversity constraints (a) in the representation space (c) degrades the model performance minorly. However, discarding the evidence constraint (b) makes the model significantly worse and more unstable as it renders erroneous prototypes for downstream imitation learning tasks. We also evaluate the effect of using a different number of prototypes, as shown in (d). It is clear that a single prototype results in inferior performance as it is inadequate to hold the needed representative skillful information. As the number of prototypes increases, the performance gets better and gradually stabilizes within a certain range. Nevertheless, it is beneficial to keep a compact prototype set for interpretation.

Table 4: Based model study on Sepsis data

	Jaccard	MI-AUC	MA-AUC
Based Model	**0.5520±0.0019**	**0.8077±0.0013**	**0.7854±0.0005**
(a) $\lambda_1 = 0$	0.5424±0.0023	0.8028±0.0012	0.7800±0.0028
(b) $\lambda_2 = 0$	0.5065±0.0316	0.7729±0.0271	0.7624±0.0126
(c) $\lambda_3 = 0$	0.5427±0.0015	0.8007±0.0014	0.7782±0.0014
(d) k = 1	0.5298±0.0018	0.7891±0.0015	0.7656±0.0013
(d) k = 10	0.5438±0.0024	0.8022±0.0009	0.7789±0.0005
(d) k = 20	0.5460±0.0026	0.8038±0.0016	0.7824±0.0011
(d) k = 40	0.5498±0.0016	0.8067±0.0013	0.7851±0.0007
(e) Hard-Skill	0.5390±0.0014	0.7975±0.0006	0.7737±0.0014

4.6.2 Soft-Skill Combination vs. Hard-Skill Selection. We explore the aforementioned hard-skill selection using a Gumbel-Softmax [14] component based on the similarity measure. It renders an inferior result, as shown in (e) of Table 4, even if we adjust the number of prototypes and tune the hyperparameters. We suggest that it is not an ideal choice for skill learning to handle unseen patterns in complex imitation learning environments.

5 CONCLUSIONS

In this paper, we proposed the FedSkill, a privacy-preserved interpretable skill learning framework for imitation learning tasks. Our proposed interpretable skill learning model is able to explicitly explain the underlying rationale of the learned skill that existed in the varying patterns from experts' demonstrations. Besides, our designed aggregation mechanism enables the utilization of global information and preserves local interpretability under the federated framework. Experiments and thorough empirical studies demonstrate promising performance as well as good interpretability under the federated setting.

ACKNOWLEDGMENTS

Yushan Jiang is partially supported by Next Gen Scholar General Electric Fellowship. Dongjin Song is partially supported by National Science Foundation (NSF CNS-2154191) and University of Connecticut (UConn) Research Excellence Program (REP) Award.

REFERENCES

[1] Michael Bain and Claude Sammut. 1995. A Framework for Behavioural Cloning.. In *Machine Intelligence 15*. 103–129.

[2] Jacek M Bajor and Thomas A Lasko. 2017. Predicting medications from diagnostic codes with recurrent neural networks. In *International conference on learning representations*.

[3] Justin Boyan and Andrew Moore. 1994. Generalization in reinforcement learning: Safely approximating the value function. *Advances in neural information processing systems* 7 (1994).

[4] Christopher Briggs, Zhong Fan, and Peter Andras. 2020. Federated learning with hierarchical clustering of local updates to improve training on non-IID data. In *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–9.

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

[6] Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. 2016. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. *Advances in neural information processing systems* 29 (2016).

[7] Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. 2018. End-to-end driving via conditional imitation learning. In *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 4693–4700.

[8] Yutong Dai, Zeyuan Chen, Junnan Li, Shelby Heinecke, Lichao Sun, and Ran Xu. 2022. Tackling Data Heterogeneity in Federated Learning with Class Prototypes. *arXiv preprint arXiv:2212.02758* (2022).

[9] Yan Duan, Marcin Andrychowicz, Bradly Stadie, OpenAI Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. 2017. One-shot imitation learning. *Advances in neural information processing systems* 30 (2017).

[10] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. 2020. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems* 33 (2020), 19586–19597.

[11] Jonathan Ho and Stefano Ermon. 2016. Generative adversarial imitation learning. *Advances in neural information processing systems* 29 (2016).

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

[13] Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. 2018. Reward learning from human preferences and demonstrations in atari. *Advances in neural information processing systems* 31 (2018).

[14] Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144* (2016).

[15] Alistair E W Johnson, Tom J Pollard, Lu Shen, Li-Wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data* 3 (May 2016), 160035. https://doi.org/10.1038/sdata.2016.35

[16] Andrej Karpathy, Justin Johnson, and Li Fei-Fei. 2015. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078* (2015).

[17] Diederik P Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR (Poster)*.

[18] Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. 2018. Neural relational inference for interacting systems. In *International conference on machine learning*. PMLR, 2688–2697.

[19] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems* 2 (2020), 429–450.

[20] Yunzhu Li, Jiaming Song, and Stefano Ermon. 2017. Infogail: Interpretable imitation learning from visual demonstrations. *Advances in Neural Information Processing Systems* 30 (2017).

[21] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. 2020. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems* 33 (2020), 2351–2363.

[22] Boyi Liu, Lujia Wang, Ming Liu, and Cheng-Zhong Xu. 2020. Federated imitation learning: A novel framework for cloud robotic systems with heterogeneous sensor data. *IEEE Robotics and Automation Letters* 5, 2 (2020), 3509–3516.

[23] Sha Luo, Hamidreza Kasaei, and Lambert Schomaker. 2021. Self-imitation learning by planning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 4823–4829.

[24] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 1273–1282.

[25] Yao Ming, Panpan Xu, Huamin Qu, and Liu Ren. 2019. Interpretable and steerable sequence learning via prototypes. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 903–913.

[26] Jingchao Ni, Zhengzhang Chen, Wei Cheng, Bo Zong, Dongjin Song, Yanchi Liu, Xuchao Zhang, and Haifeng Chen. 2021. Interpreting Convolutional Sequence Model by Learning Local Prototypes with Adaptation Regularization. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 1366–1375.

[27] Stuart Russell. 1998. Learning agents for uncertain environments. In *Proceedings of the eleventh annual conference on Computational learning theory*. 101–103.

[28] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. 2020. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems* 32, 8 (2020), 3710–3722.

[29] Mohit Sharma, Arjun Sharma, Nicholas Rhinehart, and Kris M Kitani. [n. d.]. Directed-Info GAIL: Learning Hierarchical Policies from Unsegmented Demonstrations using Directed Information. In *International Conference on Learning Representations*.

[30] Mervyn Singer, Clifford S Deutschman, Christopher Warren Seymour, Manu Shankar-Hari, Djillali Annane, Michael Bauer, Rinaldo Bellomo, Gordon R Bernard, Jean-Daniel Chiche, Craig M Coopersmith, et al. 2016. The third international consensus definitions for sepsis and septic shock (Sepsis-3). *Jama* 315, 8 (2016), 801–810.

[31] Jiaming Song, Hongyu Ren, Dorsa Sadigh, and Stefano Ermon. 2018. Multi-agent generative adversarial imitation learning. *Advances in neural information processing systems* 31 (2018).

[32] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. 2022. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[33] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. 2022. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 8432–8440.

[34] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[36] Lu Wang, Ruiming Tang, Xiaofeng He, and Xiuqiang He. 2022. 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*. 1081–1089.

[37] Lu Wang, Wenchao Yu, Xiaofeng He, Wei Cheng, Martin Renqiang Ren, Wei Wang, Bo Zong, Haifeng Chen, and Hongyuan Zha. 2020. Adversarial Cooperative Imitation Learning for Dynamic Treatment Regimes. In *Proceedings of The Web Conference 2020*. 1785–1795.

[38] Tianhao Wu, Mingzhi Jiang, Yinhui Han, Zheng Yuan, and Lin Zhang. 2021. Density-Aware Federated Imitation Learning for Connected and Automated Vehicles with Unsignalized Intersection. *arXiv preprint arXiv:2105.01889* (2021).

[39] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. 2020. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 753–763.

[40] Bo Yang, Huaguang Shi, and Xiaofang Xia. 2022. Federated Imitation Learning for UAV Swarm Coordination in Urban Traffic Monitoring. *IEEE Transactions on Industrial Informatics* (2022).

[41] Ao Yu, Qingkai Yang, Lihua Dou, and Mohamed Cheriet. 2021. Federated imitation learning: A cross-domain knowledge sharing framework for traffic scheduling in 6G ubiquitous IoT. *IEEE Network* 35, 5 (2021), 136–142.

[42] Xin Zhang, Yanhua Li, Ziming Zhang, and Zhi-Li Zhang. 2020. f-gail: Learning f-divergence for generative adversarial imitation learning. *Advances in neural information processing systems* 33 (2020), 12805–12815.

[43] Zhejun Zhang, Alexander Liniger, Dengxin Dai, Fisher Yu, and Luc Van Gool. 2021. End-to-end urban driving by imitating a reinforcement learning coach. In *Proceedings of the IEEE/CVF international conference on computer vision*. 15222–15232.

[44] Boyuan Zheng, Sunny Verma, Jianlong Zhou, Ivor W Tsang, and Fang Chen. 2022. Imitation learning: Progress, taxonomies and challenges. *IEEE Transactions on Neural Networks and Learning Systems* (2022), 1–16.

[45] Wei Zhu, Dongjin Song, Yuncong Chen, Wei Cheng, Bo Zong, Takehiko Mizoguchi, Cristian Lumezanu, Haifeng Chen, and Jiebo Luo. 2022. Deep Federated Anomaly Detection for Multivariate Time Series Data. In *2022 IEEE International Conference on Big Data (Big Data)*. 1–10. https://doi.org/10.1109/BigData55660.2022.10064694

[46] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. 2008. Maximum entropy inverse reinforcement learning.. In *Aaai*, Vol. 8. Chicago, IL, USA, 1433–1438.

[47] Konrad Zolna, Scott Reed, Alexander Novikov, Sergio Gomez Colmenarejo, David Budden, Serkan Cabi, Misha Denil, Nando de Freitas, and Ziyu Wang. 2021. Task-relevant adversarial imitation learning. In *Conference on Robot Learning*. PMLR, 247–263.