LATTE: Learning Attendance Timetables through Tailored Experiences

Group 2
Lou Zhanzhi (A0288570R, e1313679@u.nus.edu)
Da Jiaxuan (A0287922R, e1297764@u.nus.edu)
Lu Haomeng (A0281439X, e1155567@u.nus.edu)
National University of Singapore

Abstract

Most existing timetabling systems focus narrowly on producing conflict-free schedules, overlooking the behavioural factors that influence actual student attendance. We propose LATTE, an interpretable AI framework that generates university timetables aligned with individual student profiles. LATTE combines constraint satisfaction and heuristic-guided beam search with reinforcement learning from human feedback (RLHF), modeling diverse student personas (e.g., Lazy, Hardworking). Using GPT-4o to simulate pairwise preferences, we train a neural reward model to guide policy learning within a custom OpenAI Gym environment. LATTE outputs realistic attendance trajectories tailored to different behavioural profiles. Experiments show that LATTE effectively captures nuanced scheduling preferences across personas, offering a modular, extensible tool for behaviour-aware educational planning. The full implementation of our system is available at `https://github.com/zzzlou/CS3263`.

1 Introduction

University students frequently face challenges when constructing weekly timetables. While existing scheduling tools prevent module clashes, they rarely account for lifestyle or behavioural preferences—such as avoiding early classes, preserving lunch breaks, or reserving days for rest or part-time work [1]. As a result, students often receive technically feasible timetables that do not align with their actual habits. For example, a student who typically avoids 8 AM classes may still be assigned one, leading to poor attendance and academic underperformance. Such misalignments contribute to disengagement, chronic fatigue, and absenteeism, ultimately undermining academic outcomes and well-being [2].

Traditional personalised timetabling systems rely on explicit preference inputs (e.g., surveys or manual selections). However, these methods overlook a crucial insight: **students often struggle to articulate their true preferences quantitatively**, and their stated intentions may not reflect true behaviours [3]. Attendance decisions are influenced by dynamic factors such as daily fatigue, academic stress, or time-of-day preferences. For instance, a student might skip a non-compulsory afternoon session after a tiring morning, or endure a morning lecture if it is critical for GPA. Modelling such behavioural trade-offs demands learning from observed attendance patterns rather than relying solely on declared preferences [4].

To bridge this gap, we introduce LATTE (**L**earning **A**ttendance **T**imetables through **T**ailored **E**xperiences), an AI-based system that generates and refines schedules based on real attendance behaviour and learned human preferences. LATTE first constructs feasible schedule candidates using constraint satisfaction and heuristic-guided search [5] to ensure constraint adherence and alignment with broad preferences. It then models day-to-day attendance as a Markov Decision Process (MDP) [6], where each session represents a decision point balancing comfort and long-term academic impact.

We adopt Reinforcement Learning from Human Feedback (RLHF) to optimise attendance decisions. Instead of relying on hard-coded rewards, LATTE trains a neural reward model on pairwise preferences—e.g., "Schedule A is preferred over Schedule B"—simulated using GPT-4o. This model captures nuanced behavioural trade-offs such as avoiding overload or managing fatigue, and guides Q-learning to produce realistic, persona-aligned attendance trajectories. These trajectories span a range of student types (e.g., ”lazy” to ”hardworking”), enhancing both schedule quality and adherence.

2 Related Work

Our work builds upon and integrates techniques from multiple research areas, including heuristic search algorithms, scheduling optimisation, preference-based reinforcement learning, and Q-learning methods. In this section, we briefly review relevant work from each area.

2.1 Search Algorithms

Heuristic search algorithms have been extensively studied for solving complex combinatorial optimisation problems. Classical methods such as beam search and simulated annealing have shown considerable success in various domains, including resource allocation, course timetabling, and scheduling problems [7]. Beam search, for instance, selectively explores promising branches, balancing exploration and efficiency, while simulated annealing provides stochastic optimisation that can escape local minima by probabilistically accepting suboptimal solutions at earlier stages [8]. Random restarts further enhance solution quality by diversifying initial search conditions [9].

2.2 Scheduling Problem

The course timetabling problem is a well-known NP-complete optimisation challenge extensively investigated in operations research and artificial intelligence [1]. Existing work often formulates this problem as a Constraint Satisfaction Problem (CSP), utilising techniques like Arc Consistency algorithms (AC-3) to prune infeasible schedules efficiently [10; 5]. Recent approaches have incorporated various heuristic optimisation methods, including genetic algorithms [11], particle swarm optimisation [12], and hybrid search methods to improve scheduling performance and adherence to specific user-defined constraints and preferences [13].

However, traditional scheduling systems typically consider only explicit constraints (e.g., avoiding time clashes) and fail to capture implicit human preferences comprehensively. Few studies have explored how to leverage implicit behavioural data and reinforcement learning frameworks to improve personalised scheduling outcomes.

2.3 Preference-Based Reinforcement Learning and RLHF

Preference-based Reinforcement Learning (PbRL) addresses scenarios where explicit reward functions are difficult to define but human preferences can be provided as comparative feedback [3]. Christiano et al. [4] pioneered the concept of Reinforcement Learning from Human Feedback (RLHF), demonstrating its effectiveness for training complex policies by collecting human preferences on trajectory pairs and training neural reward models accordingly. This method has become a cornerstone in the training and fine-tuning of modern large language models (LLMs) such as the GPT series, significantly improving their alignment with human values and intents [14].

Applications of RLHF and PbRL extend beyond language modelling to robotics, recommendation systems, and personalised assistants, demonstrating their effectiveness in modelling nuanced human preferences and implicit trade-offs [15; 16]. Our work leverages these methodologies explicitly to distil human-aligned attendance policies, thereby significantly advancing the personalisation capabilities of timetable generation systems.

2.4 Q-learning

Q-learning, introduced by Watkins and Dayan [17], is one of the foundational reinforcement learning algorithms, particularly well-suited for discrete state and action spaces due to its simplicity and guaranteed convergence under certain conditions. In Q-learning, the optimal policy is learned directly from experience using iterative updates of Q-values, which estimate the long-term value of state-action pairs.

Despite its simplicity, Q-learning remains a strong baseline and is frequently used as a benchmark and foundational method across in many RL applications, such as resource allocation, scheduling, and pathfinding [6]. Recent work has extended Q-learning to scenarios that involve human feedback, effectively combining Q-learning with learned reward functions to improve the alignment of policy with human preferences [18; 19].

3 Problem Formulation

We formally define the personalised timetabling problem as follows. Given a set of courses $C = \{c_1, c_2, ..., c_n\}$ that a student selects for a particular semester, and a set of feasible weekly schedules S satisfying basic scheduling constraints (e.g., no time clashes), our goal is twofold:

1. Generate a feasible weekly schedule $s \in S$ that matches a student's general lifestyle and broad-stroke preferences.
2. Predict a realistic attendance trajectory A over the weekly schedule s, where each class session has an attendance decision (attend or skip).

Specifically, for the first goal, we define the problem as follows: given a set of courses $C = \{c_1, c_2, ..., c_n\}$, where each course c_i has multiple possible permutations of lectures and tutorials (e.g., different combinations of lecture and tutorial times), we first apply Constraint Satisfaction Problem (CSP) techniques, specifically the AC3 algorithm, to eliminate strictly infeasible permutations (e.g., permutations with direct time clashes). After obtaining a reduced set of feasible permutations, we perform heuristic-guided search to find an optimal permutation combination that best matches predefined, heuristic-based preference criteria. The resulting solution can be represented as a schedule $s = \{p_1, p_2, ..., p_n\}$, where each p_i is an index indicating the selected permutation for the course c_i.

Given the s obtained from the previous step, we model the attendance decision-making process on each single day as a Markov Decision Process (MDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$:

- $\mathcal{S}$ is the state space representing the student's context, characterised by variables such as GPA, fatigue level, day of the week, class start time, and whether the class is compulsory.
- $\mathcal{A}$ is the action space $\{\text{attend}, \text{skip}\}$.
- P represents state transition probabilities that capture how attending or skipping classes affects future states (e.g., GPA and fatigue changes).
- R is the reward function learned from human preference feedback, reflecting the subjective value students assign to different attendance outcomes and daily schedules.
- $\gamma \in [0, 1]$ is the discount factor balancing immediate and future rewards.

Our objective is to find a policy $\pi : \mathcal{S} \to \mathcal{A}$ that maximises the expected cumulative reward:

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t R(s_t, a_t)\right]$$

By solving this MDP using reinforcement learning, LATTE generates attendance policies and trajectories closely aligned with realistic student behaviour and preferences.

4 Methodology

4.1 Search

4.1.1 Search Initialisation and Strategy

The purpose of the search is to find a suitable starting state for reinforcement learning in order to accelerate convergence. This is based on the assumption that the user has successfully chosen a persona (heuristic weight combination) that aligns with their behaviour at this stage.

Even after AC-3 filtering, the state space remains large. For instance, a typical Year 1 Semester 1 CS student's course selection (CS1101S, CS1231S, MA1521, MA1522, IS1108), including invalid states, results in a total of 195,979,680 possible states. As such, exhaustive search is computationally infeasible. We therefore adopt local search methods to efficiently identify a high-quality initial state.

We explore several variants of beam search, including combinations with simulated annealing and random restarts, to avoid local optima. The experiment is twofold: first, we use a simple heuristic (sum of permutation indices) to compare the effectiveness of the following strategies:

- Beam search only
- Beam search with simulated annealing ($T = 100$, $T = 300$)
- Beam search with simulated annealing and 10 random starts ($T = 300$)

The maximum theoretical sum of permutation indices in our test configuration is 366.

Second, we assess how the number of random starts affects both solution quality and computational time, under the assumption that runtime grows linearly with the number of starts.

4.1.2 *Personalised Search Heuristics*

Prior to reinforcement learning, we define a rule-based scoring function that simulates student personas. This function evaluates each candidate timetable using interpretable lifestyle-based criteria without relying on any environment simulation.

We define four behavioural metrics:

- **Lunch Day Count:** Number of weekdays preserving a lunch break window (10AM–2PM)
- **Free Day Count:** Number of weekdays with no scheduled classes
- **Early Morning Count:** Number of days with classes at or before 9AM
- **Exhaustive Day Count:** Number of days with six or more scheduled hours

These metrics are linearly weighted based on the profile type (e.g., *Lazy*, *Hardworking*, *EarlyBird*, *Chill*) to produce a preference score. This scoring function provides interpretable guidance during search and serves as a prior before reward learning.

4.2 Attendance Decision as MDP

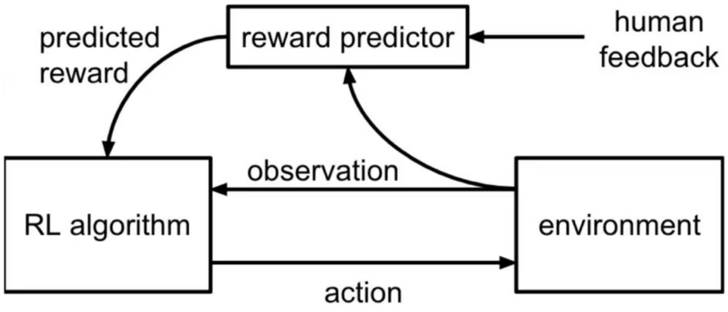


Figure 1: Overall RLHF-based attendance modeling pipeline.

In this stage, we model student attendance decisions in each single day as a Markov Decision Process (MDP), treating each class session as an independent decision point.

The MDP is defined by $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where:

- $\mathcal{S}$ is the state space defined by GPA level, fatigue level, weekday, class timing, and compulsory status.
- $\mathcal{A}$ is the action space $\{\text{attend}, \text{skip}\}$.
- $P(s'|s, a)$ captures the effect of an action (e.g., GPA drop from skipping compulsory class, fatigue increase from attending an early lecture).
- R is a reward function learned from human preferences, quantifying the desirability of each state-action outcome.

- γ is the discount factor (set to 0.95) balancing short-term comfort and long-term academic performance.

The goal is to learn an optimal attendance policy π^* that maximizes the expected cumulative reward:

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t R(s_t, a_t)\right]$$

However, defining a reward function that accurately reflects human preferences presents a major challenge. Rather than manually specifying numeric reward values, which is unnatural for subjective judgments, we adopt a Reinforcement Learning from Human Feedback (RLHF) framework to learn the reward function based on comparative preference signals.

4.2.1 Human Preference Reward Modeling

Motivation Quantifying human preferences directly with absolute scores is notoriously difficult. Students rarely express preferences as ”State A deserves a reward of 0.5 and State B deserves 0.55.” Instead, they naturally express preferences through relative judgments, such as ”I prefer schedule A over schedule B.” To capture this, we model reward learning through pairwise trajectory comparisons.

Trajectory Sampling We first generate diverse attendance trajectories to provide training data for the reward model:

- 30 weekly timetables are generated using GPT-4o simulations.
- For each timetable, we simulate 9 attendance trajectories:
  - 5 random agent trajectories
  - 2 lazy agent trajectories
  - 2 hardworking agent trajectories
- In total, 270 trajectories are sampled.

Preference Labeling For each group of 9 trajectories, we construct 20 pairwise comparisons. We then use GPT-4o to label each pair according to a predefined student persona, such as:

- Lazy persona: dislikes early classes, prefers free Fridays, values lighter schedules.
- Hardworking persona: prioritizes GPA, tolerates heavy schedules if necessary.

Prompts are carefully crafted to align GPT-4o's simulated judgments with these personas, ensuring consistent and persona-aligned preference annotations.

Reward Model Training We train a neural reward model to align cumulative trajectory rewards with human preferences. The model assigns a scalar reward to each individual state, and the total reward for a trajectory is computed as the average of the state rewards along the trajectory.

Training is conducted via supervised learning using binary cross-entropy (BCE) loss over trajectory pair preferences. Specifically, given two trajectories τ_1 and τ_2 with a human preference label $y \in \{0, 1\}$, the loss is defined as:

$$\mathcal{L} = -\left[y \cdot \log \sigma\left(\overline{R}(\tau_1) - \overline{R}(\tau_2)\right) + (1 - y) \cdot \log\left(1 - \sigma\left(\overline{R}(\tau_1) - \overline{R}(\tau_2)\right)\right)\right]$$

where:

- $\overline{R}(\tau) = \frac{1}{|\tau|} \sum_{s \in \tau} r_\theta(s)$ denotes the normalized cumulative reward of trajectory τ.
- $r_\theta(s)$ is the predicted reward for state s by the neural network with parameters θ.
- $\sigma(\cdot)$ is the sigmoid activation function.

This procedure enables us to learn a reward function that robustly captures human-like scheduling preferences, providing a crucial foundation for subsequent policy optimization.

4.2.2 *Policy Optimization via Q-learning*

With the human-aligned reward model in place, we proceed to optimize attendance policies through Q-learning:

- We implement a custom OpenAI Gym environment simulating attendance decision-making.
- Agents interact with the environment using an epsilon-greedy strategy, progressively refining Q-values based on reward model feedback.
- Training continues until convergence, typically around 3000 episodes.

In effect, the reward model acts as a soft supervisor, guiding the agent to internalize and imitate human-like scheduling trade-offs—such as balancing fatigue, GPA impact, and comfort preferences. The resulting Q-learning policies successfully distill these human preferences into practical, day-level attendance strategies, producing realistic, personalized trajectories that better match students' daily habits and academic priorities.

5 Experiments

In this section, we present the experimental evaluation of LATTE. We focus specifically on the effectiveness of our reward modelling and the subsequent reinforcement-learning-distilled attendance policies.

5.1 Experimental Setup

5.1.1 *Metrics*

To evaluate the performance of our methods comprehensively, we adopt the following metrics:

Reward Model Evaluation:

- **Accuracy (ACC):** Measures the proportion of preference pairs where the reward model agrees with the human-annotated preference.
- **Binomial Test:** Statistical test to determine if model predictions significantly exceed random guessing.

Reinforcement Learning Evaluation:

- **Average Cumulative Reward:** Average reward obtained by the learned attendance policies over multiple simulated attendance trajectories.
- **Policy Stability:** Qualitative and quantitative evaluation of policy convergence across episodes.
- **Qualitative Analysis:** Visual comparisons between attendance trajectories generated by RL-derived policies and baseline heuristic policies.

5.1.2 *Implementation Details*

We implemented LATTE using Python and PyTorch for neural modeling, and OpenAI Gym for simulating the MDP environment. Key implementation details are as follows:

Reward Model Architecture:

- A fully connected feedforward neural network with two hidden layers of sizes 32 and 16 respectively, each followed by ReLU activation and dropout (dropout probability 0.2).
- The output layer produces a scalar reward for each state.
- The model was trained separately on pairwise preference data using binary cross-entropy (BCE) loss and the Adam optimizer.
- The learning rate and other training hyperparameters were tuned externally; in deployment, the trained reward model is loaded and frozen during reinforcement learning.

Environment Setup:

- We designed a custom OpenAI Gym environment simulating daily attendance decisions.
- The state space includes discrete features: GPA level, fatigue level, weekday, whether the upcoming class is required, and the start time bucket of the next class.
- Transitions model the real-world effects of attending or skipping classes, affecting both GPA and fatigue dynamically.

Reinforcement Learning Details:

- We employed standard tabular Q-learning with an epsilon-greedy exploration strategy.
- Learning rate α was set to 0.1, discount factor γ to 0.99, and exploration rate ϵ to 0.2.
- Training was run for up to 500 episodes per day schedule, with early convergence detected if the maximum Q-value update between episodes fell below $\theta = 10^{-3}$.

Computational Environment:

- Experiments were conducted locally on a MacBook Air 15″ (M3, 2024), using the CPU backend for both simulation and model inference.

5.2 Main Results

5.2.1 Search

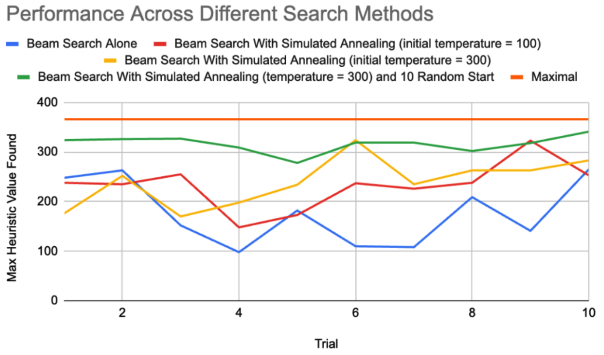


Figure 2: Performance Across Different Search Methods

The baseline, Beam Search Alone, exhibited the weakest performance with fluctuations between 100 and 250. Notable degradation in trials 4 through 6 suggests frequent entrapment in local optima. Beam Search with Simulated Annealing (temperature 100) yielded modest improvements. Raising the temperature to 300 enhanced search diversity, improving average scores.

Beam Search with Simulated Annealing (T=300) and 10 Random Starts was the most effective, consistently achieving heuristic values above 300. Its robustness stems from diversified search paths and improved avoidance of suboptimal regions.

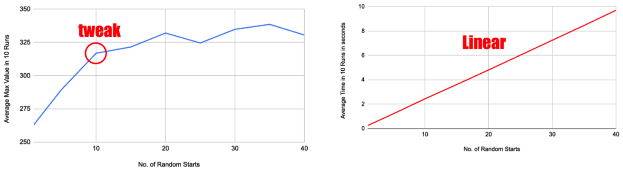


Figure 3: Performance-time Balance for Different Number of Restarts

To evaluate the impact of random restarts, we varied the number of starts while fixing other parameters (beam size 10, cooling rate 0.95, temperature 300, iterations 1000). We found diminishing returns beyond 10–20 restarts, with performance plateauing but runtime increasing linearly.

These findings confirm that while randomisation enhances robustness, excessive starts add computational overhead without proportional benefit. 10–20 restarts strike a balance between quality and efficiency.

5.2.2 *Reward Model Performance*

We evaluated the reward model on both GPT-4o-simulated and manually labelled preference datasets:

- **Accuracy:** 92% agreement with human-labeled preference pairs (50 samples).
- **Binomial Test:** p-value < 0.01, confirming performance exceeds random guessing.

The supervised training curve of the reward model is provided in Appendix 5, showing convergence after a few epochs. Qualitative inspection further revealed that the model consistently assigned higher rewards to ”lazy”-aligned schedules — those with fewer early sessions, shorter total daily hours, and avoidance of Friday classes — aligning well with intended human preferences.

5.2.3 *Personalised Timetable Distillation via RL*

To assess the effectiveness of RL-based policy learning, we compared Q-learning-derived policies against three baselines under the trained Lazy-aligned reward model:

Policy	**Average Reward**	**Std Deviation**
Random Policy	4.9	0.6
Rule-based Lazy Policy	7.1	0.4
Rule-based Hardworking Policy	6.5	0.3
RL Policy (Q-learning)	10.2	0.5

Table 1: Performance comparison of policies evaluated under the Lazy-aligned reward model. Results are averaged over 20 independent rollouts.

The Q-learning agent achieved the highest average cumulative reward, outperforming both random and rule-based policies. **It should be noted that due to environmental stochasticity and reward modeling noise, direct numeric comparisons between policies may carry inherent variance**; the results primarily serve as qualitative indicators of improvement rather than definitive superiority.

5.2.4 *Trajectory Visualization and Behavioral Alignment*

Figure 4 visualizes the effect of preference-aligned distillation on a sample weekly timetable:

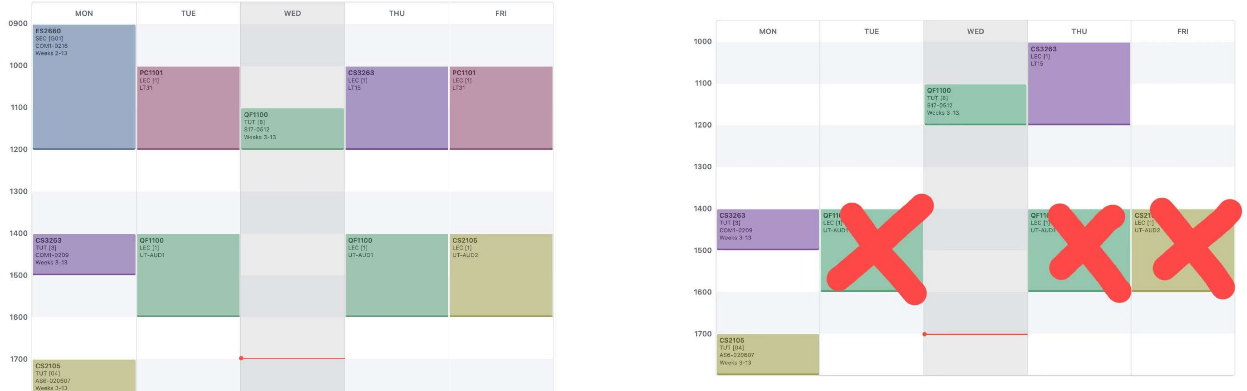
Trajectory analysis confirmed that LATTE successfully captured persona-aligned attendance behaviors:

- **Lazy persona:** The distilled timetable exhibits clear preference-driven adaptations. Early morning sessions (e.g., before 10 AM) are consistently skipped, only essential classes (affecting GPA) are selectively attended, and Friday sessions are entirely avoided, reflecting a strong preference for a lighter weekly structure.

To further validate behavioral consistency, we conducted an aggregated time-of-day analysis. Specifically, we executed the Lazy-aligned policy over 30 simulated weekly timetables, recording the timestamps of attended classes while controlling for whether attendance was mandatory.

The resulting attendance heatmap, presented in Appendix 6, reveals a strong bias toward later-day classes and minimal attendance during early morning hours, further confirming the model’s alignment with intended time-based preferences.

These results demonstrate that LATTE reliably produces feasible, preference-aligned schedules across diverse student profiles, offering a promising framework for human-centric educational planning.



(a) Original full timetable (preference-agnostic)

(b) Distilled timetable after preference-aligned modeling

Figure 4: Comparison between the original and distilled timetables under the Lazy preference profile.

It is worth noting that, given certain modeling simplifications—such as discretized state variables and reward learning from simulated preferences—the distilled plans should be interpreted as behavioral guidelines rather than strict prescriptions. Students are encouraged to adapt the recommendations based on real-world constraints and personal judgment.

6 Discussion

6.1 Limitations

While LATTE demonstrates the promise of using AI methods for personalised student timetabling, several limitations constrain its practical effectiveness and generalisability:

Heuristic Validity: The heuristic-guided search used in the initial scheduling step depends heavily on manually defined heuristics. These, while intuitive, have not been rigorously validated with real-world student data. As a result, the generated schedules may not fully reflect genuine preferences or broader behavioural trends.

Simplified Environment and Discrete State Representation: To support efficient Q-learning, we simplified the environment and used discretised representations (e.g., GPA and fatigue as 'level 1', 'level 2', or 'level 3'). Transition probabilities are approximations rather than empirically derived. **Hence, attendance plans should be viewed as illustrative guidance rather than definitive schedules.**

Stochasticity and Policy Comparison Challenges: The stochastic nature of RL rollouts introduces variability, making it challenging to compare learned attendance policies reliably. Small variations in execution can produce different trajectories, complicating performance assessment.

Lack of Robustness and Real-World Data: Due to the lack of real human-generated preference data, the reward model relies solely on GPT-simulated feedback. The model currently supports only two simplified personas ("lazy" and "hardworking"), limiting robustness and applicability across more diverse behavioural types. Without broader validation, its generalisability remains uncertain.

6.2 Future Work

To address current limitations and enhance LATTE's utility, we propose several future directions:

Validation and Refinement of Heuristics: Empirical validation of heuristics through large-scale student surveys and real scheduling data can calibrate the initial timetable generation process.

Continuous State Representations with PPO: Adopting continuous state-space representations and advanced RL methods like Proximal Policy Optimisation (PPO) can yield more expressive, nuanced behaviour modelling.

Real-World Data Collection and Human-in-the-loop Validation: Incorporating real student feedback and attendance logs into training would enhance model realism. A human-in-the-loop framework could iteratively refine both heuristic and learned components.

Extension to Diverse Personas: Expanding the persona set (e.g., ”evening-focused”, ”balanced”, ”stress-sensitive”) would improve adaptability and practical relevance.

By advancing these directions, LATTE can evolve into a robust, reliable, and widely applicable tool for AI-assisted educational planning.

7 Conclusion

In this work, we introduced LATTE, a personalised timetabling system leveraging artificial intelligence techniques to generate weekly schedules that align with individual student lifestyles and attendance preferences. Our system uniquely integrates constraint satisfaction, heuristic-guided search, and reinforcement learning, specifically utilising Reinforcement Learning from Human Feedback (RLHF) to realistically model and optimise student attendance behaviours.

By formulating attendance decisions as a Markov Decision Process and training a neural reward model from simulated human preferences, we successfully captured complex trade-offs inherent in students’ daily decision-making—balancing factors such as GPA maintenance, fatigue management, and personal comfort. Our results demonstrated that personalised timetables generated through this process meaningfully reflect distinct student personas, offering practical and realistic scheduling recommendations.

Nevertheless, LATTE faces several limitations, including reliance on manually defined heuristics, simplified discrete state representations, stochasticity in policy evaluation, and limited robustness due to simulated rather than real human data.

Ultimately, this research highlights a promising pathway towards more effective, personalised educational tools, blending AI-driven methodologies with authentic human behaviour modelling to significantly improve student engagement, academic performance, and overall satisfaction.

Miscellaneous

Teamwork and Contribution Breakdown

Lu Haomeng developed the core scheduling logic. He implemented the constraint satisfaction module based on the AC-3 algorithm, constructed the feasibility-checking system, and engineered the beam search strategy for efficient timetable candidate generation. He was also responsible for integrating external data from the NUSMods API and ensuring end-to-end consistency within the scheduling component.

Da Jiaxuan designed and implemented the personalisation framework. She formulated rule-based reward functions to simulate diverse student behaviours and curated the reward model’s training data by generating pairwise trajectory preferences through GPT-4o. Her work ensured that the reinforcement learning pipeline was grounded in interpretable, human-aligned behavioural labels.

Lou Zhanzhi led the reinforcement learning module. He implemented the Q-learning algorithm and constructed a neural reward model using the annotated preference data. In addition, he designed the reward shaping strategy, built the custom OpenAI Gym environment, and oversaw policy evaluation to ensure that learned attendance behaviours aligned with distinct student personas.

Acknowledgements

We would like to thank the OpenAI GPT-4o model for assisting in the refinement and polishing of the written content. Its support helped enhance the clarity and readability of this report.

References

[1] Burke EK, Petrovic S. Recent research directions in automated timetabling. European Journal of Operational Research. 2002;140(2):266-80. Available from: https://www.sciencedirect.com/science/

article/pii/S0377221702000693.

[2] Kassarnig V, Bjerre-Nielsen A, Mones E, Lehmann S, Lassen DD. Class attendance, peer similarity, and academic performance in a large field study. PLOS ONE. 2017 Nov;12(11):e0187078. Available from: http://dx.doi.org/10.1371/journal.pone.0187078.

[3] Wirth C, Akrour R, Neumann G, Fürnkranz J. A survey of preference-based reinforcement learning methods. J Mach Learn Res. 2017 Jan;18(1):4945–4990.

[4] Christiano P, Leike J, Brown TB, Martic M, Legg S, Amodei D. Deep reinforcement learning from human preferences; 2023. Available from: https://arxiv.org/abs/1706.03741.

[5] Dechter R. Constraint Processing. San Francisco, CA: Morgan Kaufmann; 2003.

[6] Sutton RS, Barto AG. Reinforcement Learning: An Introduction. 2nd ed. Cambridge, MA: MIT Press; 2018.

[7] Russell S, Norvig P. Artificial Intelligence: A Modern Approach. Pearson Education; 2010.

[8] Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by Simulated Annealing. Science. 1983;220(4598):671-80. Available from: https://www.science.org/doi/abs/10.1126/science.220.4598.671.

[9] Gomes CP, Selman B, Kautz H. Boosting combinatorial search through randomization. In: Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence. AAAI '98/IAAI '98. USA: American Association for Artificial Intelligence; 1998. p. 431–437.

[10] Mackworth AK. Consistency in networks of relations. Artificial Intelligence. 1977;8(1):99-118. Available from: https://www.sciencedirect.com/science/article/pii/0004370277900078.

[11] Lewis R. A survey of metaheuristic-based techniques for University Timetabling problems. OR Spectrum. 2008;30(1):167-90. Available from: https://doi.org/10.1007/s00291-007-0097-0.

[12] Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of ICNN'95 - International Conference on Neural Networks. vol. 4; 1995. p. 1942-8 vol.4.

[13] Burke EK, McCollum B, Meisels A, Petrovic S, Qu R. A graph-based hyper-heuristic for educational timetabling problems. European Journal of Operational Research. 2007;176(1):177-92. Available from: https://www.sciencedirect.com/science/article/pii/S0377221705006387.

[14] Ouyang L, Wu J, Jiang X, Almeida D, Wainwright CL, Mishkin P, et al.. Training language models to follow instructions with human feedback; 2022. Available from: https://arxiv.org/abs/2203.02155.

[15] Lee K, Smith L, Abbeel P. PEBBLE: Feedback-Efficient Interactive Reinforcement Learning via Relabeling Experience and Unsupervised Pre-training; 2021. Available from: https://arxiv.org/abs/2106.05091.

[16] Ziegler DM, Stiennon N, Wu J, Brown TB, Radford A, Amodei D, et al.. Fine-Tuning Language Models from Human Preferences; 2020. Available from: https://arxiv.org/abs/1909.08593.

[17] Watkins CJ, Dayan P. Q-learning. In: Machine Learning. vol. 8; 1992. p. 279-92.

[18] Knox WB, Stone P. Interactively shaping agents via human reinforcement: the TAMER framework. In: Proceedings of the Fifth International Conference on Knowledge Capture. K-CAP '09. New York, NY, USA: Association for Computing Machinery; 2009. p. 9–16. Available from: https://doi.org/10.1145/1597735.1597738.

[19] Griffith S, Subramanian K, Scholz J, Isbell CL, Thomaz A. Policy shaping: integrating human feedback with reinforcement learning. In: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2. NIPS'13. Red Hook, NY, USA: Curran Associates Inc.; 2013. p. 2625–2633.

Appendix A: Additional Figures

Reward Model Training Curve

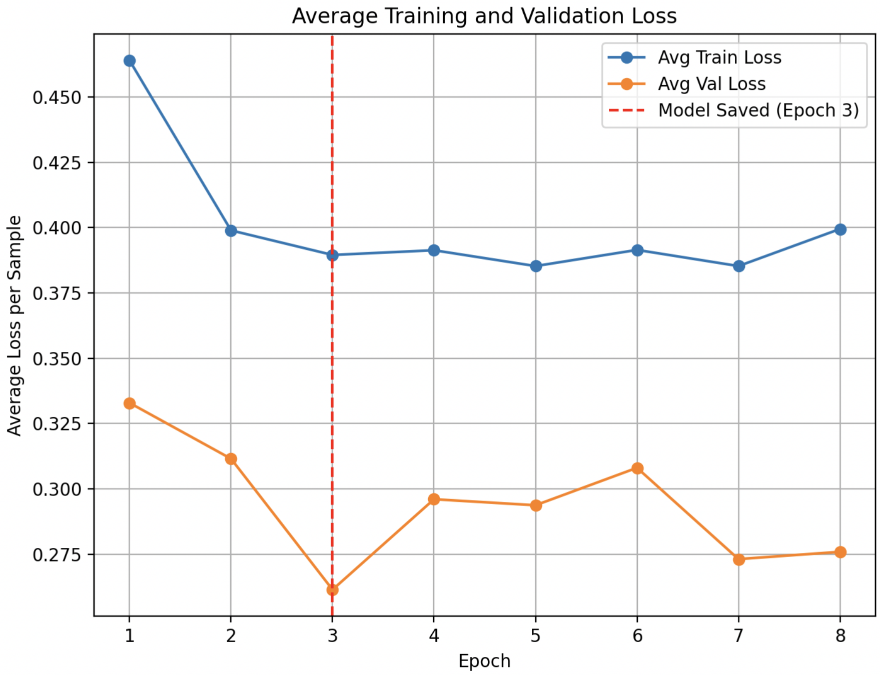


Figure 5: Training curve of the reward model. Binary cross-entropy loss converges to approximately 0.26 after a few epochs.

Aggregated Attendance Heatmap

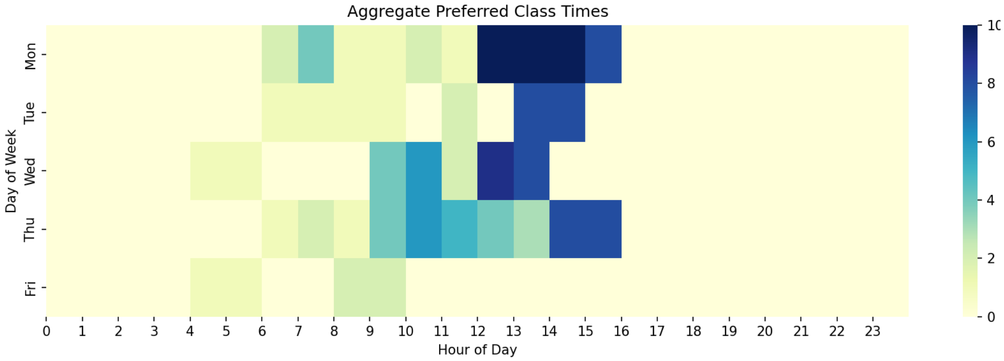


Figure 6: Aggregated attendance heatmap for the Lazy persona. Darker regions indicate higher likelihood of class attendance across time slots. Early morning sessions are sparsely attended, while mid-day and afternoon slots are preferred.

Appendix B: Prompts for Preference Simulation

To simulate human preference labels for training the reward model, we designed structured prompts provided to GPT-4o. Below we document the exact system and user prompts used for the **Lazy** persona simulation.

System Prompt

> You are a university student making daily class attendance decisions. You strongly dislike early morning classes (especially 8-9 AM) and generally prefer a lighter, more relaxed schedule. You always avoid Friday classes regardless of whether attendance is compulsory. You dislike long days with excessive total class hours, as they cause fatigue. However, you care about your GPA and will attend important classes if the academic penalty from skipping (e.g., GPA falling below level 2) is too severe. Ignore course codes — base your decisions purely on timing, structure, and academic impact.
> Example trajectory showing incremental schedule build-up:
>
> ```
> schedule trajectory: [[], [], [], [('CSE4016', (11, 12), True)], [('CSE4016', (11, 12), True)]]
> day: 1, gpa: 1
> ```
>
> Each entry like `(’CSE4016’, (11, 12), True)` indicates a course (start-end time) and whether attendance is mandatory (True).

User Prompt

> Below are two class schedule evolution trajectories. Each represents a day’s sequence of attendance decisions.
> Please evaluate them based on your preferences:
> **Trajectory A:**
>
> `{result1}`
>
> **Trajectory B:**
>
> `{result2}`
>
> Your preferences to keep in mind:
>
> - Strong dislike for early classes (especially 8-9 AM).
> - Preference for short, relaxed daily schedules.
> - Strictly avoid Friday classes, regardless of GPA impact.
> - Dislike of long days causing fatigue.
> - GPA matters: if skipping risks GPA falling below 2.0, consider attending.
>
> **Instruction:** Reply with `1` if you prefer Trajectory A, or `0` if you prefer Trajectory B. Output nothing else.