A Survey of Reinforcement Models Applied to a Model of ICU Sepsis Treatment

Author(s): Nicolas Smits, James Randolph, Abdullah Paracha

Abstract

We originally proposed developing an A2C agent for peptide sequence generation but shifted to the ICU-Sepsis environment due to the complexity of required computational biology tools. In this project, we implemented and compared DQN, Prioritized DQN, Double DQN, Dueling DQN, Multi-Step DQN, and Rainbow DQN agents. Dueling DQN achieved the best returns, while Rainbow and Multi-Step DQN failed to learn, highlighting the advantages of architectural improvements and the challenges of tuning complex methods.

1. Introduction and background

In critical care, sepsis remains one of the most complex and high-risk conditions to treat, requiring timely, contextual decisions. Reinforcement learning (RL) has demonstrated potential in supporting such decisions by learning optimal treatment strategies from experience. The ICU-Sepsis-v2 environment provides a simulated setting to evaluate RL agents in this context. In this project, we compare different RL algorithms (Double DQN, Prioritized DQN, Rainbow DQN, Deuling DQN, Noisy DQN) on the ICU-Sepsis-v2 environment. We evaluated each method in terms of return (survival rate), and episode lengths [1][2].

The ICU-Sepsis environment models sepsis treatment as a Markov Decision Process (MDP), where each episode follows a patient's trajectory through the ICU. Constructed from real patient data in the MIMIC-III database and pre-processed for clinical relevance, the MDP features 716 discrete health states that abstract combinations of physiological indicators, vitals, and clinical features [1]. The action space includes 25 discrete treatment options representing different dosages of intravenous fluids and vasopressors [1]. State transitions are based on empirical observations, with mean transition distributions applied when specific actions are unobserved, ensuring every action remains available in every state. Rewards are sparse and outcome-based: +1 for patient survival, 0 for death, and 0 for all intermediate states [1]. The environment is episodic with a discount factor $\gamma = 1$, emphasizing long-term survival over short-term gains. DQN is particularly suited to this environment due to its discrete action space, ability to learn from sparse rewards via experience replay and target networks, and use of function approximation to generalize across the high-dimensional state space [1]. Empirical results show DQN matches the performance of Proximal Policy Optimization (PPO) while outperforming traditional methods like Q-Learning, Sarsa, and Soft Actor-Critic (SAC).

Previous works have demonstrated that reinforcement learning could identify optimal treatment policies for sepsis by modelling ICU decision making as an MDP trained on previous data [3]. Their work shows the potential of RL to improve long term patient outcomes, motivating further exploration of both value-based and policy-based methods in critical health settings [3]. Subsequent studies have highlighted challenges in evaluating RL agents using offline health care data and highlighted the importance of simulated environments for safe deployment [4]. In a comparison of five RL algorithms on the ICU-sepsis environment DQN was the most performant [1]. Motivated by this strong baseline performance, we propose that augmenting the base DQN will further improve performance on ICU-sepsis environment. We found that prioritized replay and dueling nets improved performance the most compared to base DQN.

2. Model implementations

DQN: DQN approximates the action-value function Q(s, a) using a deep neural network parameterized by \theta. The Bellman equation is used as the foundation for learning:

$$Q^*(s,a) = \mathrm{E}[\, R(s,a,s') \, + \gamma \max a' Q^*(s',a')] \quad (1)$$

where s'denotes the next state, γ is the discount factor (set to 1 in ICU-Sepsis), and $R(s,a,s')$ is the reward received [2]. DQN minimizes the following loss:

$$\mathcal{L}(\theta) = \mathrm{E}(s,a,r,s') \sim \mathcal{D}\left[\left(r + \gamma \max_{a'} Q_\theta(s',a') - Q_\theta(s,a)\right)^2\right] \quad (2)$$

where $\mathcal{D}$ denotes the experience replay buffer and θ are the parameters of a periodically updated target network. Our base DQN agent uses a fully connected two-layer network architecture, ε-greedy action selection with a linear decay schedule, and a replay buffer for off-policy updates.

Prioritized DQN: To enhance sample efficiency, we incorporated Prioritized Experience Replay. Instead of uniformly sampling from $\mathcal{D}$, transitions are sampled with probabilities proportional to their temporal-difference (TD) error magnitude $|\delta|$.

$$P(i) = \frac{|\delta_i|^\alpha}{\left(\sum_j |\delta_j|^\alpha\right)}, \qquad w_i = \left(\frac{1}{N\, P(i)}\right)^\beta \quad (3)$$

where $\alpha \in [0,1]$ controls prioritization strength. Importance sampling weights, w_i, correct for the introduced bias, where β anneals toward 1 over training. We implemented a prioritized replay buffer compatible with both uniform and importance-weighted updates.

Double DQN: Standard DQN tends to overestimate action values due to the maximization step in the target computation. Double DQN addresses this by decoupling action selection and action evaluation [2]:

$$\mathcal{L}(\theta) = \mathrm{E}(s,a,r,s')\left[\left(r + \gamma Q_\theta\left(s', arg\max_{a'} Q_\theta(s',a')\right) - Q_\theta(s,a)\right)^2\right] \quad (4)$$

This reduces overoptimistic value estimates and stabilizes learning. Double DQN was implemented by modifying the target computation in the update rule.

Duelling DQN: In Dueling DQN, the Q-network architecture is restructured to separately estimate a state value $V(s)$ and advantage function $A(s,a)$, combining them as [2]:

$$Q(s,a) = V(s) + \left(A(s,a) - \frac{1}{|\mathcal{A}|}\sum_{a'} A(s,a')\right) \quad (5)$$

This design allows the network to better distinguish between the importance of being in a particular state and the importance of selecting a specific action.

Multi-Step DQN: Multi-Step DQN incorporates rewards over multiple steps to improve credit assignment and accelerate reward propagation. Instead of the standard 1-step return, the n-step return is used [2]:

$$R_t^{(n)} = \sum_{k=0}^{n-1} \gamma^k\, r_{t+k} + \gamma^n \max_{a'} Q(s_{t+n}, a') \quad (6)$$

Multi-step returns enable the agent to backpropagate richer signals from near-future rewards, balancing bias and variance in the target estimates.

Noisy-Network DQN: In noisy DQN, the traditional exploring method using ε-greedy is replaced with learnable parametrized noise added to the weights of linear layers in the Q-Network. The key idea is to replace standard linear layers with noisy layers. This method allows the agent to explore the environment in a state-dependent, adaptive manner.

$$Q(s,a) = (\mu_W + \sigma_W \cdot \varepsilon_W)x + (\mu_b + \sigma_b \cdot \varepsilon_b) \quad (7)$$

Distributional DQN: Traditional DQN approximates the expected return $Q(s,a)$ as a scalar. In contrast, Distributional DQN models the entire distribution of returns $Z(s,a)$, allowing richer information about uncertainty and variability in returns to be captured. Instead of approximating:

$$Q(s,a) = \mathrm{E}[Z(s,a)] \quad (8)$$

the agent learns a categorical distribution over possible return values, represented by a fixed set of N atoms spaced uniformly between minimum and maximum possible returns. Each action thus maps to a probability distribution over these atoms. The network outputs logits corresponding to the unnormalized probabilities over the atoms for each action. During learning, the Bellman operator is applied to the distribution, and the resulting shifted and discounted distribution is projected back onto the fixed support z_i via a projection operator Φ [2]:

$$\Phi \mathrm{T}\, Z(s,a) = project\left(r + \gamma\, Z\left(s', \max_{a'} \mathrm{E}[Z(s',a')]\right)\right) \quad (9)$$

The loss minimized is the cross-entropy between the projected distribution and the predicted distribution. This approach allows the agent to reason not just about the expected value but about the full shape and risk profile of the return distribution, potentially improving decision-making under uncertainty.

2.1 Rainbow DQN

Multi-step distributional loss is used. The value distribution in S_{t+n} is attracted according to the cumulative discount and shifted by the truncated n-step discounted return. The resulting loss is the KL divergence between the target distribution and the L2-projection of the target distribution onto z. Double Q-learning is used. Proportional prioritized replay is used. Transitions are sampled according to the KL loss. The network architecture contains a linear encoder which is fed into two-layer networks with noisy layers to estimate the value and action streams with output dimensions of $N_{atoms} x N_{actions}$. The value and action streams are aggregated like in Dueling Nets and a logarithmic softmax layer obtains the normalized parametric distributions used to return the returns' distributions. For all models, hidden dimensions are of size 128.

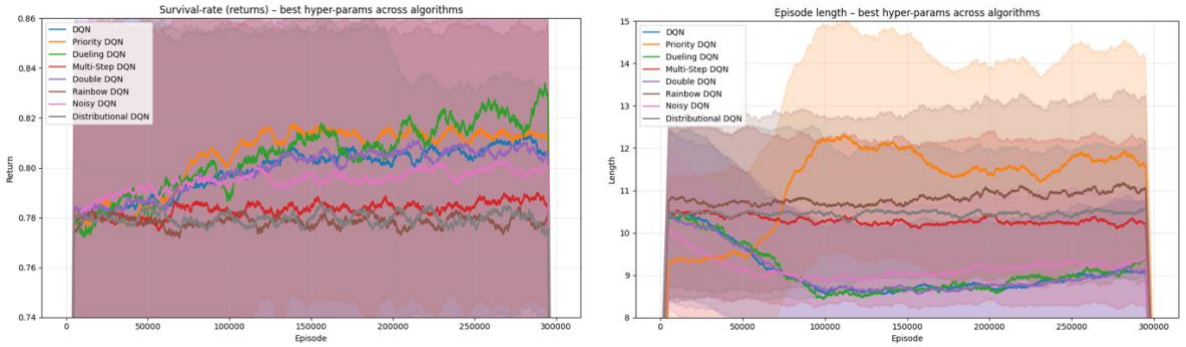
3. Results

The results in Table 1 summarize the result of relatively small (~ 4) hyperparameter grid searches over each of the algorithms evaluated for 10 seeds per environment for 300k episodes.

Table 1 Average Returns/Episode Length on Last 10,000 episodes

Model	Average Final Returns	Average Final Episode Length
DQN	0.8089	9.023
Priority DQN	0.8135	**11.774**
Dueling DQN	**0.8238**	9.061
Multi-Step DQN	0.7868	10.29
Double DQN	0.8056	8.939
Rainbow DQN	0.7782	10.977
Noisy DQN	0.8005	9.256
Distributional DQN	0.7813	10.486

Figure 1 a) Average Returns Over Training, b) Average Episode Length Over Training



Note: a sharp drop off was observed toward the very final episodes no matter the number of episodes evaluated which was deemed to be a trait of the environment, thus "Last" indicates episodes 270-280k.

Our experiments revealed several important trends across the DQN variants tested on the ICU-Sepsis environment. Dueling DQN achieved the highest final returns (~0.8238), steadily improving over episodes but exhibiting larger fluctuations compared to other methods. The duelling architecture likely enhanced the agent's ability to distinguish state importance from action advantages, leading to stronger policies at the cost of higher variance. Prioritized DQN performed second best, converging roughly 100,000 episodes sooner than Dueling DQN but reaching a slightly lower final return (0.8135). This reflects the benefits of prioritizing important transitions for faster early learning, but potentially at the cost of bias accumulation. Interestingly, its performance closely mirrored the standard DQN baseline reported in the original paper [1], although our model achieved convergence slightly earlier, likely due to differences in environment vectorization and training configuration. DQN and Double DQN displayed very similar learning curves, reaching final returns about 0.1 lower than Dueling DQN. Double DQN's decoupled target selection reduced overestimation bias but did not markedly outperform standard DQN, possibly due to sparse rewards and limited hyperparameter tuning. In contrast, Rainbow, Multi-Step, and Distributional DQN failed to learn, achieving returns similar to naive random policies (~0.78). Likely causes include the small hyperparameter grids (typically only 4 parameters) and limited runtime per model (300,000 episodes, 10 seeds), both of which hindered the tuning needed for more complex methods. Vectorized environments may also have disrupted n-step returns and episode termination tracking, particularly impacting methods like Multi-Step and Rainbow. Episode length trends further support these findings. Prioritized DQN achieved the longest episodes (11.7 steps) but still fell short of the original DQN's performance. DQN followed the original baseline trend, initially plateauing at 9.4 steps before learning to extend episodes to around 11.7. Meanwhile, Dueling, Double, and Noisy DQN initially saw episode lengths decline below their starting points (~10.5) before recovering, a pattern likely linked to the side effects of environment vectorization, which altered reset dynamics and may have temporarily disrupted learning signals.

4. Conclusion and future work

Overall, our investigation into examining and extending DQN on the ICU-Sepsis environment fell short of improving on the original study conducted by Choudhary et al through the improvements to DQN proposed by Hessel et al in their seminal paper [1][2]. The key limitation of our investigation was the limited hyperparameter search and small number of trail seeds. Thus, future work should be centered on more exhaustive grid searches. We still maintain that methods such as priority/duelling DQN hold promise in optimising the ICU-Sepsis environment as these methods outperformed the DQN baseline we established. The same may be true for the noisy, distributional, and rainbow DQN, though our results were inconclusive.

References

[1] K. Choudhary, D. Gupta, and P. S. Thomas, 'ICU-Sepsis: A Benchmark MDP Built from Real Medical Data', 2024.

[2] 'Papers with Code - Rainbow: Combining Improvements in Deep Reinforcement Learning'. Accessed: Apr. 25, 2025. [Online]. Available: https://paperswithcode.com/paper/rainbow-combining-improvements-in-deep

[3] M. Komorowski, L. A. Celi, O. Badawi, A. C. Gordon, and A. A. Faisal, "The Artificial Intelligence Clinician learns optimal treatment strategies for sepsis in intensive care," *Nature Medicine*, vol. 24, no. 11, pp. 1716–1720, 2018. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6371300/

[4] O. Gottesman, F. Johansson, J. Komorowski, A. Faisal, D. Sontag, L. A. Celi, and M. Komorowski, "Guidelines for reinforcement learning in healthcare," *Nature Medicine*, vol. 25, pp. 16–18, 2019. https://pubmed.ncbi.nlm.nih.gov/30617332/