A Generalized Apprenticeship Learning Framework for Capturing Evolving Student Pedagogical Strategies

Md Mirajul Islam[1 [0000−0001−5086−7399]], Xi Yang[2 [0000−0003−0026−9096]], Rajesh Debnath[1 [0000−0002−5460−9773]], Adittya Shoukarjya Saha[1 [0000−0001−6344−9663]], and Min Chi[1 [0000−0003−1765−7837]]

[1] North Carolina State University, Raleigh, NC 27606, USA
{mislam22, rdebnat, asaha4, mchi}@ncsu.edu
[2] IBM Research, Yorktown Heights, NY 10598, USA
xi.yang@ibm.com

Abstract. Reinforcement Learning (RL) and Deep Reinforcement Learning (DRL) have advanced rapidly in recent years and have been successfully applied to e-learning environments like intelligent tutoring systems (ITSs). Despite great success, the broader application of DRL to educational technologies has been limited due to major challenges such as *sample inefficiency* and difficulty designing the *reward function*. In contrast, Apprenticeship Learning (AL) uses a few *expert demonstrations* to infer the **expert's underlying reward functions** and derive decision-making policies that generalize and replicate optimal behavior. In this work, we leverage a generalized AL framework, THEMES, to induce effective pedagogical policies by capturing the complexities of the expert student learning process, where **multiple reward functions may dynamically evolve over time**. We evaluate the effectiveness of THEMES against six state-of-the-art baselines, demonstrating its superior performance and highlighting its potential as a powerful alternative for inducing effective pedagogical policies and show that it can achieve high performance, with an AUC of 0.899 and a Jaccard of 0.653, using only 18 trajectories of a previous semester to predict student pedagogical decisions in a later semester.

Keywords: Apprenticeship Learning · Pedagogical Policy · Reinforcement Learning · Evolving Reward Function

1 Introduction

In most STEM e-learning environments, the system follows a sequential decision-making process, deploying a *pedagogical policy* to determine *what* action to take and *when* at each discrete time step. In such environments, student-system interactions can be modeled as sequential decision-making problems under uncertainty, formulated within the framework of Reinforcement Learning (RL)—a paradigm that optimizes long-term rewards without requiring knowledge of the 'best' decisions at each immediate time step [44]. Recent research has demonstrated the effectiveness of RL and Deep RL (DRL) in deriving data-driven pedagogical policies to enhance student learning in Intelligent Tutoring Systems (ITSs) (e.g., [2, 25, 33, 42]). Despite their success, several challenges hinder its broader application in e-learning systems. One major challenge is **sample inefficiency**, as classic DRL algorithms like Deep Q-Networks (DQN) require millions of interactions to learn effective policies. Many studies attempt to derive pedagogical policies from fewer than 3,000 student-ITS interactive trajectories, often leading to ineffective results [5, 43]. Another key challenge is **reward function** design, which is crucial for guiding RL agents but is difficult to define accurately, especially in human-centric domains like education [26]. Manually designing reward functions is labor-intensive, prone to expertise blind spots, and often results in misspecified objectives that misalign with intended policies [16, 1, 3].

Unlike RL and DRL, **Apprenticeship Learning (AL)** is a machine learning approach that derives decision-making policies from a limited set of *expert demonstrations* by inferring the **expert's underlying reward functions**, enabling the system to generalize and replicate optimal behavior [1]. It assumes expert agents make *optimal* or *near-optimal* decisions based on an underlying reward function, allowing the apprentice to efficiently approximate expert behavior with fewer training samples [1, 54]. Existing AL approaches are typically online, requiring iterative interactions with the environment to collect new data and update the model [1, 54, 21, 12]. However, executing a potentially flawed policy can be costly or even dangerous in human-centric tasks such as education [32], making offline policy induction from expert demonstrated highly desirable [4, 36, 39, 46, 22]. In this work, **our "expert" demonstrations** were selected by allowing students to make pedagogical decisions, and then choosing only those in which the student greatly benefited from the learning environment. To accurately model students' complex and dynamic decision-making processes, it is crucial to incorporate *evolving reward functions* that reflect students' changing learning needs and adaptive strategies across different learning phases. Previous offline AL methods typically assume a single or fixed reward function across trajectories, failing to capture the evolving goals of students. For example, while energy-based distribution matching (EDM) [24] has advanced offline AL, it assumes a homogeneous policy with a single reward function [7]. Approaches like EM-IRL [52] address reward function heterogeneity by clustering students based on pedagogical behavior but struggle with large continuous state spaces. More recently, EM-EDM [22] extends EDM with an Expectation-Maximization(EM) framework, enabling policy induction from expert demonstrations driven by multiple heterogeneous reward functions while generalizing to continuous state spaces.

To directly induce effective pedagogical policies from expert students' interactions with an ITS, we leverage **THEMES** [50], a **T**ime-aware **H**ierarchical **EM** **E**nergy-based **S**ubtrajectory AL framework which is designed to model *multiple* reward functions *evolving* over time in an *offline* manner. THEMES consists of two key components: **(1) Sub-trajectory partitioning**, achieved by partitioning expert trajectories into fine-grained sub-trajectories while incorporating time-awareness. A hierarchically learned reward regulator captures evolving decision-making patterns. **(2) Policy induction**, performed through offline EM-EDM, which simultaneously clusters sub-trajectories and induces policies specific to each cluster. We evaluate THEMES against six competitive AL baselines and two ablations on a challenging task—predicting students' pedagogical decisions in an ITS by training on past semester trajectories and testing on a later semester. Results show that THEMES consistently outperforms all baselines and ablations across performance metrics. Our findings highlight its potential as a powerful alternative for capturing evolving student pedagogical strategies and inducing effective policies, even with as few as 18 demonstrations.

2 Related Work

RL for Pedagogical Policy Induction: Prior research has demonstrated the great potential of leveraging RL and DRL for pedagogical policy induction in ITSs [25, 33, 42, 49]. Shen et al. [43] applied a value iteration algorithm to induce policies aimed at improving student performance, showing advantages over random decision-making. Wang et al. [49] utilized DRL to optimize students' learning in an educational game, outperforming traditional RL approaches in simulations. More recently, Zhou et al. [53] employed offline Hierarchical RL to induce pedagogical policies, achieving superior results compared to flat-RL baselines in classroom studies. However, these approaches require large-scale training data—often exceeding 1,000 trajectories [43, 25, 53]—which is typically collected by constraining ITSs to make randomized yet reasonable decisions. Moreover, they rely on predefined reward functions, such as normalized learning gains [43, 53], manually designed by researchers [16], limiting adaptability to diverse student learning processes.

Offline Apprenticeship Learning (AL): The most straightforward approach to *offline* AL is behavior cloning (BC) [15], which directly maps states to actions to replicate demonstrated behaviors [37]. More robust alternatives, such as inverse reinforcement learning (IRL) [27, 23, 31, 8] and adversarial imitation learning [28, 29], infer an explicit or implicit reward function to generalize beyond observed behaviors. Standard IRL involves iterative online computations, including reward function inference, policy induction via RL, policy rollouts, and reward updates based on behavior divergence. To eliminate policy rollouts, batch-IRL methods [40] have been introduced, but they often rely on off-policy evaluation, leading to suboptimal solutions. Similarly, adversarial imitation learning iteratively trains a generator to roll out policies and a discriminator to distinguish learned behaviors from expert demonstrations. Off-policy variants based on actor-critic algorithms [28, 29] mitigate the need for policy rollouts but inherit the instability of adversarial training [21]. To address these challenges, EDM [24] was introduced, offering a more stable solution for offline settings and outperforming both IRL and adversarial imitation learning. In this work, *EDM serves as one of our baselines.* All the aforementioned methods assume a *single* reward function across all demonstrations.

To handle scenarios where *multiple* reward functions exist across trajectories but *remain fixed within each trajectory*, several methods have been developed. Dimitrakakis and Rothkopf [10] introduced Bayesian multi-task IRL, which learns a separate reward function for each trajectory while sharing a common prior distribution. Choi et al.[9] extended this idea using nonparametric Bayesian IRL, modeling the distribution of different rewards with a Dirichlet process to allow for more flexible clustering. Babes-Vroman et al.[7] proposed an EM-based framework that iteratively assigns probabilities to demonstrations belonging to different clusters and updates the corresponding cluster-wise reward functions using Maximum Likelihood IRL.

To model reward functions that *evolve* over time, Krishnan et al. [30] introduced Hierarchical IRL (HIRL), and Hausman et al. [19] proposed Multi-modal Imitation Learning. More recently, Wang et al. [48, 47] developed a hierarchical imitation learning model that learns both high-level policies and sub-policies for individual sub-tasks. These methods, often relying on generative adversarial imitation learning [21], require online interactions or off-policy evaluations, making them less suited for fully offline settings. Additionally, meta-imitation learning [13] has been explored to generalize from a few demonstrations, typically by structuring tasks into subtasks predefined by domain experts. However, manually defining subtasks is challenging in complex, uncertain environments like education. To address this, THEMES framework [50] *automatically segments trajectories into sub-trajectories*, capturing the evolving nature of reward functions in a data-driven manner.

AL for ITSs: Rafferty et al. applied IRL to infer learners' beliefs in an educational game, demonstrating that IRL could effectively recover participants' understanding of how their actions influence the environment [39]. This highlighted IRL's potential for interpreting data in interactive learning settings. In subsequent work, they used IRL to assess learners' mastery of algebraic skills, identifying misconceptions and providing personalized feedback for improvement [38]. Yang et al. introduced EM-Inverse Reinforcement Learning (EM-IRL) to cluster students based on pedagogical behavior [52]. However, like many multi-intent approaches, it was constrained to discrete state representations (e.g., 17 discrete features in [52]). More recently, Islam et al. developed EM-EDM framework [22] to induce effective policies from expert demonstrations with heterogeneous reward functions while generalizing to continuous state spaces. They demonstrated that EM-EDM surpassed two DRL and four AL baselines in a challenging student modeling task for predicting pedagogical decisions. Thus, we adopt EM-EDM as a baseline to assess the effectiveness of THEMES.

3 Methods

The input dataset $\mathcal{D}^{Total}$ consists of L student-ITS interaction trajectories, each having a fixed length of n. Each trajectory d can be viewed as: $s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} \cdots s_{n-1} \xrightarrow{a_{n-1}} s_n$. Here $s_i \xrightarrow{a_i} s_{i+1}$ designates that at the i_{th} time step in d, the student's learning status was in state s_i, the student executed the action a_i and their learning status transitioned to state s_{i+1}. Since in AL, it is typically

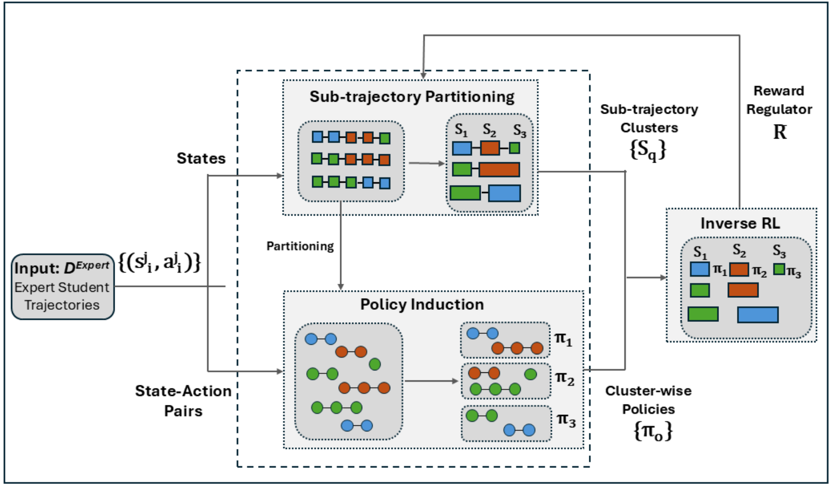


Fig. 1: Overview of the THEMES Framework.

assumed that the experts' trajectories provided as input are optimal or near-optimal according to latent reward functions [1], the quality of these trajectories is crucial for inducing more accurate policies. Therefore, we have designed a procedure outlined in Section 4 to select the expert student trajectories, based on which a subset of N trajectories $\mathcal{D}^{Expert}$ is selected from $\mathcal{D}^{Total}$, denoted as: $\{\mathcal{D}^j\} = \{(\mathbf{s}_i^j, a_i^j)|i = 1, ..., n; j = 1, ..., N\}$ where $N \leq L$.

3.1 THEMES for Evolving Student Pedagogical Strategy

Given $\mathcal{D}^{Expert}$ as input, THEMES seeks to learn the multiple underlying reward functions of students that evolve over time leveraging its two key components, *i.e.*, sub-trajectory partitioning and policy induction (Figure 1). Primarily, a time-aware sub-trajectory partitioning method [51] processes the states from input ($\mathcal{D}^{Expert}$). It partitions and clusters the states $\{\mathbf{s}_i^j\}$ into sub-trajectories such that each resulting cluster captures consistent time-invariant patterns, which can be interpreted as high-level states $\{S_q|q \in [1, Q]\}$ where Q refers to the number of sub-trajectory clusters. Then, focusing on the state-action pairs within the partitioned sub-trajectories, an Expectation-Maximization(EM) Energy-based Distribution Matching (EDM) method [24] clusters and induces policies from the state-action pairs $\{(\mathbf{s}_i^j, a_i^j)\}$ over the partitioned sub-trajectories, ensuring that each cluster exhibits consistent decision-making patterns, referred to as high-level actions $\{\pi_o|o \in [1, O]\}$. Using these high-level state-action pairs, a Maximum Likelihood Inverse Reinforcement Learning (ML-IRL) method [7] learns a high-level reward regulator $\overline{R}(\cdot)$, which is then fed back to refine sub-trajectory partitioning. This iterative procedure continues until convergence.

Sub-trajectory Partitioning As shown in Figure 1, this module first extracts the states $\{\mathbf{s}_i^j\}$ from the given expert student trajectories $\{(\mathbf{s}_i^j, a_i^j)\}$. Since the states in student trajectories are sequential and influenced by adjacent states, it is desirable to leverage the temporal dependencies rather than treating states independently. Hence, the *Subtrajectory-partitioning* module of THEMES captures this temporal dependency by investigating patterns within a sliding window of size $\omega \ll n$. For each state $\mathbf{s}_i^j$, the sub-trajectory within the window, $\mathbf{S}_i^j = \{\mathbf{s}_{i-\omega+1}^j, \ldots, \mathbf{s}_i^j\}$, is used to determine the cluster assignments $\{S_q | q \in [1, Q]\}$. Given $\mathbf{s}_i^j \in \mathbb{R}^m$, each sub-trajectory with the sliding window $\mathbf{S}_i^j$ can be viewed as a random variable in a $m\omega$-dimensional space, which can be generated by concatenating the ω consecutive states.

Each sub-trajectory cluster is represented by its mean vector (μ) and inverse covariance matrix ($\mathbf{\Theta}$) [18]. Specifically, the process of determining the optimal mean vectors, $\{\mu_q \mid q \in [1, Q]\}$, is equivalent to assigning each state to the most suitable cluster. In Intelligent Tutoring Systems (ITSs), student trajectories are often collected at irregular intervals. Consequently, both the mean vectors and inverse covariance matrices are learned across various trajectories, while *time-awareness* is incorporated during the sub-trajectory partitioning phase using a decay function that allows for a re-evaluation of cluster consistency over time [51]. To enhance the sub-trajectory clustering by accounting for decision-making behaviors, *hierarchical* rewards $\{\overline{r}_t^j \mid t \in [1, n]\}$ are inferred for each state-action pair, which then inform the adjustment of the consistency constraint. As illustrated in Figure 1, these high-level state-action pairs facilitate the inference of high-level rewards, $\overline{R}(\cdot)$, which can be learned using Maximum Likelihood Inverse Reinforcement Learning (ML-IRL) [7].

Policy Induction In this segment, EDM [24], a state-of-the-art strictly offline AL method, serves as the core component, which assumes a single reward function while learning the policy π^θ, parameterized by θ. To induce the policy π^θ, EDM aim to learn the parameter θ by minimizing the KL divergence between ρ_D and ρ_{π^θ}, $D_{KL}(\rho_D || \rho_{\pi^\theta})$, where ρ_D and ρ_{π^θ} represent the occupancy measures for the expert trajectories and for the induced policy π^θ, respectively.

Despite its success in *offline* learning, EDM assumes that all demonstrations adhere to a *single* reward function, which is often impractical for human-centric tasks in the real world, such as education, where students may have multiple reward functions evolving over time. To address the challenge of *multiple* reward functions *varying across* demonstrations, an Expectation-Maximization(EM)-EDM [22] was introduced, iteratively clustering demonstrations in the *E-step* and inducing the policy for each cluster by EDM in the *M-step*. The input to EM-EDM will be the sub-trajectories $\{\mathcal{D}^{\hat{j}} | \hat{j} = 1, ..., \hat{N}\}$, where $\hat{N}$ is the total number of sub-trajectories learned from *Sub-trajectory partitioning*. Finally, THEMES yields sub-trajectory clusters with respective policies $\{\pi_o | o \in [1, O]\}$ as output, representing multiple student pedagogical strategies evolving over time.

3.2 Baselines

Behavior Cloning (BC) Behavior cloning [45, 41] is a supervised learning approach used in AL where an agent learns to imitate an expert by directly mapping observations to actions based on the expert's demonstrations. The objective is to mimic the expert's behavior, hence, learn a policy $\pi_\theta(a|s)$ parameterized by

θ that approximates the expert behavior: $L(\theta) = \frac{1}{n}\sum_{i=1}^{n}\|\pi_\theta(a_i|s_i) - a_i\|^2$. It is important to note that while BC is straightforward, it can be sensitive to distribution shifts and compounding errors, especially when the learned policy deviates from the expert demonstrations.

Gaussian Process (GP) + Deep Q-Network (DQN) To infer immediate rewards from the final delayed reward of each trajectory, this method [6] first employs GP to model a distribution function f that captures the expected values and standard deviations of the unknown intermediate rewards. The inference is formulated as a minimum mean square error (MMSE) estimation problem, where the additive Gaussian rewards are estimated such that their sum approximates the observed delayed return [17]. This approach assumes that rewards at each timestep follow a Gaussian distribution and that the total reward across the trajectory equals the observed delayed outcome. Once immediate rewards are inferred, a Deep Q-Network (DQN) is trained on the reconstructed immediate reward-labeled trajectories. This enables the agent to learn a policy that maximizes the expected cumulative reward over the reconstructed trajectories.

Multi-modal Imitation Learning (MIL) MIL [19] tackles imitation learning from unstructured and unlabeled demonstrations by learning a multi-modal policy. Instead of assuming demonstrations are pre-segmented by task, MIL jointly segments skills and learns their corresponding policies. The method augments the standard policy $\pi(a|s)$ with a latent intention variable $i \sim p(i)$, resulting in a conditional policy $\pi(a|s,i)$. The intention variable i selects a specific mode of the policy corresponding to a distinct skill. MIL optimizes a generative adversarial imitation learning objective, extended with an additional latent intention loss, encouraging the policy to maximize the mutual information between the latent intention and the generated behaviors. Through this framework, MIL simultaneously discovers the distinct skills present in demonstrations and learns to imitate them using a single, unified policy.

Adapted Hierarchical Inverse Reinforcement Learning (AHIRL) HIRL [30] addresses the challenge of long-horizon tasks with delayed rewards by segmenting demonstrations into a sequence of locally linear sub-tasks. Each subtask is defined by consistent changes in local linearity, detected using a Bayesian nonparametric Gaussian Mixture Model (DP-GMM) over a featurized trajectory space. To ensure sequential execution, HIRL augments the original state-space $\mathcal{S}$ with a binary vector $v \in \{0,1\}^Q$ that tracks progress across Q sub-goals. A reward function $R_\theta(s,a,v)$ is then learned using Maximum Entropy IRL over this augmented space. This enables the agent to reason about both current state and sub-task completion history.

4 Experiment

This study utilized a Probability ITS (Figure 2) deployed in the Spring and Fall semesters of a large undergraduate STEM course at a public university. Designed by domain experts and overseen by departmental committees, the system aims to teach entry-level undergraduates ten key probability principles, *e.g.*, complement theorem and Bayes' rule, through solving a series of complex problems, each of which requires 30 to 60 interactions.

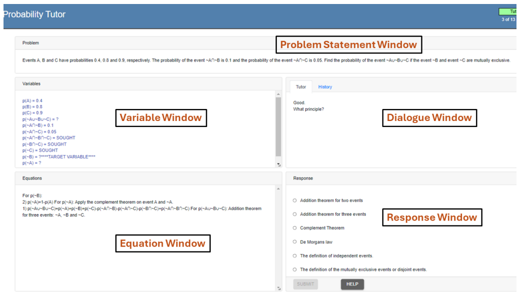
Data Collection Data were collected from 221 students over four semesters: 67 in Spring 2021 (S21), 56 in Spring 2022 (S22), 54 in Spring 2024 (S24), and 44 in Fall 2024 (F24). All students followed a standardized process, including the same textbook, pre-test, training, and post-test within our ITS. During the textbook, all students studied the 10 probability principle, read a general explanation, and worked through various single- and multi-principle problems. The pre-test and post-test comprised 8 and 12 open-ended complex problem-solving questions, respectively.

During ITS training, all students were presented with the same 12 training problems in a fixed sequence. Each problem required applying 3 to 11 domain principles for a solution. For two of these problems, all students were required to solve them independently to familiarize themselves with the ITS interface. For the remaining **10** problems, **students would decide among three approaches**: *1) solving the problem independently*, *2) collaborating with the ITS*, or *3) viewing a worked-out example*. Students' pre-test and post-test answers were graded independently by two evaluators using a double-blind process, with discrepancies resolved through discussion to reach a final grade. All test scores were normalized to the range of [0 - 100] for comparison purposes. Student data were obtained anonymously through an exempt IRB-approved protocol.

130 State Feature Representation Given that students' underlying learning status are unobservable [33], two domain experts define the ***State*** space through **130**-dimensional continuous features including: ***10*** **Autonomy** features that measure the student's work, such as the number of elicits since the last tell; ***22*** **Temporal** features that represent time-related information, like average time per step; ***31*** **Problem Solving** features that reflect contextual information about problem-solving, including difficulty level; ***57*** **Performance** features that indicate student's performance so far, such as the percentage of correct responses; ***10*** **Hints** features that show data on hint usage, such as total hints requested.

Expert Student Demonstrations To select demonstrations with optimal or near-optimal student interactions with the ITS, we used the Quantized Learning Gain (QLG) measure [34]. Students were categorized into low, medium, or high-performance groups based on their pre- and post-test scores (as shown in Figure 3). A "High" QLG is assigned to students who improved or remained in the high-performance group, while a "Low" QLG is given to those who dropped to a lower group or stayed in low/medium groups. This helped identify students who benefited most from our ITS. A total of 89 student trajectories from 4 semesters were used as expert demonstrations: 18 from S21, 24 from S22, 23 from S24, and 24 from F24.

Experimental Settings We select six competitive AL baselines for comparison against THEMES, as discussed in Section 3.1 and 3.2. The settings for the baselines are summarized as follows:



Problem

Events A, B and C have probabilities 0.4, 0.8 and 0.9, respectively. The probability of the event ~A∩~B is 0.1 and the probability of the event ~A∩~C is 0.05. Find the probability of the event ~A∪~B∪~C if the event ~B and event ~C are mutually exclusive.

Variables

p(A) = 0.4
p(B) = 0.8
p(C) = 0.9
p(~A∪~B∪~C) = ?
p(~A∩~B) = 0.1
p(~A∩~C) = 0.05
p(~A∩~B∩~C) = SOUGHT
p(~B∩~C) = SOUGHT
p(~C) = SOUGHT
p(~B) = ?****TARGET VARIABLE****
p(~A) = ?

Good.
What principle?

Equations

For p(~B):
2) p(~A)=1-p(A) For p(~A): Apply the complement theorem on event A and ~A.
1) p(~A∪~B∪~C)=p(~A)+p(~B)+p(~C)-p(~A∩~B)-p(~A∩~C)-p(~B∩~C)+p(~A∩~B∩~C) For p(~A∪~B∪~C): Addition theorem for three events: ~A, ~B and ~C.

Response

○ Addition theorem for two events
○ Addition theorem for three events
○ Complement Theorem
○ De Morgans law
○ The definition of independent events.
○ The definition of the mutually exclusive events or disjoint events.

Fig. 2: ITS Interface

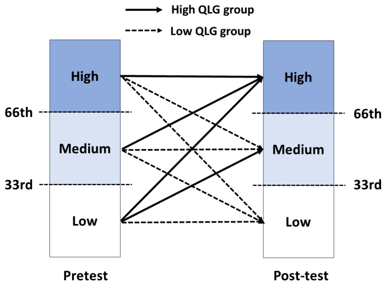


Fig. 3: Quantized Learning Gain

1) BC [15, 45]: We use the BC implementation available in the *imitation* [15] library and extend it to a customized environment corresponding to our ITS setting.
2) GP+DQN [6]: For this method, we employ Gaussian Process (GP) to infer immediate rewards first from the delayed final outcomes, e.g., learning gain of students for ITS setting, followed by a standard Deep Q-Network (DQN) implementation [11] for policy learning.
3) EDM [24]: Since EDM is strictly offline, we minimally adapt the original implementation to fit the ITS setting.
4) Multi-modal Imitation Learning (MIL) [19]: MIL leverages InfoGAN to jointly learn sub-trajectories and policies. Although originally designed for online settings, we adapt it for offline learning.
5) Adapted Hierarchical Inverse Reinforcement Learning (AHIRL) [30]: In our offline adaptation of HIRL, we reuse the segment clustering and local reward structure but employ EDM-based policy learning rather than online RL, enabling HIRL to operate effectively in a strictly offline setting.
6) EM-EDM [22]: We extend the implementation of EDM to EM-EDM by incorporating the EM algorithm and maximizing the log-likelihood of cluster assignments as described in [22].

In addition to the six AL baselines, we evaluate the performance of THEMES against its two ablations. The first one is: **THEMES**$_0$, where the high-level states are obtained through sub-trajectory partitioning [51], but the high-level actions are induced by EDM [24]. The second one is: **THEMES**$_1$, where the high-level states are also learned by sub-trajectory partitioning, but high-level actions are induced by EM-EDM [7]. Finally, in **THEMES**, we utilize reward regulated sub-trajectory clustering to partition trajectories with hierarchically learned reward regulator and induce policies over the partitioned sub-trajectories via EM-EDM. For THEMES, the hyperparameters in sub-trajectory partitioning are determined based on the Bayesian Information Criteria (BIC) [14], with the number of sub-trajectory clusters Q set to 6, window size ω set to 2, and the coefficients for sparsity and consistency terms set to 1e-3 and 4, respectively. For EM-EDM, the optimal number of clusters O was determined heuristically as 3, by iteratively implementing the EM algorithm until either empty clusters are generated or the log-likelihood of the clustering results varies by less than a predefined threshold. The iterations for the overall THEMES framework are set as 10, since the clustering likelihood for both sub-trajectory clustering and EM-EDM converge within 10 iterations.

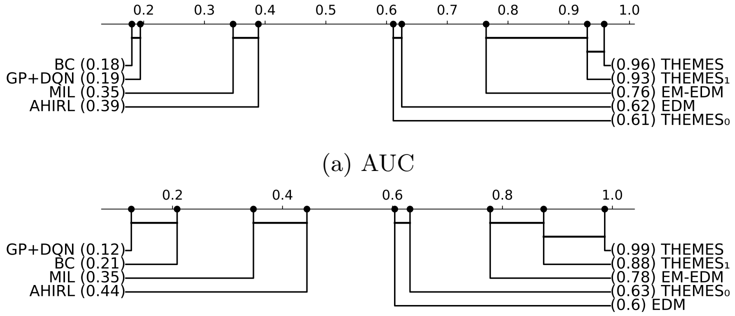
Experiment Setup and Evaluation Metrics It is important to emphasize that all models are evaluated using semester-based temporal cross-validation (3-fold) in this task, which only applied data from previous semesters individually or combined for training and is a much stricter approach than the standard cross-validation [35]. We report Accuracy, Recall, Precision, F1 Score, AUC (Area Under the ROC Curve), APR (Area Under the Precision-Recall Curve), and Jaccard score. The Jaccard score measures the similarity between the predicted and true sets of labels, calculated as the size of the intersection divided by the size of the union of the two sets. Given the nature of our task, we consider AUC and Jaccard score to be the most critical metrics, as they are generally robust.

5 Results

Table 1: Comparing THEMES with baselines and ablations

Methods	Acc	Rec	Prec	F1	AUC	AP	JAC
1. BC [15]	$.366_{\pm .03}$	$.366_{\pm .03}$	$.384_{\pm .04}$	$.363_{\pm .03}$	$.523_{\pm .03}$	$.362_{\pm .01}$	$.224_{\pm .02}$
2.GP+DQN [6]	$.389_{\pm .06}$	$.389_{\pm .06}$	$.399_{\pm .10}$	$.297_{\pm .05}$	$.525_{\pm .06}$	$.386_{\pm .03}$	$.193_{\pm .04}$
3. EDM [24]	$.656_{\pm .05}$	$.656_{\pm .05}$	$.686_{\pm .03}$	$.658_{\pm .05}$	$.801_{\pm .03}$	$.662_{\pm .04}$	$.506_{\pm .05}$
4. MIL [19]	$.508_{\pm .06}$	$.508_{\pm .06}$	$.486_{\pm .11}$	$.456_{\pm .06}$	$.627_{\pm .06}$	$.465_{\pm .06}$	$.319_{\pm .06}$
5. AHIRL [30]	$.545_{\pm .07}$	$.545_{\pm .07}$	$.566_{\pm .07}$	$.545_{\pm .07}$	$.688_{\pm .07}$	$.537_{\pm .07}$	$.388_{\pm .07}$
6. EM-EDM [7]	$\mathbf{.705}_{\pm \mathbf{.06}}$	$\mathbf{.705}_{\pm \mathbf{.06}}$	$\mathbf{.725}_{\pm \mathbf{.04}}$	$\mathbf{.709}_{\pm \mathbf{.05}}$	$\mathbf{.847}_{\pm \mathbf{.03}}$	$\mathbf{.724}_{\pm \mathbf{.05}}$	$\mathbf{.561}_{\pm \mathbf{.07}}$
7. THEMES_0	$.638_{\pm .05}$	$.638_{\pm .05}$	$.671_{\pm .06}$	$.642_{\pm .06}$	$.783_{\pm .05}$	$.643_{\pm .07}$	$.487_{\pm .06}$
8. THEMES_1	$.745_{\pm .04}$	$.745_{\pm .04}$	$.761_{\pm .03}$	$.747_{\pm .03}$	$.875_{\pm .02}$	$.768_{\pm .03}$	$.605_{\pm .04}$
9. THEMES	$\mathbf{.763}_{\pm \mathbf{.02}}^{*}$	$\mathbf{.763}_{\pm \mathbf{.02}}^{*}$	$\mathbf{.773}_{\pm \mathbf{.02}}^{*}$	$\mathbf{.765}_{\pm \mathbf{.02}}^{*}$	$\mathbf{.879}_{\pm \mathbf{.02}}^{*}$	$\mathbf{.773}_{\pm \mathbf{.04}}^{*}$	$\mathbf{.626}_{\pm \mathbf{.03}}^{*}$

Table 1 reports the average and standard deviation of the overall semester-based temporal cross-validation results for the different models, with the best baselines and ablations highlighted in bold, and the overall best results are highlighted with *. The critical difference diagram, based on the Conover-Friedman tests ($p = 0.05$), for AUC and Jaccard, is presented in Figure 4. The results clearly show that THEMES *significantly* outperform all other methods in all evaluation metrics. Specifically, THEMES outperforms THEMES_1, demonstrating the effectiveness of hierarchically learned *reward regulator* in incorporating decision-making patterns during sub-trajectory partitioning. Furthermore, the three THEMES-based methods consistently and significantly outperform competitive baselines such as AHIRL and MIL, which share the assumption of *multiple* reward functions *evolving* over time. This suggests that the effectiveness of high-level state representation is enhanced by incorporating *time-awareness* during sub-trajectory partitioning. When comparing THEMES and THEMES_0 with EM-EDM, significant improvements are observed, highlighting the power of high-level state representation to capture time-varying patterns across partitioned sub-trajectories. Although EM-EDM is better than EDM, which supports the assumption of *multiple* reward functions, the superior performance of THEMES over EM-EDM suggests that the assumption was insufficient to model time-varying decision-making. Moreover, THEMES demonstrates exceptional consistency between semesters, as evidenced by the minimal standard deviation in Table 1. Notably, among all individual fold results of semester-based cross validation, we found that THEMES can achieve an AUC of 0.899 and a Jaccard of 0.653, using only *18* trajectories of S21 to predict student strategies in S22. Finally, all of these results underscore the potential of THEMES as a sample-efficient alternative to capture evolving student pedagogical strategies.



(a) AUC

(b) Jaccard Score

Fig. 4: Critical difference diagram for AUC and Jaccard Score

A Case Study Figure 5 presents a t-SNE visualization of the original 130-state space from all expert student demonstrations. To facilitate better visualization, we applied t-distributed stochastic neighbor embedding (t-SNE) to project the data into a lower-dimensional space [20]. In this figure, different colors correspond to the six clusters identified by sub-trajectory clusters or high-level states in THEMES framework. The blue and black lines represent the expert student demonstrations from two distinct groups. The blue trajectory remains consistently within sub-trajectory cluster 1, indicating a student likely driven by a single, stable intention throughout their learning process. In contrast, the black trajectory begins in cluster 2 but evolves over time, transitioning through clusters 6, 1, 2, 6, and 3. This suggests that the latter student's learning is guided by multiple, shifting intentions. AL methods like EDM, which assume a single reward function for each demonstration, can effectively capture the former trajectory but may struggle to account for the dynamic nature of the latter. In contrast, THEMES leverages sub-trajectory partitioning and clustering, enabling it to capture and adapt to these evolving intentions, resulting in a more accurate policy for the latter trajectory.

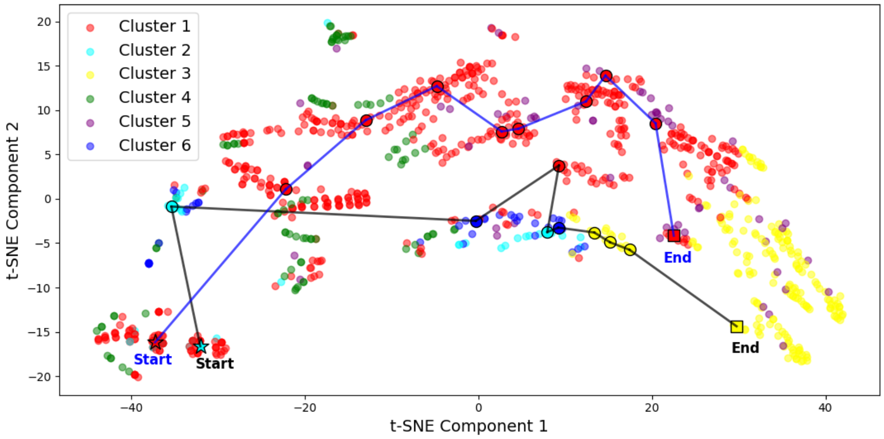


Fig. 5: t-SNE Visualization of Expert Student Demonstrations— blue Trajectory (representing a fixed reward function) vs. black Trajectory (indicating evolving reward functions)

6 Caveats, Discussion & Conclusion

We leverage a generalized time-aware hierarchical AL framework, THEMES, to induce pedagogical policies from expert student demonstrations, with the assumption that multiple reward functions evolve over time. By integrating time-awareness with hierarchically learned reward patterns, THEMES enables more precise sub-trajectory partitions, resulting in highly effective pedagogical policies that align closely with expert student decisions. However, a limitation of this work lies in the relatively limited interpretation of evolving multiple reward functions in student demonstrations. Furthermore, this study is focused on a single ITS, although the time-aware hierarchical AL approach holds significant promise for other e-learning environments. Using student-ITS interaction logs, it can potentially validate the presence of multiple evolving intentions in diverse contexts. Moreover, while the trajectory lengths in our ITS were shorter than those in many real-world human-centric environments, THEMES still outperformed other approaches. Future work exploring e-learning environments with longer trajectories may reveal even more impressive results, further solidifying the framework's effectiveness in complex, evolving pedagogical settings.

7 Acknowledgments

This material is based on work supported by the National Science Foundation AI Institute for Engaged Learning (EngageAI Institute) under Grant No. DRL-2112635, Generalizing Data-Driven Technologies to Improve Individualized STEM Instruction by Intelligent Tutors (2013502), Integrated Data-driven Technologies for Individualized Instruction in STEM Learning Environments (1726550), CAREER: Improving Adaptive Decision Making in Interactive Learning Environments (1651909). Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

1. Abbeel, P., Ng, A.Y.: Apprenticeship learning via inverse reinforcement learning. In: Proceedings of the 21st international conference on Machine learning. p. 1. ACM (2004)
2. Abdelshiheed, M., Hostetter, J.W., Barnes, T., Chi, M.: Bridging declarative, procedural, and conditional metacognitive knowledge gap using deep reinforcement learning. In: Proceedings of the Annual Meeting of the Cognitive Science Society. No. 45 in CogSci '23' (2023)
3. Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., Mané, D.: Concrete problems in ai safety. arXiv preprint arXiv:1606.06565 (2016)
4. Asoh, H., Akaho, M.S.S., Kamishima, T., Hasida, K., Aramaki, E., Kohro, T.: An application of inverse reinforcement learning to medical records of diabetes treatment. In: ECMLPKDD2013 workshop on reinforcement learning with generalized feedback (2013)
5. Ausin, M.S., Maniktala, M., Barnes, T., Chi, M.: Exploring the impact of simple explanations and agency on batch deep reinforcement learning induced pedagogical policies. In: AIED. pp. 472–485 (2020)
6. Azizsoltani, H., Sadeghi, E.: Adaptive sequential strategy for risk estimation of engineering systems using gaussian process regression active learning. Engineering Applications of Artificial Intelligence **74**, 146–165 (2018)
7. Babes, M., Marivate, V., Subramanian, K., Littman, M.L.: Apprenticeship learning about multiple intentions. In: Proceedings of the 28th ICML. pp. 897–904 (2011)
8. Chan, A.J., van der Schaar, M.: Scalable bayesian inverse reinforcement learning. arXiv preprint arXiv:2102.06483 (2021)
9. Choi, J., Kim, K.E.: Nonparametric bayesian inverse reinforcement learning for multiple reward functions. In: Advances in Neural Information Processing Systems. pp. 305–313 (2012)
10. Dimitrakakis, C., Rothkopf, C.A.: Bayesian multitask inverse reinforcement learning. In: European workshop on reinforcement learning. pp. 273–284. Springer (2011)
11. Fan, J., Wang, Z., Xie, Y., Yang, Z.: A theoretical analysis of deep q-learning. In: Learning for Dynamics and Control. pp. 486–489. PMLR (2020)
12. Finn, C., Levine, S., Abbeel, P.: Guided cost learning: Deep inverse optimal control via policy optimization. In: International conference on machine learning. pp. 49–58. PMLR (2016)
13. Finn, C., Yu, T., Zhang, T., Abbeel, P., Levine, S.: One-shot visual imitation learning via meta-learning. In: Conference on robot learning. pp. 357–368. PMLR (2017)
14. Friedman, J., Hastie, T., Tibshirani, R.: The elements of statistical learning, vol. 1. Springer series in statistics New York (2001)
15. Gleave, A., Taufeeque, M., Rocamonde, J., Jenner, E., Wang, S.H., Toyer, S., Ernestus, M., Belrose, N., Emmons, S., Russell, S.: imitation: Clean imitation learning implementations (2022)
16. Goyal, P., Niekum, S., Mooney, R.J.: Using natural language for reward shaping in reinforcement learning. arXiv preprint arXiv:1903.02020 (2019)
17. Guo, D., Shamai, S., Verdú, S.: Mutual information and minimum mean-square error in gaussian channels. IEEE Transactions on Information Theory **51**(4), 1261–1282 (2005)

18. Hallac, D., Vare, S., Boyd, S., Leskovec, J.: Toeplitz inverse covariance-based clustering of multivariate time series data. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 215–223 (2017)
19. Hausman, K., Chebotar, Y., Schaal, S., Sukhatme, G., Lim, J.J.: Multi-modal imitation learning from unstructured demonstrations using generative adversarial nets. Advances in neural information processing systems **30** (2017)
20. Hinton, G., Roweis, S.: Stochastic neighbor embedding. In: Proceedings of the 16th International Conference on Neural Information Processing Systems. p. 857–864. NIPS'02, MIT Press, Cambridge, MA, USA (2002)
21. Ho, J., Ermon, S.: Generative adversarial imitation learning. Advances in neural information processing systems **29** (2016)
22. Islam, M.M., Yang, X., Hostetter, J., Saha, A.S., Chi, M.: A generalized apprenticeship learning framework for modeling heterogeneous student pedagogical strategies. In: Proceedings of the 17th International Conference on Educational Data Mining. pp. 131–142. International Educational Data Mining Society (2024)
23. Jain, V., Doshi, P., Banerjee, B.: Model-free irl using maximum likelihood estimation. In: Proceedings of the AAAI. vol. 33, pp. 3951–3958 (2019)
24. Jarrett, D., Bica, I., van der Schaar, M.: Strictly batch imitation learning by energy-based distribution matching. Advances in Neural Information Processing Systems **33**, 7354–7365 (2020)
25. Ju, S., Zhou, G., Abdelshiheed, M., Barnes, T., Chi:, M.: Evaluating critical reinforcement learning framework in the field. AIED pp. 215–227 (2021)
26. Kaelbling, L.P., Littman, M.L., Moore, A.W.: Reinforcement learning: A survey. Journal of artificial intelligence research **4**, 237–285 (1996)
27. Klein, E., Geist, M., Pietquin, O.: Batch, off-policy and model-free apprenticeship learning. In: European Workshop on Reinforcement Learning. pp. 285–296. Springer (2011)
28. Kostrikov, I., Agrawal, K.K., Dwibedi, D., Levine, S., Tompson, J.: Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. International Conference on Learning Representations (2018)
29. Kostrikov, I., Nachum, O., Tompson, J.: Imitation learning via off-policy distribution matching. International Conference on Learning Representations (2020)
30. Krishnan, S., Garg, A., Liaw, R., Miller, L., Pokorny, F.T., Goldberg, K.: Hirl: Hierarchical inverse reinforcement learning for long-horizon tasks with delayed rewards. arXiv preprint arXiv:1604.06508 (2016)
31. Lee, D., Srinivasan, S., Doshi-Velez, F.: Truly batch apprenticeship learning with deep successor features. arXiv preprint arXiv:1903.10077 (2019)
32. Levine, S., Kumar, A., Tucker, G., Fu, J.: Offline reinforcement learning: Tutorial, review, and perspectives on open problems. arXiv preprint arXiv:2005.01643 (2020)
33. Mandel, T., et al.: Offline policy evaluation across representations with applications to educational games. In: AAMAS. pp. 1077–1084 (2014)
34. Mao, Y., Lin, C., Chi, M.: Deep learning vs. bayesian knowledge tracing: Student models for interventions. Journal of Educational Data Mining (2018)
35. Mao, Y., Marwan, S., Price, T.W., Barnes, T., Chi, M.: What time is it? student modeling needs to know. In: Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020). pp. 171–182 (2020)
36. Pan, M., Li, Y., Zhou, X., Liu, Z., Song, R., Lu, H., Luo, J.: Dissecting the learning curve of taxi drivers: A data-driven approach. In: Proceedings of the 2019 SIAM International Conference on Data Mining. pp. 783–791 (2019)

37. Pomerleau, D.A.: Efficient training of artificial neural networks for autonomous navigation. Neural computation **3**(1), 88–97 (1991)
38. Rafferty, A.N., Jansen, R., Griffiths, T.L.: Using inverse planning for personalized feedback. EDM **16**, 472–477 (2016)
39. Rafferty, A.N., LaMar, M.M., Griffiths, T.L.: Inferring learners' knowledge from their actions. Cognitive Science **39**(3), 584–618 (2015)
40. Raghu, A., Komorowski, M., Celi, L.A., Szolovits, P., Ghassemi, M.: Continuous state-space models for optimal sepsis treatment: a deep reinforcement learning approach. In: Machine Learning for Healthcare Conference. pp. 147–163 (2017)
41. Raza, S., Haider, S., Williams, M.A.: Teaching coordinated strategies to soccer robots via imitation. In: IEEE International Conference on Robotics and Biomimetics. pp. 1434–1439 (2012)
42. Rowe, J.P., Lester, J.C.: Improving student problem solving in narrative-centered learning environments: A modular reinforcement learning framework. In: AIED. pp. 419–428. Springer (2015)
43. Shen, S., Chi, M.: Reinforcement learning: the sooner the better, or the later the better? In: UMAP. pp. 37–44. ACM (2016)
44. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. MIT press (2018)
45. Syed, U., Schapire, R.E.: Imitation learning with a value-based prior. arXiv preprint arXiv:1206.5290 (2012)
46. Wang, J., Roberts, D., Enquobahrie, A.: Inferring continuous treatment doses from historical data via model-based entropy-regularized reinforcement learning. In: Asian Conference on Machine Learning. pp. 433–448. PMLR (2020)
47. Wang, L., Tang, R., He, X., He, X.: Hierarchical imitation learning via subgoal representation learning for dynamic treatment recommendation. In: Proceedings of the Fifteenth ACM International Conference on WSDM (2022)
48. Wang, L., Yu, W., Cheng, W., Zong, B., Chen, H.: Hierarchical imitation learning with contextual bandits for dynamic treatment regimes. In: RL4RealLife Workshop in the 38 th ICML (2021)
49. Wang, P., Rowe, J., Min, W., Mott, B., Lester, J.: Interactive narrative personalization with deep reinforcement learning. In: IJCAI (2017)
50. Yang, X., Gao, G., Chi, M.: Hierarchical apprenticeship learning for disease progression modeling. In: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence. IJCAI '23 (2023)
51. Yang, X., Zhang, Y., Chi, M.: Multi-series time-aware sequence partitioning for disease progression modeling. In: Proceedings of the 30th International Joint Conference on Artificial Intelligence (2021)
52. Yang, X., Zhou, G., Taub, M., Azevedo, R., Chi, M.: Student subtyping via em-inverse reinforcement learning. International Educational Data Mining Society (2020)
53. Zhou, G., Azizsoltani, H., Ausin, M.S., Barnes, T., Chi, M.: Hierarchical reinforcement learning for pedagogical policy induction. In: AIED. pp. 544–556 (2019)
54. Ziebart, B.D., Maas, A.L., Bagnell, J.A., Dey, A.K., et al.: Maximum entropy inverse reinforcement learning. In: AAAI. vol. 8, pp. 1433–1438. Chicago, IL, USA (2008)