PedagoRelearn: A Reinforcement Learning Framework for Adaptive Cross-Cultural Competence Training

CSCE 642: Reinforcement Learning Project Proposal

Thomas F. Hallmark
Department of Teaching, Learning, & Culture
Texas A&M University
College Station, TX, USA
UIN: 902 009 173

Jun Kwon
Department of Computer Science
Texas A&M University
College Station, TX, USA
UIN: 237007680

***Abstract*—Cross-cultural competence (CCC) training faces persistent challenges in adapting to individual learning trajectories. Traditional curricula remain static, failing to adjust to learners' cognitive states or to integrate principles such as spaced repetition effectively. This project presents PedagoReLearn, a reinforcement learning agent that learns optimal pedagogical strategies for cross-cultural training through sequential decision-making in an adaptive tutoring environment. Grounded in John Dewey's progressive educational philosophy, PedagoReLearn discovers teaching strategies through experience rather than following pre-programmed rules. The agent models the tutoring process as a Markov Decision Process (MDP), dynamically choosing when to introduce, review, or assess cultural concepts based on learner mastery and forgetting dynamics. We implement tabular SARSA with state aggregation as the primary method and systematically explore how aggregation schemes affect learning efficiency and policy quality. PedagoReLearn is evaluated against fixed-curriculum and random baselines using steps-to-mastery, cumulative reward, and quiz accuracy. Our initial target is to show that a SARSA agent with basic aggregation outperforms a fixed curriculum on a three-rule German etiquette domain. Stretch goals include comprehensive analysis of multiple aggregation schemes and addition of a heuristic baseline. Overall, the project illustrates how reinforcement learning can autonomously discover pedagogical strategies that align with Dewey's learner-centered vision while addressing practical issues of state-space complexity in educational domains.**

I. Team Members

The project team consists of Thomas F. Hallmark (UIN 902 009 173) from the Department of Teaching, Learning, & Culture and Jun Kwon (UIN 237007680) from the Department of Computer Science.

II. Proposed Starting Point

A. Codebase Foundation

We build on an existing German language learning program that, by Week 4 of development, already supports vocabulary training with basic quiz mechanics. During the current week of the semester (Week 7), we extend this program to include cultural competence rules and to track mastery, recency, and forgetting dynamics.

B. Implementation Stack

The implementation uses Python 3.x with Gymnasium for environment design, NumPy for numerical computations, Matplotlib for visualization, and PyYAML for configuration. The work is self-contained: there are no specialized hardware needs or external datasets, and simulated student behavior is generated via probabilistic models. The codebase is organized for clarity and reproducibility with separate modules for the environment implementation (Gymnasium-compliant), agent implementations (SARSA and baselines), utilities (aggregation methods, logging, and visualization), configuration files (YAML), experiment scripts, and results storage and analysis.

C. Required Resources

We do not require any resources beyond standard Python libraries available through pip. No specialized hardware, external APIs, or proprietary datasets are needed for this work.

III. Problem Description and Motivation

A. The Application Domain

Cross-cultural competence (CCC) is critical for military personnel, diplomats, educators, and business professionals who operate across cultural boundaries. Effective participation in new contexts depends on knowing and applying appropriate social norms—how to greet others, how to address authority, and which topics are suitable in conversation. Current approaches rely heavily on static instruction—handbooks of "dos and don'ts," lecture slides, and generic online modules—that cannot adjust to individual needs or to the way memory decays over time. Learners who quickly master some rules but struggle with others receive the same sequence as everyone else. Research in cognitive psychology shows that such one-size-fits-all methods produce inefficient retention and weak transfer [1].

B. Market Demand

Market trends reinforce the need for alternatives: the global adaptive learning market expanded from $2.87B in 2024 to $4.39B in 2025, highlighting growing demand for personalized instruction [9].

C. Why This Problem Is Sequential

The tutoring problem is inherently sequential for several reasons. First, prerequisite dependencies mean that the value of any given action depends on what came before. Quizzing before teaching frustrates learners; teaching without later practice invites forgetting; reviewing too soon wastes time, while reviewing too late allows knowledge to fade. Second, learning progresses cumulatively from unfamiliar to mastered, typically over multiple interactions, and the rewards associated with full mastery are delayed. Third, memory decays over time following Ebbinghaus forgetting curves [6], so the optimal timing of review depends on how long ago material was taught. Finally, the learner's true state is latent; we only observe noisy quiz outcomes, meaning the agent must infer mastery levels from behavioral evidence. These characteristics—temporal dependence, delayed rewards, and partial observability—make reinforcement learning an appropriate formalism.

D. Theoretical Grounding: Dewey's Progressive Education

Dewey's progressive education [3], [4] situates this formalism in a broader philosophy: instruction should be a continuing reconstruction of experience, not a rigid script. Rather than programming a fixed sequence, PedagoReLearn learns from the consequences of its choices. If learned policies naturally exhibit spaced repetition, retrieval practice, and adaptive sequencing without those behaviors being explicitly hand-coded, they would provide computational support for Dewey's claim that good pedagogy emerges from the structure of learning itself.

IV. MDP Formulation

We frame tutoring as a Markov Decision Process (S, A, P, R, γ) where the agent must learn an optimal policy $\pi^* : S \to A$ that maximizes expected cumulative discounted reward.

A. State Space Definition

Each state $s \in S$ encodes mastery levels, recency counters, and global context. For mastery levels, each of N cultural rules has a mastery level $m_i \in \{0, 1, 2, 3\}$ where 0 represents unknown (never taught), 1 represents exposed (taught once), 2 represents familiar (multiple exposures with some success), and 3 represents mastered (consistent success). For recency tracking, each rule has a counter $t_i \in \{0, 1, 2, 3, 4, 5\}$ indicating timesteps since last interaction, where 0 indicates just interacted and 5 represents maximum staleness. Additionally, a global step count tracks episode progress. For $N = 3$ rules, the state space contains $4^3 \times 6^3 = 13{,}824$ distinct states—large enough that aggregation is advantageous while remaining tractable for tabular methods. The complete state representation is given by $s = (m_1, m_2, m_3, t_1, t_2, t_3, \text{step\_count})$.

B. Action Space Definition

The action set A captures core pedagogical moves through four action types. The $\texttt{teach}(i)$ action presents rule i with explanation and examples, increases mastery level probabilistically, and resets recency to $t_i \leftarrow 0$. The $\texttt{quiz}(i)$ action assesses the learner on rule i, where success probability depends on mastery, provides a reward signal, and resets recency to $t_i \leftarrow 0$. The $\texttt{review}(i)$ action offers a brief reminder of rule i without full instruction, prevents forgetting, occasionally increases mastery, and resets recency to $t_i \leftarrow 0$. Finally, the $\texttt{no-op}$ action skips the timestep, is used for pacing and ablation studies, and advances all recency counters. For $N = 3$ rules, the action space size is $|A| = 3 \times 3 + 1 = 10$ actions.

C. Transition Function

Transitions $P(s'|s, a)$ are stochastic, implemented through a probabilistic student model inspired by Bayesian Knowledge Tracing [2] and Ebbinghaus-style forgetting [6]. Teaching dynamics are characterized by mastery-level-dependent success probabilities. When teaching a rule with current mastery m_i, the probability of increasing to $m_i + 1$ is given by

$$P(m_i \to m_i + 1|\texttt{teach}(i), m_i) = \begin{cases} 0.9 & \text{if } m_i = 0 \\ 0.6 & \text{if } m_i = 1 \\ 0.4 & \text{if } m_i = 2 \\ 0 & \text{if } m_i = 3 \end{cases} \quad (1)$$

Quiz dynamics depend on current mastery level, with success probability determined by

$$P(\text{success}|m_i) = \{0.1, 0.4, 0.7, 0.95\} \text{ for } m_i \in \{0, 1, 2, 3\} \quad (2)$$

Successful quizzes may increase mastery with probability $P(m_i \to m_i + 1|\text{success}) = 0.3$, while failed quizzes may decrease mastery with probability $P(m_i \to m_i - 1|\text{failure}) = 0.2$. Review dynamics reset recency and provide a small probability of mastery increase with $P(m_i \to m_i + 1) = 0.2$.

Forgetting dynamics govern knowledge decay over time. Each timestep where rule i is not interacted with causes t_i to increase by 1. When $t_i \geq 4$, the probability of mastery decrease is given by

$$P(m_i \to m_i - 1) = \begin{cases} 0.15 & \text{if } m_i \in \{1, 2\} \\ 0.05 & \text{if } m_i = 3 \end{cases} \quad (3)$$

This captures the increased resistance to forgetting at higher mastery levels.

D. Reward Function

Rewards balance effectiveness and efficiency across multiple dimensions. Quiz outcomes provide the primary learning signal, with correct quizzes at mastery level 3 receiving reward $r(\texttt{quiz}(i), \text{correct}, m_i = 3) = +10$, correct quizzes at lower mastery receiving $r(\texttt{quiz}(i), \text{correct}, m_i < 3) = +5$, and incorrect quizzes incurring penalty $r(\texttt{quiz}(i), \text{incorrect}) = -2$. An efficiency incentive adds a per-step penalty $r(\text{any action}) = -0.1$ to encourage compact teaching sequences. A terminal reward $r(\text{all rules mastered}) = +50$ provides strong positive signal for achieving the learning objective. Teaching and review actions receive no immediate reward; their value emerges through enabling future quiz success.

E. Discount Factor

We set the discount factor to $\gamma = 0.95$ to balance short-term and long-term rewards, ensuring the agent values long-term mastery achievement while retaining sensitivity to near-term feedback.

F. Episode Termination

Episodes terminate under three conditions: when all rules reach mastery level 3 (success), when maximum timesteps of 100 steps are reached, or when the agent takes no-op three consecutive times (indicating it has given up).

V. Algorithmic Approach

A. Primary Method: SARSA(0)

Our primary method is on-policy temporal-difference learning with SARSA(0). The Q-value update rule is given by

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma Q(s', a') - Q(s, a)] \quad (4)$$

where (s, a, r, s', a') is the experience tuple, s' is the next state, and a' is the action actually taken following the current policy. SARSA is a natural choice for educational settings because it evaluates the policy actually executed during learning, leading to safer exploration than off-policy methods like Q-learning. The agent learns from actions it actually takes during instruction, mirroring how human teachers adapt based on real pedagogical choices rather than hypothetical optimal ones.

B. Exploration Strategy

We use ϵ-greedy exploration where ϵ decays linearly from 0.3 to 0.05 over 400 episodes. With probability ϵ, the agent selects a random action, while with probability $1-\epsilon$, it selects $\arg\max_a Q(s, a)$. This schedule balances early exploration with later exploitation as the agent gains experience.

C. Hyperparameters

The learning rate is initialized at $\alpha = 0.1$ and may decay to 0.01 in later episodes. The discount factor is set to $\gamma = 0.95$. We use optimistic initialization with $Q(s, a) = 10$ for all state-action pairs to encourage exploration. Training consists of 500 episodes with each episode limited to 100 timesteps.

D. State Aggregation Schemes

To improve data efficiency, we compare several state-aggregation schemes implemented via a mapping function $\phi : S \to S_{\text{agg}}$ that maps full states to aggregate indices so that multiple concrete states share Q-values and generalize from one another.

Scheme 1 serves as our no-aggregation control, using the full state representation with $|S_{\text{agg}}| = 13{,}824$ states. Scheme 2 focuses on mastery by grouping states according to total mastery $\sum_i m_i$ and coarse recency bins (recent: 0-1, moderate: 2-3, stale: 4-5), mapping to approximately 180 aggregate states. Scheme 3 focuses on recency by preserving individual mastery levels while binning recency into three categories per rule, mapping to approximately 432 aggregate states. Scheme 4 provides a hybrid approach that balances both dimensions by grouping rules into mastery categories of unknown, exposed, or familiar/mastered with coarse recency bins, mapping to approximately 300 aggregate states.

E. Research Questions

We investigate four central questions through our experiments. First, does SARSA outperform static curricula in terms of learning efficiency and effectiveness? Second, how does state aggregation affect both convergence speed and asymptotic performance? Third, which features—mastery versus recency—are most predictive for optimal action selection? Fourth, do pedagogical patterns such as spaced repetition and retrieval practice emerge spontaneously without explicit programming?

VI. Baselines and Evaluation

A. Baseline Policies

We evaluate against three baseline policies. The random policy (mandatory) selects actions uniformly at random, providing a lower bound on performance and verifying that improvements arise from learning rather than chance. The fixed curriculum (mandatory) embodies current practice by teaching rule 1 three times, then teaching rule 2 three times, then teaching rule 3 three times, followed by quizzing each rule twice and reviewing any failed rules. The heuristic spaced repetition policy (stretch goal) implements a priority-based approach that, when time permits, teaches unknown rules ($m_i = 0$), reviews stale rules ($t_i > 3$), and quizzes familiar rules ($m_i = 2$) in that order.

B. Evaluation Metrics

Performance is measured through three primary metrics. Steps to mastery measures the average number of timesteps until all rules reach $m_i = 3$, capturing efficiency. Cumulative reward measures the total reward per episode, directly reflecting the learned objective function. Quiz accuracy measures the percentage of quiz attempts answered correctly, indicating teaching effectiveness. Secondary metrics include learning curves plotted across episodes, mastery distributions at episode termination, and action-type frequencies to illuminate the learned policy's character.

C. Experimental Protocol

Each configuration is trained for 500 episodes with 10 different random seeds to ensure statistical validity. Greedy-policy evaluation with $\epsilon = 0$ is conducted every 50 episodes over 20 test episodes. Statistical analysis reports mean performance with standard deviations, constructs 95% confidence intervals using the t-distribution, performs paired t-tests with Bonferroni correction for multiple comparisons, and calculates Cohen's d to distinguish statistical significance from practical significance.

VII. Simplest Possible First Result

Our minimum viable deliverable ensures a complete, graded-ready project structured across four development stages and final documentation.

A. Environment (Week 7-8)

The environment phase delivers a Gymnasium-compliant environment with 3 German cultural rules, correct implementation of state tracking, action processing, forgetting dynamics, rewards, and termination conditions, unit tests validating all transitions, and verification that a random policy runs without errors.

B. SARSA Agent (Week 8-9)

The agent phase implements a working SARSA algorithm with ϵ-greedy exploration, trains over 500 episodes, and logs rewards, states, and actions for analysis.

C. Aggregation and Baselines (Week 10)

The comparison phase implements a single hybrid aggregation scheme, compares aggregated versus non-aggregated SARSA, and establishes both random and fixed-curriculum baselines.

D. Results (Week 11-12)

The results phase generates learning curves showing improvement over training, produces summary tables with mean and standard deviation across 10 seeds, performs statistical tests comparing SARSA to baselines, and demonstrates that the learned policy beats the fixed curriculum.

E. Documentation (Week 13-14)

The documentation phase delivers complete code with clear reproduction instructions, a final report documenting methods and results, and a public repository with a requirements file for dependency management.

VIII. Stretch Goals

If core work completes ahead of schedule, we extend analysis in several explicitly optional directions. First, we complete the full aggregation ablation across all four schemes to quantify which state features are essential. Second, we implement the heuristic spaced-repetition baseline for stronger comparison. Third, we conduct sensitivity analysis studying the impact of learning rate α, discount factor γ, and ϵ-decay schedule on performance. Fourth, we create policy visualizations through state-action heatmaps to aid interpretability. Finally, we scale the domain from three to four rules to examine whether aggregation becomes more critical as state space grows.

IX. Implementation Timeline

We are currently at Week 7 of the semester and Week 4 of coding, with cultural aspects being integrated this week. Approximately 8 weeks remain until the project deadline. Week 7 focuses on finishing the environment with all dynamics working correctly. Week 8 implements the SARSA agent from scratch. Week 9 scales to the full N=3 rule environment with debugging and tuning. Week 10 integrates baselines and aggregation while running preliminary comparisons. Week 11 executes the full experimental suite across all seeds. Week 12 performs analysis and visualization of results. Weeks 13-14 are devoted to writing, polishing, and final packaging of all deliverables.

Our buffer plan governs scope reductions if delays occur. Environment or learning difficulties trigger permanent reduction to N=2 rules. Experimental delays lead to freezing aggregation at a single hybrid scheme. Runtime issues prompt reduction from 10 to 5 random seeds. Time pressure focuses analysis exclusively on core comparisons.

X. Reproducibility

All experiments are configured via YAML files that specify hyperparameters, random seeds, and student-model settings. A single command launches each configuration, allowing complete reproducibility. Results are logged to CSV files with sufficient detail to regenerate all plots and tables. The repository includes a requirements file listing all dependencies, installation instructions for setting up the environment, exact run commands for reproducing experiments, and clear descriptions of the directory layout. Version control through Git tracks the full project trajectory from initial prototypes through final experiments.

XI. Related Work

Intelligent tutoring systems (ITS) have demonstrated that adaptive methods can approach human tutoring effectiveness [10], but many rely on hand-crafted pedagogical rules derived from expert knowledge rather than learned policies. Deep Knowledge Tracing [7] uses recurrent neural networks to predict learner knowledge states from interaction histories but does not address the action selection problem of determining what to teach next. Reinforcement learning approaches in education have optimized specific subproblems such as adaptive test termination [8] and problem difficulty selection [5], but rarely address the full instructional sequencing problem that must balance introducing new material, reinforcing existing knowledge, and assessing mastery under forgetting dynamics.

Our contribution targets this gap by treating the complete teaching sequence as a sequential decision problem. We add a systematic investigation of state aggregation—an established RL technique that remains under-explored in educational domains—to understand which state features are most critical for pedagogical decision-making. The student model grounds our work in Bayesian Knowledge Tracing [2] and Ebbinghaus's forgetting curve [6], while the reward design reflects empirical evidence from spaced-repetition research [1].

XII. Risk Mitigation and Scope Management

A. Technical Risks

Technical risks include slow convergence, excessive variance, and environment bugs. We mitigate slow convergence by starting with an N=2 rule environment to validate that learning occurs before scaling up. Environment bugs are addressed through comprehensive unit tests and scripted policies that verify transitions match specifications. If SARSA underperforms despite tuning efforts, we maintain Q-learning as a fallback option while attempting to preserve the on-policy framing that better matches educational contexts.

B. Timeline Risks

Timeline risks are managed by strict adherence to the minimum viable deliverable defined in Section VII. We defer all stretch goals until core results are secure and stable. The buffer plan provides clear decision points for reducing complexity without compromising the study's integrity. This disciplined approach maximizes the probability of delivering a robust, well-documented project on schedule.

XIII. Expected Contributions

A. Practical Contributions

From a practical standpoint, PedagoReLearn demonstrates an RL-based pipeline for adaptive tutoring that discovers effective policies without requiring hand-coded pedagogical rules. The system provides a reusable Gymnasium environment for further research in educational RL applications, offering a concrete foundation for future work in this domain.

B. Theoretical Contributions

Theoretically, this work operationalizes Dewey's progressive education philosophy [3], [4], demonstrating how experiential interaction with learners yields effective instructional strategies. If learned policies naturally adopt patterns such as spaced repetition or retrieval practice without explicit programming, this suggests that such pedagogical strategies represent optimal solutions to the sequential teaching problem rather than arbitrary conventions, providing computational support for Dewey's educational philosophy.

C. Educational Contributions

From an educational perspective, PedagoReLearn points toward scalable personalization for CCC training where human-like adaptivity can be approximated at scale. This addresses a critical need for adaptive instruction in military, diplomatic, and business contexts where cultural competence has tangible consequences for mission success and relationship building.

XIV. Conclusion

PedagoReLearn addresses adaptive cross-cultural training by casting instruction as a sequential decision problem where the agent learns when to teach, review, and assess based on mastery and forgetting dynamics. Our focus on tabular SARSA with state aggregation balances technical ambition with practical feasibility: the approach is technically substantive, pedagogically meaningful, and achievable within a semester timeline. The project structure clearly distinguishes core deliverables from optional extensions, with explicit timeline buffers and safeguards built into the schedule. Ultimately, this work links reinforcement learning methodology to progressive educational theory and offers a concrete, reproducible pathway for building adaptive tutoring systems that are both theoretically principled and practically implementable.

References

[1] N. J. Cepeda, H. Pashler, E. Vul, J. T. Wixted, and D. Rohrer, "Distributed practice in verbal recall tasks: A review and quantitative synthesis," *Psychological Bulletin*, vol. 132, no. 3, pp. 354–380, 2006.

[2] A. T. Corbett and J. R. Anderson, "Knowledge tracing: Modeling the acquisition of procedural knowledge," *User Modeling and User-Adapted Interaction*, vol. 4, no. 4, pp. 253–278, 1995.

[3] J. Dewey, *Democracy and Education*. New York: Macmillan, 1916.

[4] J. Dewey, *Experience and Education*. New York: Macmillan, 1938.

[5] Z. Liu, K. R. Koedinger, and E. A. McLaughlin, "Leveraging problem difficulty in the selection of instructional practice opportunities," in *Proc. 13th Int. Conf. Educational Data Mining*, 2020.

[6] J. M. J. Murre and J. Dros, "Replication and analysis of Ebbinghaus' forgetting curve," *PLOS ONE*, vol. 10, no. 7, p. e0120644, 2015.

[7] C. Piech et al., "Deep knowledge tracing," in *Advances in Neural Information Processing Systems*, vol. 28, pp. 505–513, 2015.

[8] A. N. Rafferty, E. Brunskill, T. L. Griffiths, and P. Shafto, "Faster teaching via POMDP planning," *Cognitive Science*, vol. 40, no. 6, pp. 1290–1332, 2016.

[9] Research and Markets, "Adaptive learning market by component, deployment, and end-user—Global forecast to 2025," 2024.

[10] K. VanLehn, "The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems," *Educational Psychologist*, vol. 46, no. 4, pp. 197–221, 2011.