Extended Abstract

Motivation Emergency room triage remains a significant issue in the healthcare system, with many hospitals struggling with overcrowding, triaging errors, and lack of resources. A study by Kaiser Permanente found that "mistriage occurred in nearly 32.2% of 5 million encounters" Chang et al. (2022) in hospitals they observed. Due to these inefficiencies, wait times have become longer, worsening patient outcomes over time. In the US, the average wait time before treatment starts is nearly 27 minutes, demonstrating the critical need for improvements. This research project aims to investigate how reinforcement learning can assist ER nurses in making more informed decisions to help reduce errors in patient triage and improve operational efficiency and patient outcomes.

Method We developed a custom environment using Gymnasium to model the key dynamics of a basic ER department in hospitals. The environment includes three key classes for patients, hospital staff, and representing the emergency room. During the environment initialization, we randomly initialize between 3 to 7 hospital staff and generate 50 new patients with varying age, gender, and pain level. During the initial timestep, all patients are triaged and enter the waiting queue. At each step, the RL agent chooses a patient ID in the waiting queue to be admitted for treatment. Once admitted, patients are treated for the estimated timesteps and later discharged.

Based on the environment dynamics, we developed a dense reward function on a scale between [0, 1]. The agent was evaluated at each step based on the weighted average of the patient's wait time and severity level. Additionally, the agent was penalized if beds and staff members were idle when queued patients needed them. For this setting, we benchmarked Implicit Q-learning against behavior cloning and modified its objectives to achieve the highest performance. For IQL, we implemented two variants: one that relies exclusively on offline data for policy learning and another incorporating environment interaction to enable online data collection.

Implementation After implementing both agents and IQL variants, hyperparameter optimization was necessary, given the development of a custom environment and expert data. The expert data was manually created using a heuristic learned from the following dataset, "Patient Flow and Triage Simulation" Mahato (2023). The BC and IQL agents were tested for 700 and 2,000 iterations, respectively. We utilized the g4dn.xlarge spot instance on AWS, which consists of 1 NVIDIA T4 GPU with 16 GB of GPU memory and four vCPUs.

Results The BC agent provided a solid benchmark to improve upon using the IQL agent. The BC agent had a average return around 0.45 with a standard deviation of 0.15 across the 700 timesteps. The average wait time for the BC agent was 26.02 timesteps (minutes) with staff and bed utilization remaining low around 81& and 34%, respectively. The offline variant of IQL performed highest among the agents with an average reward of 0.52 and a standard deviation of 0.17. Additionally, it had the lowest average wait time for patients at 22.71 timesteps, saving 4 minutes from BC agent. Additionally, staff and beds were utilized efficiently at 97% and 92%, respectively.

Discussion The environment-specific metrics revealed that the results for the BC agent were similar to those of ER departments in the US. The average wait time was around 26.02 minutes, approximately the national average. Furthermore, the bed and staff utilization was low due to the expert having limited information regarding the current state of the emergency room department. By leveraging the observation space containing the entire state of the ER, the IQL agent could take more informed actions when prioritizing patient treatment. his advantage is reflected in the plots, where the bed and staff utilization peaked at approximately 97% and 92%, respectively. Additionally, the agent reduced the wait time by nearly 4 minutes, demonstrating its ability to improve patient outcomes without delaying necessary care.

Conclusion This paper aims to solve these issues by presenting a novel method to integrate reinforcement learning into ER flow. After creating a custom ER environment and testing offline RL agents, we demonstrate the potential for RL agents to make dynamic, context-aware triage decisions that consider patient acuity and real-time resource availability. While challenges remain in terms of interpretability and privacy, this works a step towards augmenting triage decision-making, leading to better patient outcomes.

A Novel Approach Using Implicit Q-Learning to Optimize ER Patient Triage

Aadhav Prabu
Department of Computer Science
Stanford University
aprabu2@stanford.edu

Abstract

Today, emergency room (ER) nurses in hospitals face significant triage challenges due to the increasing number of patients, limited resources, and critical time-sensitive decisions. They must be able to quickly rank patients based on their symptom type, severity level, and treatment specialty within the constraints of limited resources, overcrowding, and inadequate information at intake. A study by Kaiser Permanente found that "mistriage occurred in nearly 32.2% of 5 million encounters" Chang et al. (2022) in hospitals they observed. This research project aims to investigate how reinforcement learning can assist ER nurses in making more informed decisions when prioritizing patients in the emergency room. We created a custom gym environment to simulate ER dynamics, such as triaging, admitting, and discharging patients, based on the current state of the ER. Algorithmically, we implemented a Behavior Cloning (BC) agent and a variant of Implicit Q-learning (IQL) agent to provide optimal suggestions for ER nurses during triage. Our results indicate that the IQL agent improved from the baseline set by the BC agent. The IQL agent received a higher reward of 0.52 compared to 0.45 for the BC agent while lowering the average patient wait by four timesteps. Additionally, it was more efficient in using the available resources of staff and beds by nearly 17% and 58%, respectively. This project addresses a critical gap in emergency care: reducing errors in patient triage and helping improve operational efficiency and patient outcomes.

1 Introduction

Emergency room triage remains a significant issue in the healthcare system, with many hospitals struggling with overcrowding, triaging errors, and lack of resources. Due to these inefficiencies, wait times have become longer, worsening patient outcomes over time. While crowding and resource limitations have strained ER departments, the biggest issue can be linked to triage errors. ER nurses face intense pressure to make time-sensitive, high-stakes decisions that can have life-or-death consequences for many people. Given the rise of AI in clinical workflows, reinforcement learning presents a promising approach to optimizing the decision-making process in ER triage.

1.1 Emergency Room Triage

Before analyzing the issue with ER triage, it is vital to understand its mechanics. A typical ER flow can be categorized into three phases: Input, Throughput, and Output Samadbeik et al. (2024). The input phase starts with a patient's arrival via a walk-in or ambulance. According to the CDC, there were nearly 139.8 million arrivals in the past year Centers for Disease Control and Prevention (2025). During the throughput phase, a clinician will screen the patient and order labs or imaging for further diagnosis. Based on the diagnosis, a patient is assigned a score based on the Emergency

Severity Index (ESI). Accordingly, the patient is assigned to one of the following treatment zones: resuscitation, fast-track, or main ED. After treatment, the patient is either discharged or transferred to a different department.

A critical challenge in the emergency room triage flow is misclassifying patients into the wrong treatment zones. Overcrowding forces ER nurses to make quick decisions regarding the patient's ESI and their position within the queue. Errors in triage caused by underestimating pain severity or delaying care can lead to harmful patient outcomes Guttmann et al. (2011). Compounding this issue, nurses must also account for dynamic hospital constraints, including the real-time and projected availability of hospital resources, such as the number of available beds, doctors, and labs. Balancing urgency and resource limitations makes ER triaging an inherently complex process.

1.2 AI in Medicine

With the rise of generative AI, more healthcare organizations are integrating AI into their clinical workflows. EHR companies, such as Epic Systems and Oracle Cerner, are utilizing AI to reduce the administrative burden on doctors. Currently, over 1,000 AI health systems have been approved by the FDA, indicating the rise of AI in medicine Smith (2025). New systems must be built with strong ethical foundations, prioritizing trust, transparency, and accountability. Given the sensitivity of patient health data, privacy and algorithmic bias must be adequately addressed.

2 Related Work

2.1 Supervised Learning Approaches

Much of the current literature emphasizes using supervised learning to enhance the triaging process in emergency rooms. These approaches involve aggregating patient data to classify them based on the Emergency Severity Index (ESI) without considering the ER department's current situation, including available resources and staff to treat patients. As such, the main goal of these works is to augment nurses' decision-making when initially classifying each patient's level of need.

A recent study focused on enhancing nurses' subjective assessments when evaluating patients for their ESI. The researchers implemented recurrent neural networks and attention mechanisms trained on patient medical records and achieved an accuracy rate of 87% across nearly 118,000 patients. Yao et al. (2021) A similar study by Beth Israel Medical Center evaluated various supervised learning approaches, including gradient tree boosting and a two-layer neural network, on structured patient data, reaching 80% classification success, on average Goodwin et al. (2024). While these studies highlight the improvement of evaluating ER patients during triage, these approaches fail to consider the current state of the ER, which can significantly influence patient outcomes in the emergency setting.

2.2 Reinforcement Learning Approaches

One particular paper by Babylon Health in the UK presented a different approach from the limited hard-coded decision trees. Using 1,374 clinical vignettes representing about three triage decisions by doctors, the researchers trained a Deep Q-learning system to effectively classify patients based on limited patient information. Buchard et al. (2020) At each step, the agent would determine if it is appropriate to make a triage decision at the current state or if a follow-up question should be asked. While this approach performed similarly to earlier supervised methods, it was better at adapting to unseen cases by asking additional questions—something standard supervised learning methods cannot do.

Another approach used a partially observable Markov Decision Process to provide real-time information support and direction for patients in remote locations. Thapa et al. (2005) Doctors would automatically be notified if the patient's wearable triggered an alert and would assign the patient to the nearest location with the available doctor and resources to treat the patient. Existing literature does not adequately explore the application of reinforcement learning in medical triage. While the two above-mentioned papers utilized reinforcement learning to optimize medical triage, their findings are limited and not sufficiently encompassing for real-world situations, offering little practical value. This research aims to address these shortcomings.

3 Method

3.1 Custom ER Environment

Existing literature explores this problem by classifying patients according to the Emergency Severity Index (ESI) using supervised learning. One paper utilizes RL to effectively rank patients using a limited number of questions or information. Our novel approach lies in simulating a resource-constrained ER triage as a Markov Decision Process and utilizing RL agents to effectively rank patients under uncertainty. We will integrate patient information from the "Patient Flow and Triage Simulation" Mahato (2023) dataset from Kaggle to accurately model environment dynamics. We developed a custom environment using Gymnasium to model the key dynamics of a basic ER department in hospitals. The environment includes three key classes: Patient, HospitalStaff, and EmergencyRoom. Each class contains specific attributes designed to replicate essential components of ER operations. This modular design enables better experimentation with different environment dynamics, such as staffing policies and patient flows.

3.1.1 Patients

Each patient was identified with a unique ID and health metadata, including their age, gender, arrival time, and pain level. For this environment, we assumed all patients were created at the initial timestep and placed into the category of not triaged. Additionally, we assumed that all non-triaged patients were immediately triaged in the next timestep and were diagnosed with a severity level, estimated treatment time, and whether they required a bed (overnight stay). The diagnosed severity level and treatment time were determined based on the patient's age and self-reported pain level. Based on the chosen action of the RL agent, a patient can be admitted and assigned to a staff member to begin treatment. At the end of treatment, the patient is automatically discharged. The wait time of each patient was calculated based on the number of timesteps between triage and admittance.

3.1.2 Hospital Staff

Similar to patients, each staff member was identified with a unique ID. Each member was restricted to a specific capacity and allowed to treat between 1 and 3 patients at a given timestep. In future iterations, each staff member can be restricted from treating particular symptoms based on their specialty and role.

3.1.3 Emergency Room

During the environment initialization, we randomly initialize between 3 to 7 hospital staff and generate 50 new patients with varying age, gender, and pain level. During the initial timestep, all patients are triaged and enter the waiting queue. At each step, the RL agent chooses a patient ID in the waiting queue to be admitted for treatment. We first validate whether this patient ID exists in the queue and determine if any staff can treat the patient. Additionally, patients who require a bed are only admitted if beds are available. Once admitted, patients are treated for the estimated treatment timesteps and later discharged. The iteration is complete after all patients are treated or the number of timesteps exceeds 200.

Based on the environment dynamics, we developed a dense reward function on a scale between $[0, 1]$. The agent was evaluated at each step based on the weighted average of the patient's wait time and severity level. Additionally, the agent was penalized if beds and staff members were idle when queued patients needed them. If the agent selected an invalid action, such as choosing a patient not in the waiting queue, it received a reward of 0. The total cost was normalized to be scaled between $[0, 1]$. Although the theoretical maximum is 1, the agent is expected to be unable to achieve this value, as the queue introduces unavoidable penalties. We also considered developing a more sparse reward function but opted against it as enough expert data was not collected.

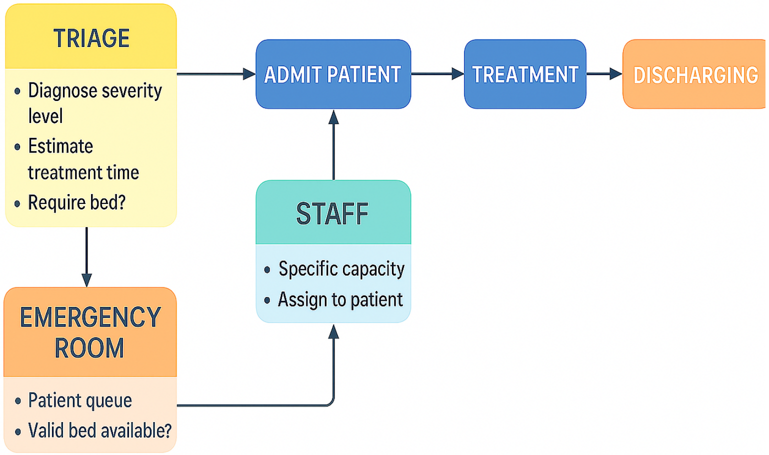


Figure 1: Custom ER Environment Flow

3.2 Reinforcement Learning Algorithms

We implemented Implicit Q-learning as our reinforcement learning agent for training. For this setting, we benchmarked it against behavior cloning and modified its objectives to achieve the highest performance. Both agents will be evaluated using several metrics, such as their ability to reduce the average patient wait time and efficiently use the available hospital staff and beds. Several helper classes were created to create expert data from the downloaded dataset, train the RL agents, and log relevant metrics to tensorboard. We adapted the infrastructure code from the class homework to help implement these classes for a discrete action space.

3.2.1 Behavior Cloning Agent

The behavior cloning (BC) agent serves as a supervised learning baseline by mimicking the actions from the generated expert dataset. Rather than exploring the environment independently, the BC agent attempts to minimize the difference between its predicted actions and those in the dataset. While this approach provides a sufficient baseline, the agent is only as performant as the expert dataset. Therefore, in more complex environments like the created ER environment, we expect that the behavior cloning agent will have respectable performance but will not significantly improve from the current performance of real-world ER departments.

3.2.2 Implicit Q-Learning Agent

Unlike the behavior cloning agent, we expect the Implicit Q-Learning agent to have high performance and reveal opportunities for using RL in the ER department. IQL is an offline reinforcement learning algorithm that is designed to learn effective policies from expert data. However, unlike the BC agent, IQL estimates value functions for each transition. Using this value function, IQL can update its Q-values to create an advantage estimate for (state, action) pairs, allowing it to extract better actions even if they are underrepresented in the data.

To adapt IQL to our environment and allow it to achieve the best performance, we modified the observation space to include information about the patients in the waiting queue and available staff and beds. Additionally, since the action space (the waiting queue) changed during each environment step, we created an action mask within the info variable to restrict the agent from taking invalid actions (choosing patient ID outside the waiting queue). For IQL, we implemented two variants: one that relies exclusively on offline data for policy learning and another incorporating environment interaction to enable online data collection.

4 Experimental Setup

After implementing both agents and IQL variants, hyperparameter optimization was necessary, given the development of a custom environment and expert data. The expert data was manually created using a heuristic learned from the following dataset, "Patient Flow and Triage Simulation" Mahato (2023). The BC and IQL agents were tested for 700 and 2,000 iterations, respectively. We utilized the g4dn.xlarge spot instance on AWS, which consists of 1 NVIDIA T4 GPU with 16 GB of GPU memory and four vCPUs. This setup provided the necessary GPU capabilities to train our lightweight RL models while remaining cost-efficient.

Key Hyperparamter Values

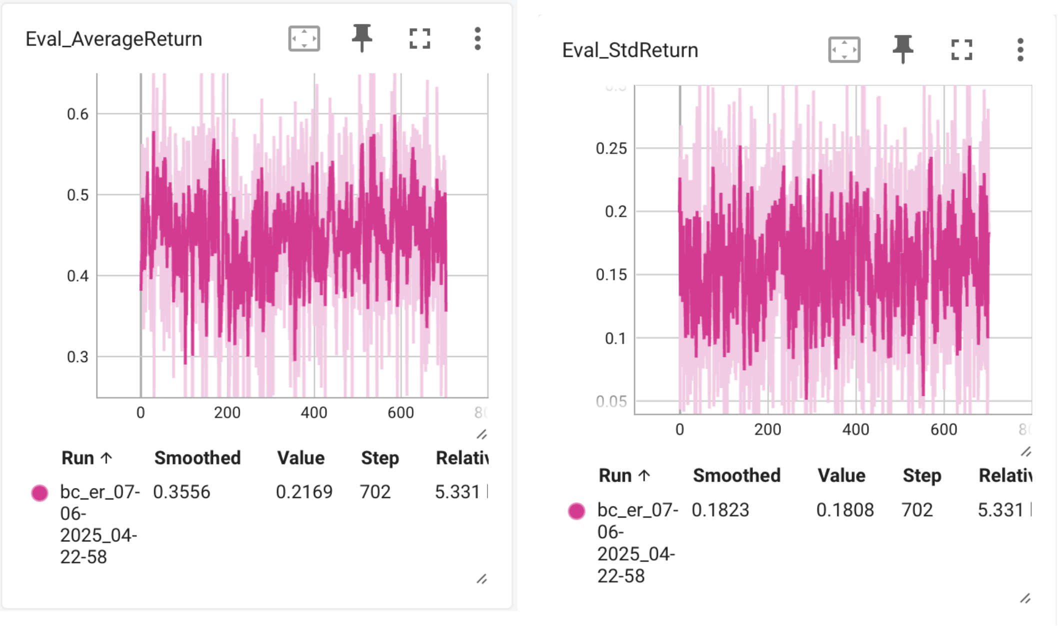
- Train Batch Size: 1024
- Evaluation Batch Size: 256
- AWAC Lambda (IQL): 0.5
- Learning Rate (IQL): 1×10^{-3}
- Expectile (IQL): 0.8

Evaluation Metrics

- Average Return
- Standard Deviation
- Average Patient Wait Time
- Average Severity-Weighted Patient Wait Time
- Bed Utilization (Total beds used / Total beds needed)
- Staff Utilization (Total staff assigned / Total staff capacity)

5 Results

5.1 Behavior Cloning Agent



Eval_AverageReturn

Run ↑	Smoothed	Value	Step	Relati
● bc_er_07-06-2025_04-22-58	0.3556	0.2169	702	5.331

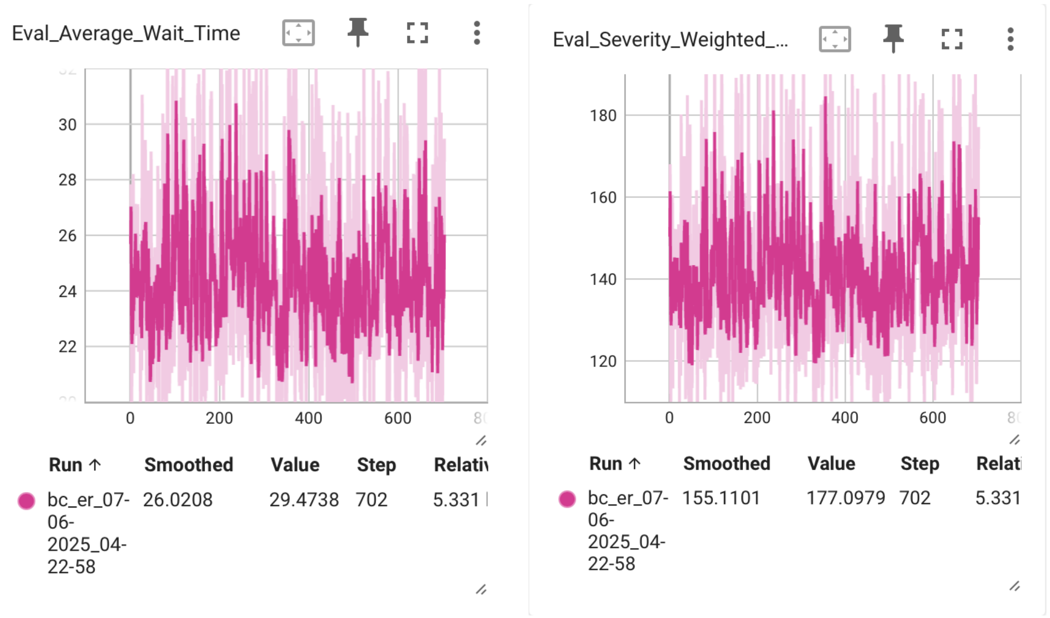
Eval_StdReturn

Run ↑	Smoothed	Value	Step	Relati
● bc_er_07-06-2025_04-22-58	0.1823	0.1808	702	5.331

(a) Average Return

(b) Standard Deviation

Figure 2: The BC agent had a average return around 0.45 with a standard deviation of 0.15 across the 700 timesteps.



Eval_Average_Wait_Time

Run ↑	Smoothed	Value	Step	Relativ
● bc_er_07-06-2025_04-22-58	26.0208	29.4738	702	5.331

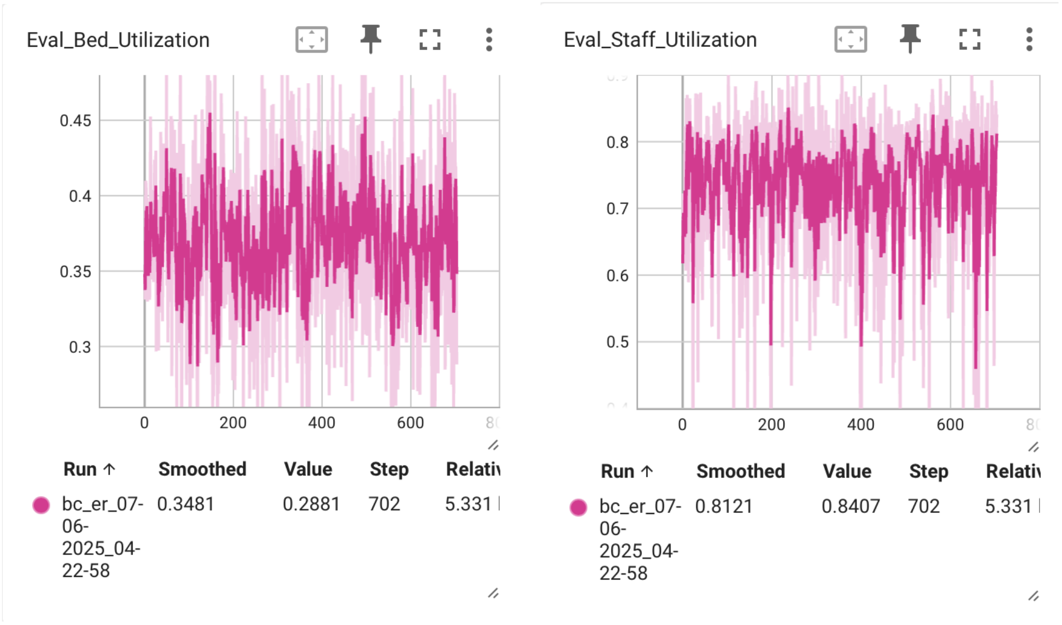
Eval_Severity_Weighted_...

Run ↑	Smoothed	Value	Step	Relat
● bc_er_07-06-2025_04-22-58	155.1101	177.0979	702	5.331

(a) Average Patient Wait Time

(b) Severity-Weighted Patient Wait Time

Figure 3: The average wait time for the BC agent was 26.02 timesteps or similar to the average wait time in the US. The wait time weighted by the patient's severity level was around 150 timesteps.



Eval_Bed_Utilization

Run ↑	Smoothed	Value	Step	Relativ
● bc_er_07-06-2025_04-22-58	0.3481	0.2881	702	5.331

Eval_Staff_Utilization

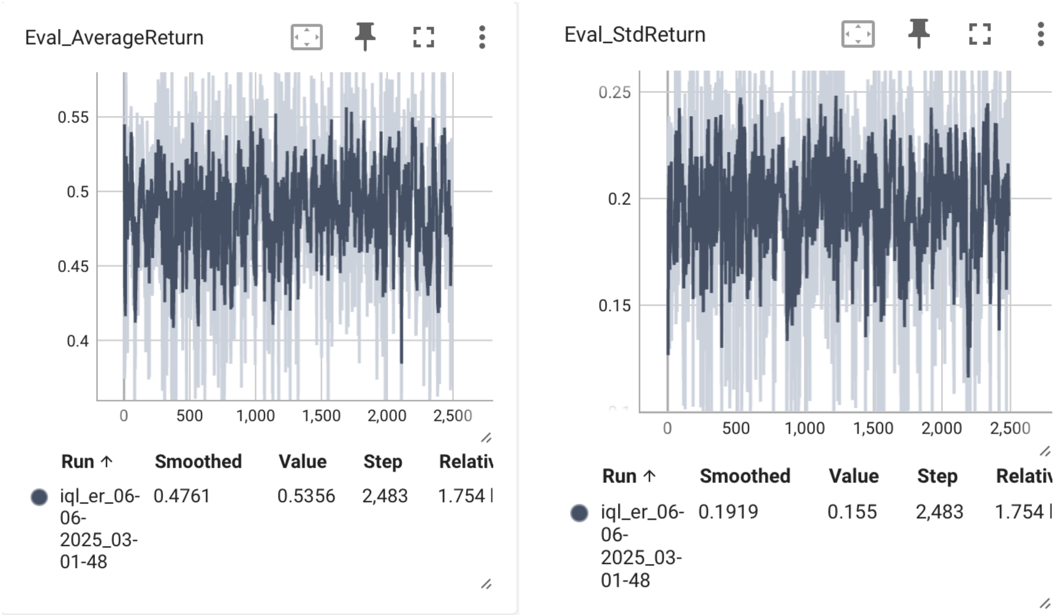
Run ↑	Smoothed	Value	Step	Relativ
● bc_er_07-06-2025_04-22-58	0.8121	0.8407	702	5.331

(a) Bed Utilization

(b) Staff Utilization

Figure 4: The agent had a staff utilization around 0.81, meaning that 19% of the available staff were not assigned any patients. Similarly, the bed utilization was extremely low around 0.34, indicating that many patients that required beds were not given one.

5.2 IQL Agent



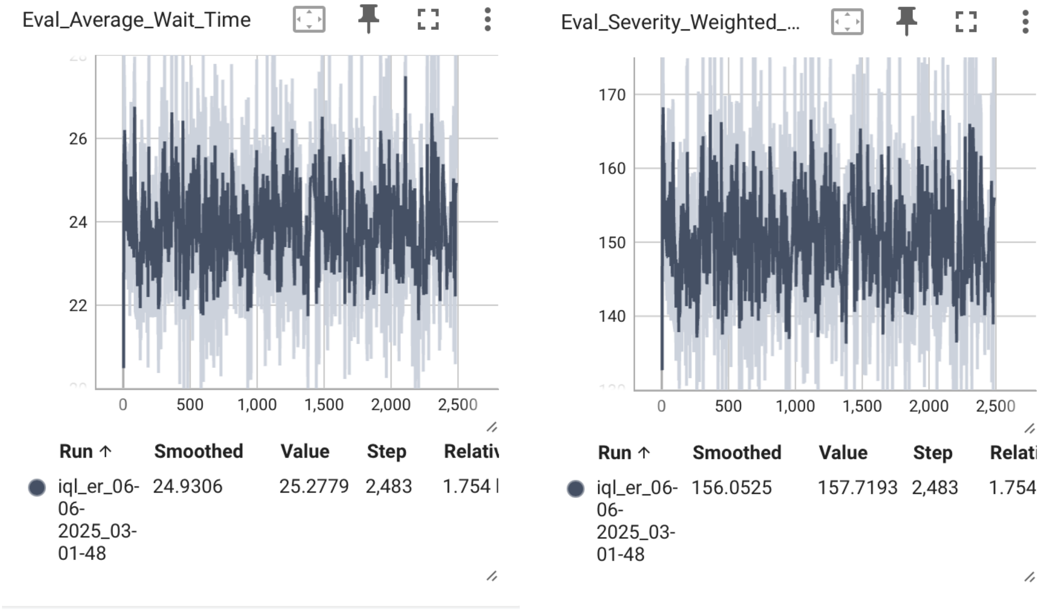
Run ↑	Smoothed	Value	Step	Relativ
● iql_er_06-06-2025_03-01-48	0.4761	0.5356	2,483	1.754 I

Run ↑	Smoothed	Value	Step	Relativ
● iql_er_06-06-2025_03-01-48	0.1919	0.155	2,483	1.754 I

(a) Average Return

(b) Standard Deviation

Figure 5: The IQL agent, which was initially trained on expert data, had an average return of 0.48 with a standard deviation of 0.19 across 2,000 iterations.



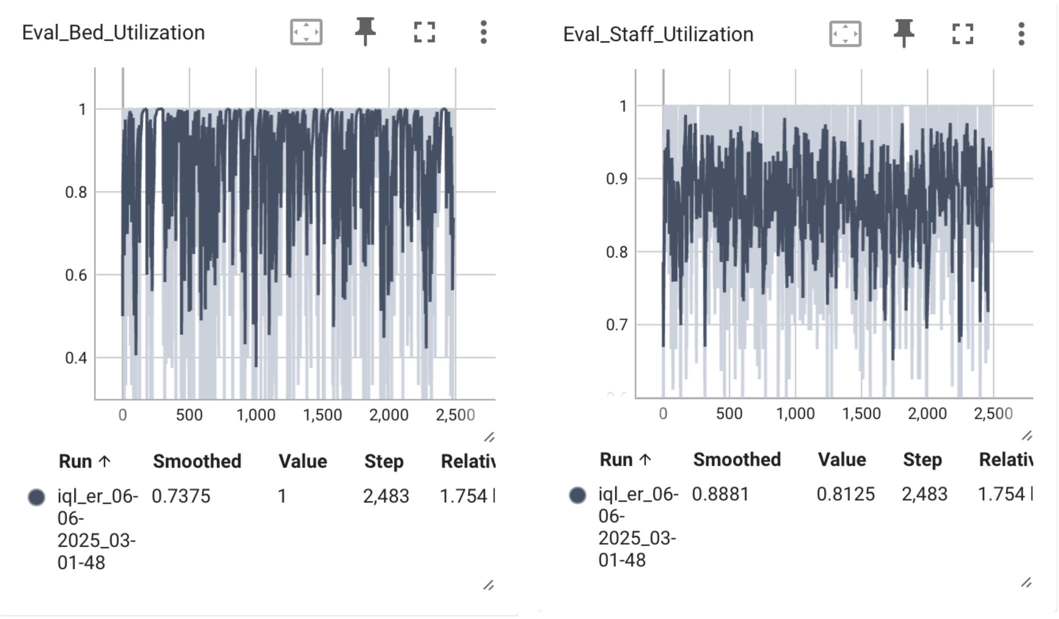
Run ↑	Smoothed	Value	Step	Relativ
● iql_er_06-06-2025_03-01-48	24.9306	25.2779	2,483	1.754 I

Run ↑	Smoothed	Value	Step	Relati
● iql_er_06-06-2025_03-01-48	156.0525	157.7193	2,483	1.754

(a) Average Patient Wait Time

(b) Severity-Weighted Patient Wait Time

Figure 6: The higher average reward indicates that the IQL agent should have a slightly lower patient wait time as shown in the plots with an average patient wait time of 24.93 timesteps across the 2,000 iterations.



Run ↑	Smoothed	Value	Step	Relativ
● iql_er_06-06-2025_03-01-48	0.7375	1	2,483	1.754 l

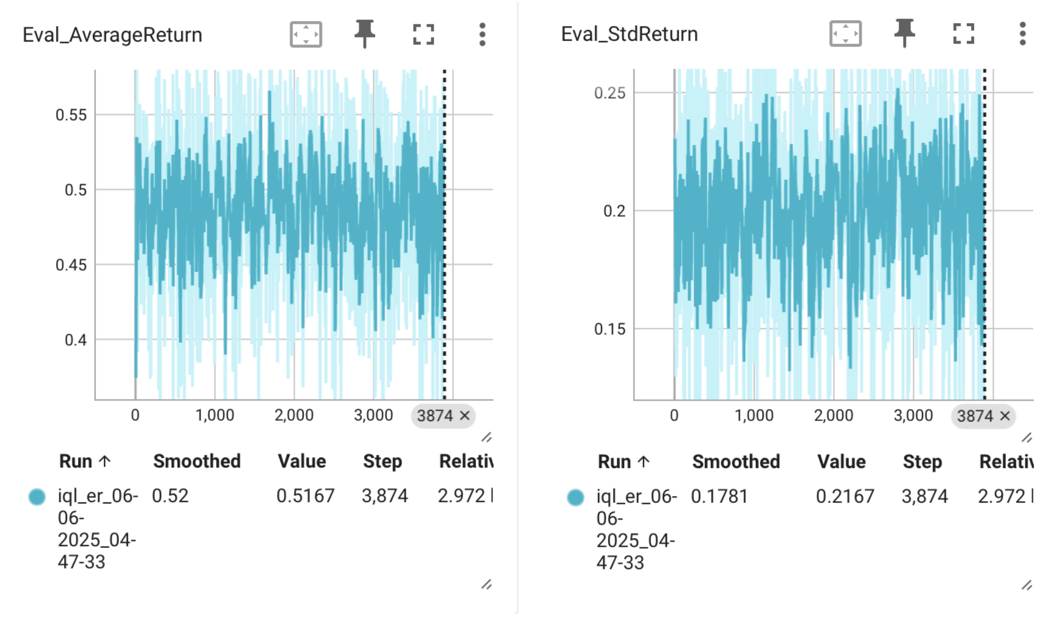
Run ↑	Smoothed	Value	Step	Relativ
● iql_er_06-06-2025_03-01-48	0.8881	0.8125	2,483	1.754 l

(a) Bed Utilization

(b) Staff Utilization

Figure 7: The IQL agent efficiently assigned beds and hospital staff members to patients. Bed utilization was around 74% and staff utilization was close to 88%. This efficient utilization explains the higher rewards received by the agent.

5.3 IQL Variant Agent



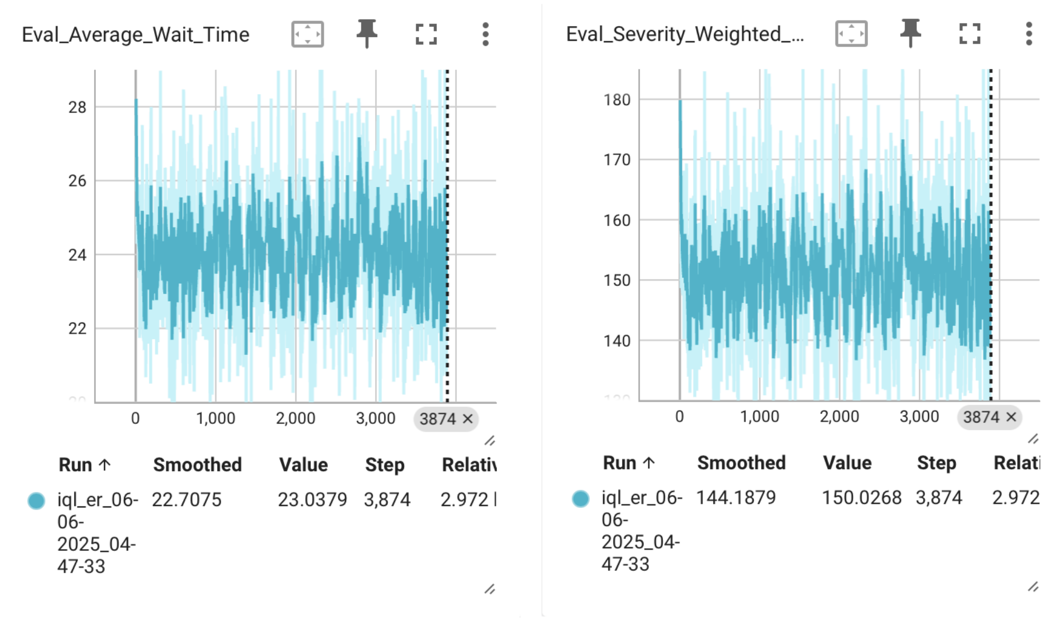
Run ↑	Smoothed	Value	Step	Relativ
● iql_er_06-06-2025_04-47-33	0.52	0.5167	3,874	2.972 l

Run ↑	Smoothed	Value	Step	Relativ
● iql_er_06-06-2025_04-47-33	0.1781	0.2167	3,874	2.972 l

(a) Average Return

(b) Standard Deviation

Figure 8: The offline-exploitation variant of IQL had an average reward of 0.52 with a standard deviation of 0.17, which is the best performance across all of the agents.



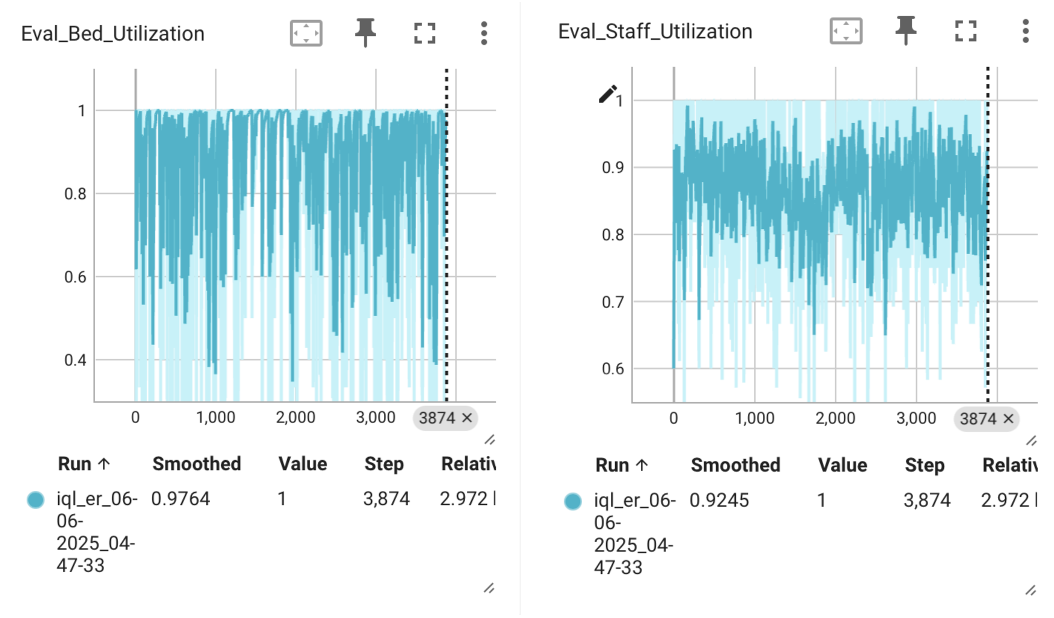
Run ↑	Smoothed	Value	Step	Relativ
iql_er_06-06-2025_04-47-33	22.7075	23.0379	3,874	2.972 l

Run ↑	Smoothed	Value	Step	Relati
iql_er_06-06-2025_04-47-33	144.1879	150.0268	3,874	2.972

(a) Average Patient Wait Time

(b) Severity-Weighted Patient Wait Time

Figure 9: The average wait time for this IQL variant was 22.71 timesteps, which was lowest among all of the tested agents. Similarly, the severity weighted wait time was 144.18 timesteps, lowest among all agents, partially explaining the higher reward received by the agent.



Run ↑	Smoothed	Value	Step	Relativ
iql_er_06-06-2025_04-47-33	0.9764	1	3,874	2.972 l

Run ↑	Smoothed	Value	Step	Relativ
iql_er_06-06-2025_04-47-33	0.9245	1	3,874	2.972 l

(a) Bed Utilization

(b) Staff Utilization

Figure 10: Staff and bed utilization for this IQL variant was 97.64% and 92.45%, respectively. These results and the previous plots explain the higher average reward for the agent and demonstrate its better performance compared to the other agents.

5.4 Quantitative Evaluation

We quantitatively evaluated the performance of all three agents using key metrics, such as average return, patient wait time, and resource utilization. The table below summarizes these results, providing a side-by-side comparison highlighting each agent's ability to manage emergency room operations.

Table 1: Performance Comparison

Method	Avg Reward	Std Dev	Avg Wait Time	Staff Utilization	Bed Utilization
Behavior Cloning	0.45	**0.15**	26.02	0.81	0.34
IQL	0.48	0.19	24.93	0.74	0.88
Offline IQL	**0.52**	0.17	**22.71**	**0.98**	**0.92**

6 Discussion & Future Work

The behavior cloning agent trained on the manually created expert data established a comparable baseline to test the efficacy of the IQL agent in this environment. Despite a theoretical maximum reward of 1, the BC agent achieved a reward of 0.45, representing a decent return with a relatively low standard deviation of 0.15. The environment-specific metrics revealed that the results were similar to those of ER departments in the US. The average wait time was around 26.02 minutes, approximately the national average Horwitz et al. (2010). Furthermore, the bed and staff utilization was low due to the expert having limited information regarding the current state of the emergency room department.

Both IQL agents improved on the benchmark set by behavior cloning, with the offline-exploitation variant achieving the highest performance among all agents. By leveraging the observation space containing the entire state of the ER, the agent could take more informed actions when prioritizing patient treatment. This advantage is reflected in the plots, where the bed and staff utilization peaked at approximately 97% and 92%, respectively. Additionally, the agent reduced the wait time by nearly 4 minutes, demonstrating its ability to improve patient outcomes without delaying necessary care.

The IQL agent's performance demonstrates RL's potential to optimize triage decision-making. By leveraging a comprehensive, high-dimensional representation of the ER state, the agent can be used in conjunction with clinicians to prioritize patients more effectively. Rather than replacing clinical judgment, these models serve as assistive tools, allowing staff to focus more on accurate patient symptom logging and diagnosis. Future work includes adopting a hybrid approach by combining offline and online RL. The offline model will provide a baseline, while the online version will incorporate real-time feedback using direct preference optimization.

Integrating this agent within existing EHR systems will require interpretability, trust, and ethical oversight. To mitigate any potential bias, the model must be trained on a diverse and representative dataset of patient health records. Given the sensitivity of protected health information (PHI), secure data access and HIPAA compliance must be established to protect patient privacy. Additionally, the recommended actions taken by the agent will need to be reviewed by a certified nurse, as incorporating a human-in-the-loop is essential in the medical field, particularly for diagnostic systems.

7 Conclusion

Emergency room triaging remains a significant issue in the medical field due to resource limitations, staff burnout, and overcrowding. This paper aims to solve these issues by presenting a novel method to integrate reinforcement learning into ER flow. After creating a custom ER environment and testing offline RL agents, we demonstrate the potential for RL agents to make dynamic, context-aware triage decisions that consider patient acuity and real-time resource availability. While challenges remain in terms of interpretability and privacy, this works a step towards augmenting triage decision-making, leading to better patient outcomes.

8 Team Contributions

Since I completed the entire project individually, the project was entirely my contribution.

Changes from Proposal The main goal of the proposal has remained the same throughout the research project. Initially, we proposed using additional offline agents, such as CQL, and implementing DPO to provide real-time human feedback to the agent for more realistic agent performance. However, due to time constraints, we opted against implementing CQL and DPO, instead focusing on improving the ER environment and IQL's performance.

References

Albert Buchard, Baptiste Bouvier, Giulia Prando, Rory Beard, Michail Livieratos, Dan Busbridge, Daniel Thompson, Jonathan Richens, Yuanzhao Zhang, Adam Baker, Yura Perov, Kostis Gourgoulias, and Saurabh Johri. 2020. Learning Medical Triage from Clinicians Using Deep Q-Learning. arXiv:2003.12828 [cs.AI] https://arxiv.org/abs/2003.12828

Centers for Disease Control and Prevention. 2025. *FastStats: Emergency Department Visits*. https://www.cdc.gov/nchs/fastats/emergency-department.htm

Alexander M. Chang, Jeffrey M. Caterino, Carlos A. Camargo, Edward R. Melnick, and Judd E. Hollander. 2022. Evaluation of Version 4 of the Emergency Severity Index in US Emergency Departments for the Rate of Mistriage. *JAMA Network Open* 5, 9 (2022), e2233066. https://doi.org/10.1001/jamanetworkopen.2022.33066

Rebecca Goodwin, John Cyrus, Radina L. Lilova, Sreedhatri Kandlakunta, and Taruna Aurora. 2024. Emergency department observation units: A scoping review. *JACEP Open* 5, 4 (2024), e13254. https://doi.org/10.1002/emp2.13254

Astrid Guttmann, Michael J. Schull, Marian J. Vermeulen, and Therese A. Stukel. 2011. Association between waiting times and shortterm mortality and hospital admission after departure from emergency department: population based cohort study from Ontario, Canada. *BMJ* 342 (2011), d2983. https://doi.org/10.1136/bmj.d2983 Published 1 June 2011, accessed June 8, 2025.

Leora I. Horwitz, Jeremy Green, and Elizabeth H. Bradley. 2010. US emergency department performance on wait time and length of visit. *Annals of Emergency Medicine* 55, 2 (2010), 133–141. https://doi.org/10.1016/j.annemergmed.2009.07.023

Riya Amit Mahato. 2023. Patient Flow and Triage Simulation Data. https://www.kaggle.com/datasets/riyaamitmahato/patient-flow-and-triage-simulation-data. Accessed: April 23, 2025.

Mahnaz Samadbeik, Andrew Staib, Justin Boyle, Sankalp Khanna, Emma Bosley, Daniel Bodnar, James Lind, Jodie A. Austin, Sarah Tanner, Yasaman Meshkat, Barbora de Courten, and Clair Sullivan. 2024. Patient flow in emergency departments: a comprehensive umbrella review of solutions and challenges across the health system. *BMC Health Services Research* 24, 1 (2024), 274. https://doi.org/10.1186/s12913-024-10725-6

Matthew S. Smith. 2025. Radiology embraces generative AI to streamline productivity. *Business Insider* (5 June 2025). https://www.businessinsider.com/radiology-embraces-generative-ai-to-streamline-productivity-2025-6 Accessed June 8, 2025.

Devinder Thapa, In-Sung Jung, and Gi-Nam Wang. 2005. Agent Based Decision Support System Using Reinforcement Learning Under Emergency Circumstances. In *Advances in Natural Computation (ICNC 2005) (Lecture Notes in Computer Science, Vol. 3610)*, Lipo Wang, Ke Chen, and Yew-Soon Ong (Eds.). Springer, 888–892. https://doi.org/10.1007/11539087_119

Li-Hung Yao, Ka-Chun Leung, Chu-Lin Tsai, Chien-Hua Huang, and Li-Chen Fu. 2021. A Novel Deep Learning–Based System for Triage in the Emergency Department Using Electronic Medical Records: Retrospective Cohort Study. *Journal of Medical Internet Research* 23, 12 (2021), e27008. https://doi.org/10.2196/27008