Deep Attention Q-Network for Medical Recommendation

Ibrahim Al Khalil RIDENE
ibrahim-al-khalil.ridene@student-cs.fr

Omar ARBI
omar.arbi@student-cs.fr

Abstract

This project presents a Deep Attention Q-Network (DAQN) for personalized medical recommendation using reinforcement learning. By leveraging a Transformer-based attention mechanism, DAQN attends to sequential patient data for context-aware decisions. We simulate a medical recommendation environment with custom rewards based on medical effectiveness and safety. Experimental results show that DAQN outperforms a baseline DQN, highlighting the benefit of temporal modeling in clinical decision support.

1. Introduction

Reinforcement Learning (RL) has shown promising results in sequential decision-making tasks across various domains such as robotics, games, and healthcare. In the medical field, RL has the potential to assist in optimizing treatment strategies for patients by learning policies that maximize long-term clinical outcomes. A particularly relevant problem is **personalized treatment recommendation**, where the goal is to choose the right medication or intervention over time for a specific patient based on their evolving health state.

The paper *Deep Attention Q-Network for Personalized Treatment Recommendation* [1] introduces a novel architecture—Deep Attention Q-Network (DAQN)—that extends the classical Deep Q-Network (DQN) [2] by incorporating attention mechanisms. This allows the model to selectively focus on the most informative past health states when estimating the action-value function.

1.1. From DQN to DAQN

DQN approximates the optimal action-value function $Q^*(s,a)$ using a deep neural network with parameters θ, and updates are made using the Bellman equation:

$$Q^*(s,a) = \mathbb{E}_{s'}\left[r + \gamma \max_{a'} Q^*(s',a') \mid s,a\right], \tag{1}$$

where s and a are the current state and action, r is the immediate reward, and γ is the discount factor. The network is trained by minimizing the loss:

$$L(\theta) = \mathbb{E}_{(s,a,r,s')\sim D}\left[(y - Q(s,a;\theta))^2\right], \tag{2}$$

with the target $y = r + \gamma \max_{a'} Q(s',a';\theta^-)$, and θ^- being the parameters of a target network.

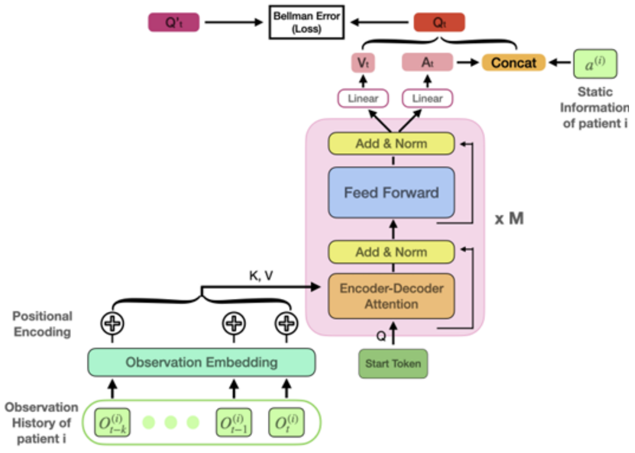

Figure 1. Architecture of DAQN [1]. Attention is applied to historical observations to dynamically focus on important time steps, combining this with static patient information.

DAQN enhances DQN by introducing a temporal attention module over the patient's historical states. Instead of simply feeding the last k states into the Q-network, DAQN uses an attention mechanism to weight past hidden states:

$$\alpha_t = \text{softmax}\left(h_t^\top W h_{t-1}\right), \tag{3}$$

$$c_t = \sum_{i=1}^{t} \alpha_i h_i, \tag{4}$$

where h_t represents hidden states and c_t is the context vector passed to the Q-network. This allows DAQN to dynamically emphasize relevant time steps in the patient's history, enabling more informed treatment recommendations.

Figure 1 illustrates the overall architecture of DAQN. The key stages include:

- **Observation Embedding:** Historical observations of patient i, denoted as $O^{(i)}_{t-k}, \ldots, O^{(i)}_{t}$, are passed through an embedding layer with positional encoding to preserve temporal order.

- **Encoder-Decoder Attention:** A transformer-based attention block takes the embedded sequence as *keys* and *values*, while the query is initialized from a start token. The attention weights allow the model to selectively focus on different parts of the patient's history depending on the current context.
- **Feed Forward + Normalization:** The output of the attention layer is processed through multiple transformer blocks (stacked M times), with layer normalization and residual connections.
- **Static Patient Features:** Static attributes $a^{(i)}$ (e.g., age, gender, conditions) are concatenated with the transformer output before passing through a linear layer to compute Q_t.
- **Loss Computation:** The Q-value Q_t is compared against the target value Q'_t to compute the Bellman loss, which is backpropagated to update the model.

This attention mechanism allows DAQN to focus on relevant portions of the observation sequence, leading to better treatment decisions in cases where long-term patient history matters.

1.2. Our Experimental Setting

Despite the promise of DAQN, one major limitation we encountered was the lack of publicly available medical datasets containing both detailed patient trajectories and corresponding drug/intervention actions. To address this, we designed a two-phase experimental setup:

Approach 1: Synthetic Actions on Time Series Patient Data. In the first approach, we used real-world patient time series data that contained no action (drug) information. To emulate a treatment policy, we synthetically generated action labels and associated rewards. We then applied DAQN to examine the effectiveness of RL in such constrained settings. The results were analyzed with respect to the accumulated rewards and learning curves. However, this approach was limited by the unrealistic nature of the synthetic actions and lacked interpretability in a clinical context.

Approach 2: Real Patient Data with Drug Interventions. Due to the limitations of the first approach, we later obtained a dataset that includes both patient states and drug prescriptions. This allowed us to implement and compare both DQN and DAQN architectures under more realistic conditions. We evaluated their performance across different metrics and hyperparameters (reward, policy coefficient ϵ , discount factor γ, ...), observing the improvements enabled by attention-based learning over standard DQN in terms of both stability and accumulated reward.

Our experiments highlight the significance of using attention in reinforcement learning for medical decision-making, especially when dealing with long-term patient histories.

2. Fisrt Approach: Synthetic Actions on Time Series Patient

In this work, we propose a reinforcement learning (RL) framework for optimizing medical recommendations using a real-world Intensive Care Unit (ICU) dataset. The design of the RL environment, along with the action, reward, and observation structures, was tailored to capture the complexities of patient care in a clinical setting.

2.1. Environment and Dataset

The RL environment was built around the `processed_icu_data.csv` dataset, which comprises time-series data of patients admitted to the ICU. Each record in the dataset is identified by a unique `RecordID` and includes various physiological measurements such as the Glasgow Coma Scale (GCS), heart rate (HR), mean arterial pressure (NIMAP), urine output, temperature, respiratory rate, and additional hemodynamic and biochemical parameters. The environment simulates patient trajectories over time by generating episodes using a sliding window approach over each patient's sequential data.

To illustrate the variability and characteristics of key input features in the dataset, Figure 2 shows the distribution and potential outliers for selected physiological features.

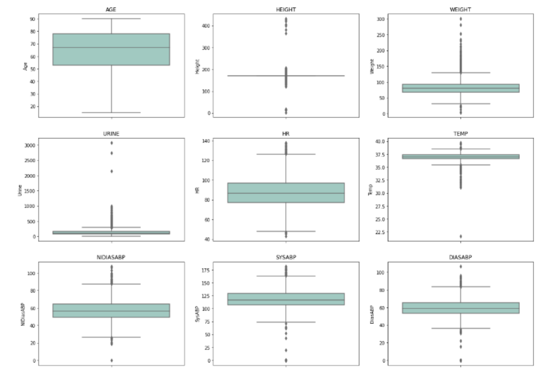

Figure 2. Distribution and potential outliers for selected physiological features from the ICU dataset used to construct the RL environment.

Due to the substantial presence of missing values in critical features (Figure 3), we adopted multiple imputation strategies instead of dropping incomplete records, as doing so would drastically reduce the available patient data. Specifically, missing values were handled using interpolation for time-series features, mean imputation for static features, and filling with zeros where clinically appropriate.

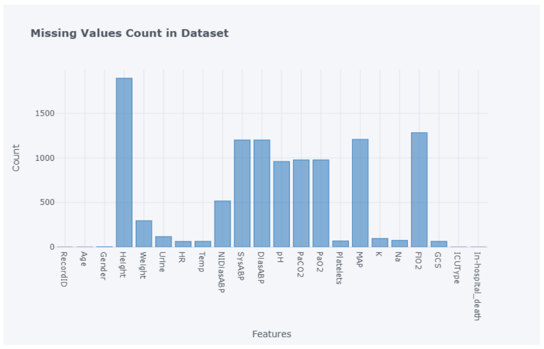

Figure 3. Count of missing values per feature in the ICU dataset prior to imputation.

2.2. Actions, Rewards, and Observations

Actions: The action space in the environment consists of four discrete actions. The optimal action at each time step is derived from a rule-based mapping:

- If the patient's NIMAP is below 65, an intervention corresponding to the administration of vasopressors is suggested.
- If the urine output is below 30, an intervention corresponding to the administration of IV fluids is recommended.

These conditions are cumulative, yielding four possible actions ranging from no intervention (action 0) to combined interventions (action 3).

Rewards: The reward function is designed to reflect the clinical effectiveness of the interventions. Specifically, when the urine output exceeds 30 and NIMAP is above 55, the patient state is considered acceptable, and a reward of zero is provided. Negative rewards are assigned for deviations from the optimal range, with larger penalties for greater deviations. The raw rewards are normalized across the dataset to ensure stability during training.

Observations: The observations used by the RL agent consist of a subset of the patient's physiological measurements, excluding metadata such as `RecordID`, `Time`, and the computed `Action` and `Reward` fields. These features capture the evolving clinical state of the patient and are provided to the agent as sequential inputs over a fixed history window, thereby enabling the agent to model temporal dependencies in the patient's condition.

2.3. Experimental Results and Analysis

We conducted experiments to evaluate our reinforcement learning framework, particularly focusing on the convergence of training, the estimated treatment effects, the reward distributions, and hyperparameter optimization.

2.3.1. Training Loss Analysis

Figure 4 illustrates the training loss over 2000 episodes. The loss curve shows a rapid initial decline, stabilizing after approximately 500 episodes. This pattern suggests that the agent effectively learns to approximate optimal Q-values over training, demonstrating stable convergence and effective learning dynamics.

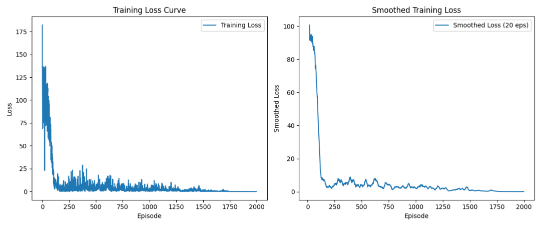

Figure 4. Training loss over 2000 episodes. The left panel displays raw loss values, while the right panel shows a smoothed curve computed over 20 episodes.

2.3.2. Estimated Treatment Effects

Next, we aimed to assess the estimated treatment effects derived from the learned policy. Surprisingly, the resulting treatment effect estimates shown in Figure 5 appeared nearly identical across the different interventions, suggesting minimal differentiation in the effectiveness of distinct actions. To investigate this unexpected result further, we analyzed the distribution of rewards associated with each action.

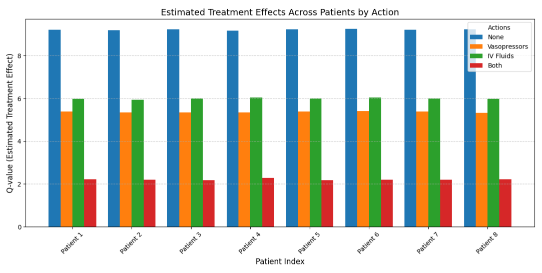

Figure 5. Estimated treatment effects (Q-values) across different actions for a sample of eight patients. Notably, the Q-values appear highly uniform across patients, indicating minimal variation in the model's perceived effectiveness of interventions.

2.3.3. Reward Distribution Analysis

Figure 6 presents the distribution of rewards corresponding to each action category. The analysis reveals that most of the rewards concentrate in very similar ranges, explaining the minimal differentiation observed in the estimated treatment effects. Specifically, the actions ”IV fluids” and ”None” provided almost no variability in rewards, while ”Both” and ”Vasopressors” actions showed higher variability. This homogeneous reward landscape likely contributed to the agent's difficulty in distinctly valuing different interventions.

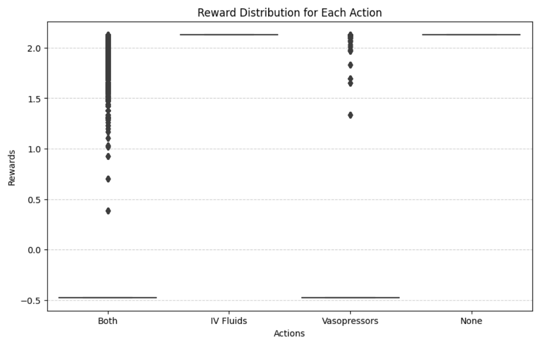

Figure 6. Reward distribution for each action category.

2.4. Hyperparameter Optimization

To further improve the RL agent's performance, we conducted hyperparameter optimization using Optuna, performing 23 trials aimed at maximizing the overall policy reward. Figure 7 illustrates the optimization history, showing how quickly the policy reward improved and stabilized within the initial trials, indicating the effectiveness of Optuna in efficiently identifying high-performing hyperparameter configurations.

Figure 8 provides insights into the relative importance of each hyperparameter on maximizing the cumulative reward. The discount factor γ emerged as the most critical parameter, contributing nearly half (48%) of the total importance. This underscores the essential role of appropriate discounting in balancing immediate and future rewards within clinical decision-making tasks. Other parameters, such as the model dimension (`model_dim`), batch size, and learning rate (`lr`), also showed notable importance, suggesting that careful tuning of these parameters is beneficial for enhancing the agent's effectiveness.

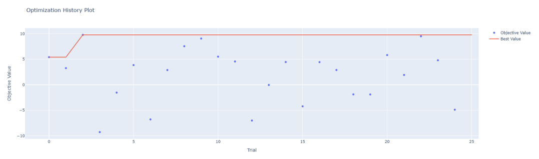

Figure 7. Optimization history plot from Optuna, demonstrating the rapid improvement and stabilization of the cumulative reward across trials.

Overall, the optimization process validated the significance of carefully chosen hyperparameters in enhancing the RL agent's performance and provided a clear direction for further refinements to achieve even greater clinical effectiveness.

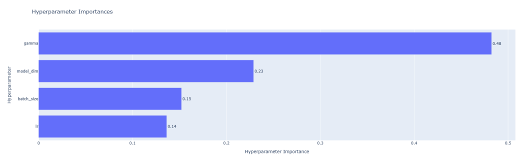

Figure 8. Hyperparameter importance plot, showing the relative contribution of each parameter toward maximizing overall policy reward.

3. Second Approach: Real Patient Data with Drug Interventions

In this work, we implement a drug recommendation system using Reinforcement Learning (RL), with a focus on modeling real patient data. We propose and compare two architectures: a standard Deep Q-Network (DQN) and a Deep Attention Q-Network (DAQN) based on transformers. The DQN processes single time-step data, while the DAQN leverages sequential information over multiple clinical observations.

3.1. Data Description

The dataset used in this project consists of two main components: a patient dataset and a drug dataset.

3.1.1. Patient Data

Each patient is described by a combination of numerical and categorical features. The numerical features include *age*, *heart rate*, *temperature*, and the decomposed values of *blood pressure* (systolic and diastolic). Categorical features consist of *gender*, up to three medical conditions (*condition_1*, *condition_2*, *condition_3*), and *allergies*.

DQN Patient Format: Each patient is represented by a single row containing all features. This format is suitable for the Deep Q-Network (DQN) model, which uses a flat representation of patient states.

DAQN Patient Format: To support sequence modeling in DAQN, the patient data is augmented into sequences of fixed length (e.g., 5). Categorical features remain constant across the sequence, while numerical values vary nearby the original value to act as a short period time series.

3.1.2. Drug Data

The drug dataset contains metadata and therapeutic profiles for a variety of medications. Each entry includes a unique *drug ID*, *drug name*, a *primary condition* and *secondary condition*, known *contraindications*, *side effects*, and a base *effectiveness score* in $[0, 1]$.

3.1.3. Reward Signal Construction

The reward function is based on the compatibility between the selected drug and the patient's clinical profile. It increases with matching primary and secondary conditions and penalizes for contraindications, allergies, and multiple side effects. The resulting reward is scaled between $[0, 1]$, ensuring stable learning across agents.

3.2. Model Architectures

We implement two distinct architectures for drug recommendation:

Deep Q-Network (DQN): A feedforward neural network that maps a single patient observation (comprising both static and dynamic features) to Q-values over the action space (available drugs). The model consists of fully connected layers with ReLU activations and dropout for regularization.

Deep Attention Q-Network (DAQN): A transformer-based architecture designed to capture temporal patterns in patient data. It uses a decoder-only attention mechanism 1 where a learnable query token attends over a sequence of dynamic observations. Static features are embedded and fused with the attention output to compute final Q-values.

Both models are trained using the standard Q-learning objective with experience replay and ϵ-greedy exploration.

3.3. Environment & Reward Design

We model the drug recommendation task as a reinforcement learning environment using the OpenAI Gym interface. At each episode, the agent is presented with a patient state and must choose one drug from a discrete set of available medications.

Observation Space:

- *DQN:* A single patient observation, combining both numerical and categorical features.
- *DAQN:* A sequence of numerical observations (e.g., vital signs) with static categorical features (e.g., gender, conditions).

Action Space: A discrete action representing the selection of a drug from the dataset.

Reward Function: The reward captures clinical compatibility between the selected drug and patient profile. It is based on:

- Matching primary and secondary conditions (+)
- Absence of contraindications and allergies (+)
- Presence of contraindications or allergies (–)
- Number of side effects (penalty)

The final reward is scaled between 0 and 1.

3.4. Training & Evaluation

We train both the DQN and DAQN agents using episodic interaction with the environment for a fixed number of episodes. Each episode corresponds to a single patient interaction and ends after one action is taken.

Training Configuration:

- Optimizer: Adam with learning rate 0.001
- Discount factor $\gamma \in \{0.3, 0.5, 0.7, 1.0\}$
- Epsilon-greedy policy for exploration, with both fixed and decaying schedules
- Replay memory size: 10,000; batch size: 32

Evaluation Protocol: Agents are evaluated over 1000 unseen patient cases. The evaluation uses a greedy policy (no exploration), and the performance is measured by the average reward per episode. Additionally, ablation experiments are conducted to study the effects of exploration strategies and discount factors.

3.5. Results

This section presents a comprehensive evaluation of our proposed DAQN model compared to a standard DQN baseline. We analyze the performance across different configurations, including model architecture, exploration strategies, and discount factors. Finally, we test the trained agent's ability to recommend appropriate drugs for unseen patient profiles.

3.5.1. Comparison: DAQN vs. DQN

In this subsection, we compare the performance of the proposed Deep Attention Q-Network (DAQN) against a standard Deep Q-Network (DQN) baseline. Both models were trained for 1500 episodes across 100 runs, and their average rewards per episode were recorded.

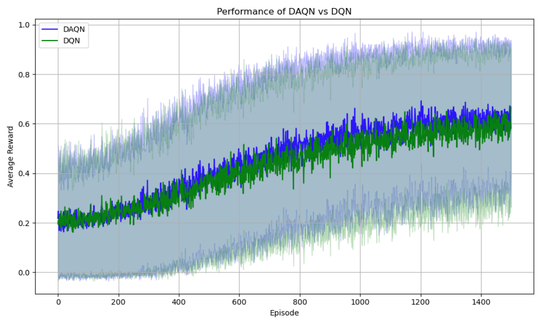

Figure 9. DAQN vs. DQN: Average Reward per Episode

Analysis. The DAQN outperforms the DQN in terms of learning speed, stability, and final performance. Several observations support this:

- **Faster Convergence:** DAQN starts improving rapidly in the first few hundred episodes, with its curve becoming noticeably steeper than DQN's early on.
- **Higher Performance Plateau:** Around episode 1000, the DAQN achieves an average reward of approximately

0.60, whereas the DQN only reaches around **0.55**. This gap remains consistent through the rest of training.

- **Stability:** The DAQN curve exhibits slightly less fluctuation near the convergence phase, with a narrower standard deviation band, indicating more stable learning and better policy generalization.

Limitations. Despite the observable improvement, the performance gap between DAQN and DQN remains relatively small. This is primarily due to the synthetic nature of the time-series data used in training DAQN. The numerical features vary slighty across time steps, but categorical variables are fixed, limiting the richness of temporal patterns the model can exploit. In more realistic clinical scenarios, where patients' health evolves meaningfully over time, we expect the advantage of DAQN to become more significant.

3.5.2. Exploration Strategy Comparison: Fixed vs. Decreasing Epsilon

This experiment evaluates the impact of exploration strategy on the performance of the DAQN agent. Specifically, we compare a fixed $\epsilon = 0.1$ policy with a variable ϵ that decays from 1.0 to 0.05. The results over 100 runs and 1500 episodes are visualized in Figure 10.

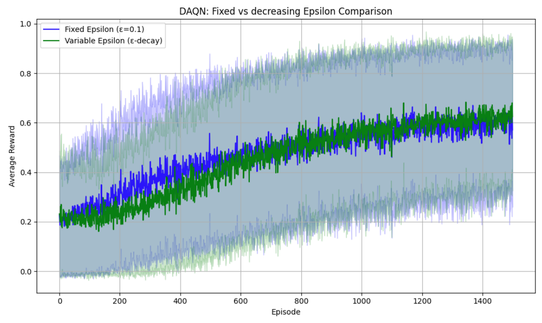

Figure 10. DAQN: Fixed vs Decreasing Epsilon Strategy Comparison

Analysis. Both exploration strategies lead to progressive improvements in performance, but they exhibit distinct learning dynamics:

- **Early Training (Episodes 0–500):** The fixed epsilon strategy outperforms the decaying one. For instance, by episode 400, the fixed epsilon agent reaches an average reward near **0.45**, whereas the decaying epsilon agent remains closer to **0.35**. This suggests that maintaining a moderate and consistent exploration level helps discover effective actions early (not excessive exploitation like decreasing epsilon strategy in this episode range).
- **Mid to Late Training (Episodes 500–1500):** The variable epsilon strategy gradually catches up and eventually surpasses the fixed epsilon policy. Around episode **1200**, the variable epsilon reaches an average reward of **0.63**, slightly ahead of the fixed strategy's **0.60**.
- **Stability and Variance:** The confidence interval for the variable epsilon policy becomes slightly narrower in later episodes, indicating better stability and policy convergence as exploration decays over time.

While both strategies perform competitively, the decaying epsilon policy enables more stable and optimal long-term learning by shifting gradually from exploration to exploitation. However, the fixed epsilon policy accelerates early reward acquisition, making it suitable for shorter training horizons.

3.5.3. Effect of Discount Factor γ

To investigate the role of the discount factor γ, we trained the DAQN agent using several values: $\gamma \in \{1.0,\ 0.7,\ 0.5,\ 0.3\}$. The results are presented in Figure 11, showing the average reward per episode across different settings.

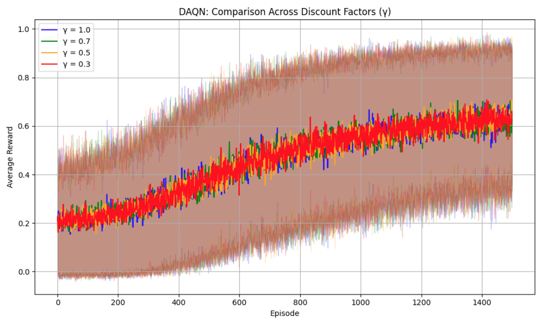

Figure 11. DAQN: Comparison Across Discount Factors γ

While the curves appear similar across all γ values, this behavior is expected given the design of our environment. Each episode consists of a single decision step, after which the episode terminates. As a result, the return depends solely on the immediate reward, and the discount factor has no effective influence:

$$Q(s,a) = r \quad \text{since the episode ends after one step.}$$

Therefore, differences in performance are not attributable to γ but rather to randomness in training and exploration. This confirms that in single-step environments, the discount factor does not meaningfully impact learning outcomes.

3.5.4. Qualitative Evaluation via Patient Testing

To qualitatively assess the models, we performed inference using both trained agents (DQN and DAQN) on a held-out test set of four patients. Each model outputs a ranked list of drug recommendations, from which we extract the top choice and analyze its suitability with respect to patient conditions, contraindications, and effectiveness.

Example of tested Patients

Patient 1: Hypertension, Diabetes — Allergic to Penicillin

DQN Recommendation: **Sumatriptan** — Secondary condition match: *hypertension*, but contraindicated for *diabetes*, lowering its suitability.

DAQN Recommendation: **Amitriptyline** — Matches *hypertension* as secondary condition without directly conflicting contraindications. Thus, a safer and more compatible option.

Patient 2: Hypertension, Heart Failure, Cholesterol

DQN Recommendation: **Sumatriptan** — Same as Patient 1, includes contraindications (*heart disease*) relevant to this patient.

DAQN Recommendation: **Amitriptyline** — Again, better aligned due to the absence of immediate contraindications. The DAQN agent shows improved consistency across patients with similar profiles.s

Overall, both models provide reasonable recommendations; however, DAQN shows slight improvements in safety and clinical match—especially in patients with multiple conditions and contraindications. Differences in selected drugs reflect DAQN's enhanced ability to model temporal trends, even when limited by synthetic sequence generation. While both systems agree on general categories of treatment, DAQN recommendations often show better condition alignment and avoidance of risky drug interactions.

4. Conclusion

This project introduced a Deep Attention Q-Network (DAQN) for personalized medical recommendation, leveraging temporal patient data through a Transformer-based architecture. Compared to a standard DQN, DAQN showed improved convergence and slightly higher reward performance, benefiting from its ability to attend over sequential observations. However, due to the synthetic nature of the time series, the gains remained moderate.

Evaluation on test patients confirmed DAQN's ability to recommend drugs better aligned with patient conditions and safety profiles. These results validate the potential of attention-based RL models in healthcare decision-making, especially when enriched with real temporal data.

References

[1] Yinchong Li et al. Deep attention q-network for personalized treatment recommendation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):795–802, 2020. 1

[2] Volodymyr Mnih et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. 1