Interpretable Dynamic Treatment Regimes via Upside-Down Reinforcement Learning

Siddhant Kapse
Indian Institute of Technology
Kharagpur
India
siddhantkapse4@iitgp.ac.in

Julien Hansen
Université de Liège
Belgium
Julien.Hansen@uliege.be

Damien Ernst
Université de Liège
Belgium
dernst@uliege.be

Abstract

Reinforcement Learning (RL) is increasingly explored for sequential decision-making in safety-critical domains such as healthcare, yet evaluation remains dominated by aggregate reward metrics that provide limited insight into how policies arrive at their decisions. This gap is particularly important in Dynamic Treatment Regimes (DTRs), where policies must be both effective and auditable.

We investigate an interpretability and auditing pipeline built on Upside-Down Reinforcement Learning (UDRL), which reformulates policy learning as a goal-conditioned supervised mapping from state and desired cumulative return to action. This formulation provides a structured interface for analysing policy behaviour under explicit outcome objectives and enables the use of both neural and intrinsically interpretable model classes within a unified framework.

The proposed pipeline combines temporal local feature attribution with retraining-based feature ablation. Feature importance is estimated independently at each timestep and aggregated across rollouts to produce temporally resolved importance profiles. These qualitative explanations are then evaluated through interventional analysis by removing selected features from the state representation and retraining policies under identical conditions. Resulting performance degradation is used as evidence of functional feature reliance rather than as a direct measure of causal importance.

The framework is evaluated in a controlled benchmark environment (CartPole) and across clinically-inspired simulations from DTR-Bench. The experiments show that policies achieving comparable rewards can nevertheless exhibit substantially different attribution stability and feature dependence patterns. In sparse-reward environments, the pipeline additionally identifies policies that fail to develop consistent state-dependent structure despite achieving non-trivial returns.

These results position interpretability not merely as a post-hoc visualization tool, but as a diagnostic mechanism for analysing policy reliability and behavioural structure in sequential decision-making systems.

Keywords

Reinforcement Learning, Interpretability, Dynamic Treatment Regimes, Upside-Down Reinforcement Learning, Feature Ablation

1 Introduction

Artificial intelligence (AI) is increasingly being integrated into clinical decision-making systems, enabling a transition from standardised treatment protocols toward more adaptive and patient-specific care strategies. This shift is closely related to the framework of Dynamic Treatment Regimes (DTRs), where medical interventions are selected sequentially in response to a patient's evolving physiological condition [32]. DTRs are particularly relevant in chronic and high-risk conditions such as cancer, sepsis, and diabetes, where treatment decisions must continuously adapt to changing patient trajectories.

Reinforcement Learning (RL) has emerged as a natural framework for modelling such sequential decision processes, and recent benchmarking efforts such as DTR-Bench provide controlled simulation environments for evaluating RL-based treatment policies [9, 30]. Most existing evaluations, however, remain primarily focused on aggregate reward metrics. While these metrics are useful for measuring optimization performance, they provide limited insight into how policies utilise patient information, whether policies rely consistently on clinically meaningful variables, or whether apparently successful policies have learned robust decision structure at all.

This lack of transparency remains a major barrier to the deployment of RL systems in healthcare. Standard value-based and policy-gradient methods frequently rely on highly parameterised function approximators whose internal decision structure is difficult to inspect. As a result, policies may achieve strong numerical performance while remaining effectively opaque to clinicians and regulators [35]. In safety-critical domains, however, reward optimization alone is insufficient; policies must also be analysable, auditable, and diagnostically interpretable.

This work investigates Upside-Down Reinforcement Learning (UDRL) as a framework for structured policy auditing [5]. Unlike conventional RL formulations based on bootstrapped value estimation, UDRL reformulates policy learning as a supervised mapping from state and desired outcome commands to actions [38]. Concretely, policies are trained as mappings of the form

$$f(s_t, d_r, d_t) \rightarrow a_t$$

where s_t denotes the current state, d_r the desired cumulative return, and d_t the desired remaining horizon. This goal-conditioned formulation provides a comparatively clean interface for analysing how policies respond to explicit outcome objectives over time. It additionally enables the use of both neural and intrinsically interpretable supervised learning models within the same behavioural framework.

Building on this formulation, we propose a temporal interpretability pipeline that combines local feature attribution methods with interventional validation through feature ablation. Rather than relying solely on static global importance estimates, explanations are computed locally at each timestep along policy trajectories and

subsequently aggregated across rollouts to analyse how feature reliance evolves over time. The resulting attribution patterns are then evaluated through retraining-based ablation experiments designed to test whether features identified as important correspond to functional policy dependence.

The primary contribution of this work is therefore not a new reinforcement learning algorithm, but a structured methodology for auditing sequential decision-making policies. We evaluate the framework first in a controlled control environment (CartPole), where fixed-state temporal analysis is feasible, and subsequently in clinically-inspired DTR environments characterised by stochastic patient initialization and high-dimensional state spaces. Across these settings, the results illustrate how policies with similar reward regimes may nevertheless exhibit substantially different interpretability characteristics, attribution stability, and dependence on input variables.

More broadly, this work argues that interpretability should be treated not merely as a post-hoc explanatory layer, but as a practical mechanism for evaluating policy reliability in high-stakes sequential decision-making systems.

2 Related Work

Reinforcement Learning (RL) provides a general framework for sequential decision-making under uncertainty and has been widely applied across control, robotics, and healthcare. Classical value-based and policy-gradient methods learn policies either through Bellman bootstrapping or direct gradient optimization of expected return. In recent years, substantial progress has been achieved through deep value-based methods, actor-critic architectures, and offline RL algorithms [14, 25, 26].

Beyond deep policy optimization, a parallel line of research has sought to improve the interpretability of RL policies through post-hoc explanation methods and inherently interpretable approximators. Within the batch RL paradigm, Fitted Q-Iteration (FQI) and related methods employ supervised regression over Bellman targets, often using tree-based ensembles such as Random Forests and Extra-Trees [10]. These approaches naturally expose feature relevance through split statistics and value-function structure. Similarly, deep value-based methods such as DQN have been analyzed using perturbation-based saliency maps and local sensitivity visualizations to identify influential state regions during decision-making [16, 19]. More recent Explainable RL (XRL) research has explored broader structural abstractions, including reward decomposition methods that separate composite reward signals into semantically meaningful components [22], causal explanation frameworks based on Structural Causal Models [31], and concept-level policy representations using interpretable bottlenecks or symbolic abstractions.

However, in many classical RL settings, explanations remain fundamentally coupled to value estimation quality and Bellman bootstrapping stability. Feature attributions are therefore computed relative to estimated long-horizon value functions rather than directly against outcome-conditioned policy objectives. Consequently, attribution quality may itself be influenced by approximation error, distributional shift, or instability in the underlying RL optimization process, motivating frameworks that decouple policy auditing from intermediate value estimation.

To address this limitation, recent work in sequence-modeling and goal-conditioned reinforcement learning reformulates policy acquisition as a supervised learning problem over trajectories [7, 37]. UDRL directly conditions policies on desired cumulative return and remaining horizon, learning a mapping from the extended state $[s_t, d_r, d_t]$ to actions without relying on iterative Bellman updates [37, 38]. Related sequence-modeling approaches, including Decision Transformer and trajectory-based conditional prediction methods, similarly cast RL as sequence prediction conditioned on target returns [7, 20]. While these methods primarily focus on performance and stability in offline settings, they provide a broader perspective in which policies are explicitly parameterized by desired outcomes.

Our work leverages this goal-conditioned formulation specifically for interpretability. By grounding policy explanations in outcome-conditioned action mappings rather than intermediate value estimates, the proposed framework avoids attribution errors introduced by Bellman approximation, providing a structurally motivated interface for auditing feature dependence under explicit long-horizon objectives. This property is particularly relevant in safety-critical domains where interpretability must remain stable under clinically meaningful outcome targets.

The optimization of DTRs—sequential, adaptive decision rules for patient care—represents a central challenge in precision medicine [6]. RL provides a natural formalism for modeling sequential treatment decisions in this setting, where a policy selects interventions according to a patient's evolving physiological state to optimize long-term health outcomes [27, 41]. This alignment has motivated growing research in critical care, oncology, and chronic disease management.

However, the safety-critical nature of medicine introduces fundamental constraints absent from conventional RL domains. Online exploration through trial-and-error interaction is ethically infeasible in real clinical settings [39]. As a result, most clinical RL research operates within the offline RL paradigm, where policies must be learned exclusively from historical datasets [25, 39]. This restriction introduces well-known technical challenges including distributional shift, extrapolation error, partial observability, and irregular sampling [14, 23, 27]. Related work on Medical Dead-ends [11, 24] identifies high-risk clinical states using value-function analysis in offline sepsis data, providing a complementary perspective to our sensitivity-based auditing framework, which instead focuses on functional feature reliance without requiring explicit value estimation.

A further obstacle is the evaluation gap in clinical RL. Because newly learned policies cannot be prospectively deployed on patients during development, researchers depend heavily on Off-Policy Evaluation (OPE) methods [21, 42]. However, empirical evidence suggests that OPE estimates can become unreliable in moderate-to-low data regimes commonly encountered in healthcare, limiting confidence in estimated policy quality [29]. This validation bottleneck remains a major barrier to trustworthy clinical deployment [13].

To support safer and more reproducible evaluation, several high-fidelity in silico clinical benchmarks have recently been introduced. Frameworks such as DTR-Bench [30] and EpiCare [18] simulate

clinically-inspired patient dynamics under varying levels of noise, missingness, and treatment complexity. These environments provide controlled testbeds for comparing RL algorithms under realistic sequential decision-making conditions [29]. Importantly, evaluations on these benchmarks have demonstrated that high-capacity black-box policies can exhibit brittle or unstable behavior under realistic perturbations [30]. When failures occur, the opacity of such models makes it difficult to identify whether policies rely on clinically meaningful variables, spurious correlations, or unstable reward shortcuts [13]. These findings motivate the need for RL frameworks that support not only competitive performance, but also systematic auditing of feature dependence and policy behavior under clinically relevant outcome objectives.

3 Part 1 : Interpretation in CartPole Environment

Before applying our UDRL-based interpretation pipeline to the complex domain of Dynamic Treatment Regimes (DTR), we first validate its efficacy and reliability on a well-understood control benchmark: the CartPole [3] environment. We use CartPole as a controlled sanity check to validate the correctness of the interpretability pipeline before applying it to higher-dimensional domains. The CartPole task involves balancing a pole vertically on a moving cart, where the environment provides a continuous 4-dimensional state vector as described in Table 1.

Table 1: CartPole Environment Components

States $\mathcal{S}$	**Actions $\mathcal{A}$**	**Rewards r_t**
Continuous 4D vector: - x: Cart Position - $\dot{x}$: Cart Velocity - θ: Pole Angle - $\dot{\theta}$: Pole Angular Velocity	Discrete: 0: Push left 1: Push right	+1 for every step taken

Following the Upside-Down Reinforcement Learning (UDRL) paradigm [5, 37], we recast the reinforcement learning problem as a supervised learning task. Instead of learning a value function or policy directly from rewards, UDRL learns a behavior function f that maps the current state and a desired outcome to an action.

Concretely, at each timestep t, we construct training samples of the form $(s_t, R_t, T_t) \rightarrow a_t$, where s_t is the observed state, $R_t = \sum_{k=t}^{T} r_k$ denotes the return-to-go, and $T_t = T - t$ represents the remaining time until the end of the trajectory. The model is trained to predict the action taken under these conditions using standard supervised learning objectives.

At inference time, the behavior function is queried as $f(s_t, d_r, d_t)$, where d_r (desired return) and d_t (desired time horizon) specify the outcome the agent is expected to achieve. For the CartPole environment, following prior work, both d_r and d_t are set to 200, corresponding to the maximum achievable cumulative reward in the episodic setting.

A key distinction of UDRL is that the desired return d_r and desired time horizon d_t act as dynamic command variables that condition the policy. Unlike standard RL policies that depend solely on the state s_t, the UDRL behavior function explicitly incorporates the intended outcome into its decision-making process. As a result, the same state can yield different actions depending on the specified (d_r, d_t).

This formulation has important implications for interpretability. Since (d_r, d_t) are treated as part of the model input, they are included alongside state variables in all feature attribution analyses. This allows us to examine not only which environmental features influence decisions, but also how strongly the policy conditions on the desired objective and planning horizon.

In our implementation, we follow the training procedure described in prior UDRL work, generating trajectories through interaction with the CartPole environment and converting them into supervised datasets. Since the UDRL formalism introduced by [36] allows for using any machine learning algorithm suitable for solving supervised learning tasks, the behavior functions are trained using multiple model classes, including Neural Networks, Random Forest [4], Extra Trees [15], AdaBoost [12], and XGBoost [8], enabling a comparative analysis of interpretability across different function approximators.

Experimental setup: All reinforcement learning environments were implemented using the Gymnasium framework [40].

We follow the standard UDRL training procedure described in prior work [5, 37], while explicitly specifying the configuration used in our experiments. Each behavior function is trained for 500 episodes of interaction with the CartPole environment, which is sufficient for convergence in this setting.

During training, transitions of the form (s_t, R_t, H_t, a_t) are stored in an experience replay buffer with a capacity of 700, from which mini-batches are sampled to construct the supervised learning problem. Exploration during data collection is governed by an ϵ-greedy strategy with $\epsilon = 0.2$, which remains fixed throughout training.

For the neural network policy, we use a multi-layer perceptron with two hidden layers of size 64 and ReLU activations, optimized using the Adam optimizer with a learning rate of 10^{-3}. Tree-based models (Random Forest, Extra Trees, AdaBoost, and XGBoost) are implemented using standard library implementations with consistent hyperparameter settings across environments to ensure comparability.

To account for training variability, all experiments are repeated across $K = 5$ independent runs with different random seeds. Each trained policy is evaluated over 100 standard environment episodes initialized using the default stochastic reset mechanism of the CartPole environment. Policies are executed deterministically at evaluation time using the learned behavior function conditioned on high desired return and horizon values.

For each run i, the mean episodic reward μ_i is computed over the 100 evaluation episodes. Final performance statistics are reported as:

$$\mu = \frac{1}{K}\sum_{i=1}^{K}\mu_i, \quad \sigma = \sqrt{\frac{1}{K}\sum_{i=1}^{K}(\mu_i - \mu)^2}$$

where σ reflects variability across independent training runs rather than within-episode reward variance.

3.1 Fixed-State Temporal Interpretation Analysis

A key challenge in interpreting reinforcement learning policies is understanding how feature reliance dynamically evolves across consecutive timesteps within an episode. In unconstrained environments, evaluating local attributions along completely randomized rollouts introduces severe compounding variance, often rendering post-hoc explanations unfalsifiable or highly subjective [2]. To isolate purely temporal variations in policy behavior from initial condition noise, we establish a trajectory-conditioned diagnostic probing protocol, anchoring our methodology within established frameworks for instance-level behavior analysis [16, 33].

Importantly, this protocol is used exclusively for interpretability analysis and not for quantitative policy evaluation. All reward-based performance results reported previously are obtained using the standard stochastic initialization mechanism of the environment.

For temporal interpretation, all rollouts are initialized from the same predefined state

$$s_0 = [0.1, 0.1, 0.1, 0.1]$$

rather than randomly sampled initial conditions. This controlled initialization reduces variability induced by differing starting states and allows the analysis to focus specifically on how feature importance evolves throughout the trajectory.

From this shared initial state, we execute the trained UDRL policy for $N = 100$ independent rollouts. At each timestep t, local feature attribution scores are computed for the action selected by the policy. Because rollout lengths vary across simulations, temporal aggregation at timestep t is performed only over trajectories that remain active at that timestep. This avoids biasing late-episode importance estimates toward shorter failed trajectories, while naturally reflecting the behavior of policies that sustain longer episodes. The resulting procedure yields a discrete time-indexed importance function $I_f(t)$ for each feature f, enabling analysis of how policy reliance on different variables evolves over time.

In the UDRL setting, the command variables d_r and d_t denote the desired cumulative return and remaining time horizon, respectively. These variables are included alongside the environmental state variables during attribution analysis.

For visualization, we report representative early-, mid-, and late-episode timesteps. In Figures 1 and 2, the selected timesteps correspond to comparable phases of trajectory progression rather than fixed absolute indices, since episode lengths may vary slightly across rollouts and model classes.

3.1.1 *Interpretability of Tree-Based Models (RF, ET, Boost).*

Path-Conditioned Local Impurity Attribution. For tree-based behavior functions, we estimate feature importance using a path-conditioned local impurity attribution method derived from Mean Decrease in Impurity (MDI) [17]. Unlike classical global MDI, which aggregates impurity reduction across all nodes and samples in the training distribution, our analysis is conditioned on the exact decision path traversed by each state instance during policy execution.

Concretely, for an extended state $\tilde{s}_t = [s_t, d_r, d_t]$ at timestep t, we trace the corresponding decision path through each tree in the ensemble using `decision_path`. Impurity reductions are computed only for nodes visited by that state instance, thereby attributing importance exclusively to features involved in the routing of the current policy decision. For tree k with traversed node sequence $\mathcal{P}_k(\tilde{s}_t)$, the local attribution for feature f is computed as:

$$\phi_f^{(k)}(\tilde{s}_t) = \sum_{n \in \mathcal{P}_k(\tilde{s}_t),\, v_n = f} [\mathrm{imp}(n) - \mathrm{imp}(\mathrm{child}(n))]$$

where v_n denotes the splitting feature at node n, and $\mathrm{imp}(\cdot)$ represents the node impurity (Gini impurity in our implementation). Local attributions are summed across all trees in the ensemble and subsequently averaged across $N = 100$ fixed-state rollouts to obtain a temporally resolved importance profile $I_f^{\mathrm{MDI}}(t)$.

This local path-conditioned formulation is deliberate: rather than capturing average feature relevance over the full training distribution, it characterizes which features actively influence the policy's decisions along the observed trajectory progression.

Local Perturbation Sensitivity *(Permutation-Style).* As a complementary model-agnostic analysis, we estimate feature sensitivity through a local perturbation-based procedure inspired by Permutation Importance (PI) [1]. However, unlike classical permutation importance, which evaluates dataset-level performance degradation after globally shuffling a feature, our implementation operates at the individual state level.

For each feature i at timestep t, Gaussian noise $\delta \sim \mathcal{N}(0, 0.1)$ is injected into the corresponding feature dimension while all remaining features are held fixed. The importance score is then computed by measuring whether the perturbation changes the policy's predicted action:

$$\phi_i^{\mathrm{perturb}}(\tilde{s}_t) = \mathbb{1}\left[\pi(\tilde{s}_t^{(i+\delta)}) \neq \pi(\tilde{s}_t)\right]$$

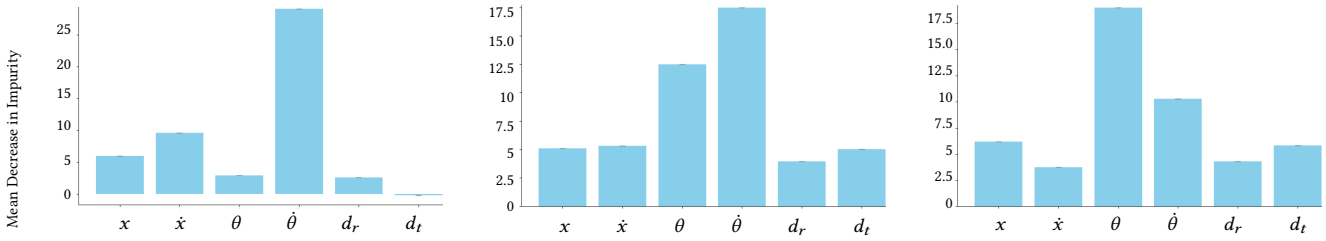
where $\pi(\cdot)$ denotes the learned behavior policy. This procedure therefore measures the local sensitivity of the policy's action selection to perturbations in each feature dimension.

The resulting instance-wise sensitivity scores are averaged across the $N = 100$ fixed-state rollouts to obtain temporal sensitivity profiles, with both mean and standard deviation reported at each timestep. Although PI is model-agnostic and was computed across all trained policies, its temporal visualization is shown here only for the representative Random Forest model for clarity and comparison with MDI. Complete temporal plots for all model-method combinations are omitted for brevity.

3.1.2 *Interpretation of the Neural Network (NN).*

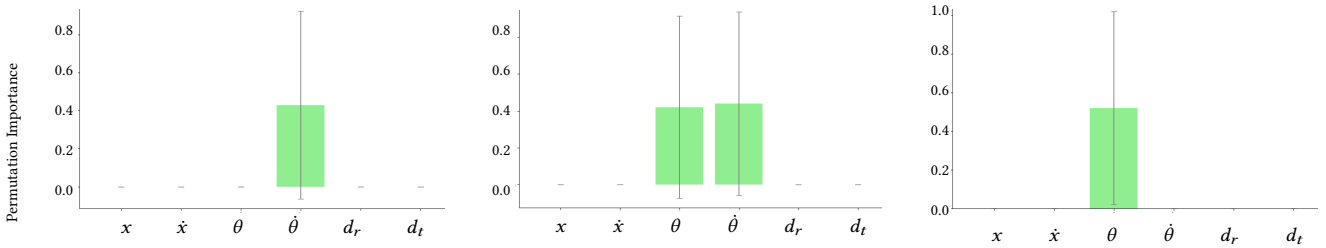
For the NN policy, we apply two post-hoc explanation techniques.

SHAP. To analyze the neural network behavior function, we employ SHAP (SHapley Additive exPlanations) [28], which estimates feature contributions through Shapley value approximations derived from cooperative game theory. For each extended state $\tilde{s}_t = [s_t, d_r, d_t]$ encountered during policy execution, KernelSHAP is used to compute local feature attributions with respect to the predicted action probabilities.



(a) Early Episode (b) Mid Episode (c) Late Episode

Figure 1: ***Observation:*** **MDI identifies angular velocity ($\dot{\theta}$) as the primary global feature for impurity reduction, reinforcing the structural importance of dynamics over position in the learned policy.**



(a) Early Episode (b) Mid Episode (c) Late Episode

Figure 2: ***Permutation Importance Observation:*** **Perturbations in θ and $\dot{\theta}$ induce significant changes in the predicted action distribution, whereas perturbations in x and $\dot{x}$ have negligible effect, indicating dominant reliance on angular features.**

Unlike global feature importance methods, SHAP explanations are computed independently at each timestep, thereby capturing how the policy's reliance on individual state variables evolves along the trajectory. The resulting local SHAP values are aggregated across the $N = 100$ fixed-state rollouts to produce temporally resolved importance profiles. For visualization, we report the mean absolute SHAP magnitude together with the corresponding standard deviation across rollouts.

LIME. We additionally employ LIME (Local Interpretable Model-agnostic Explanations) [34] to obtain local surrogate explanations of the neural network policy. At each timestep t, a `LimeTabularExplainer` fits a sparse linear model in the neighborhood of the current extended state $\tilde{s}_t = [s_t, d_r, d_t]$. The resulting coefficients approximate the local sensitivity of the policy's predicted action to perturbations in each input feature.

For all interpretability methods, local explanations are computed independently at each timestep and aggregated across $N = 100$ rollouts. The temporal importance profile for feature f is computed as:

$$I_f(t) = \frac{1}{N}\sum_{j=1}^{N}\phi_f^{(j)}(t), \quad \sigma_f(t) = \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(\phi_f^{(j)}(t) - I_f(t)\right)^2}$$

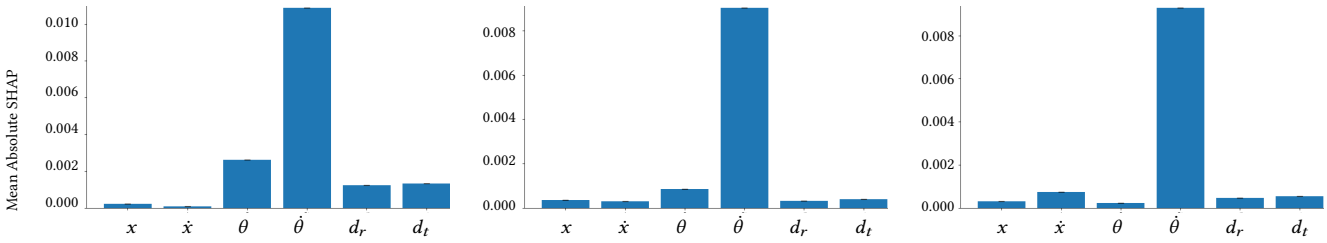
Across all interpretability methods, the pole angle (θ) and pole angular velocity ($\dot{\theta}$) emerge as dominant features for maintaining balance, while cart position (x) exhibits comparatively diminishing influence, demonstrating the robustness of the extracted feature importance across different model explanations [33].

Importantly, absolute attribution magnitudes are not directly comparable across these interpretability methods, as each method defines feature importance within a fundamentally different mathematical space (e.g., local linear surrogate coefficients for LIME versus additive game-theoretic probability contributions for SHAP) [28, 34]. Consequently, our evaluation focuses on relative temporal patterns and cross-method feature ranking consensus rather than direct magnitude comparisons, justifying the independent scaling of the respective episodic visualization axes.

3.2 Validation via Feature Ablation

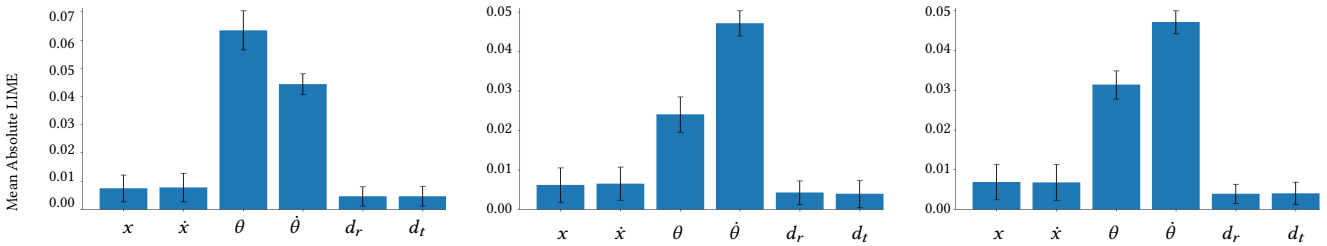
To quantitatively validate the feature importance rankings derived from our temporal analysis, we conduct feature ablation experiments. Based on the observations from Section 3.1, we identify x as a low-importance feature and $\dot{\theta}$ as a high-importance feature. For each selected feature, we construct modified training datasets in which the corresponding feature is removed from the state representation. New UDRL policies are then trained from scratch on these reduced state spaces using the same training protocol as the baseline models. This approach evaluates whether removing a feature significantly degrades policy performance, thereby providing a causal validation of its functional importance. We emphasize that feature ablation provides an interventional measure of functional importance, and does not establish causal relationships.

Evaluation protocol For visualization purposes, we present representative training trajectories from a single run to illustrate qualitative differences in learning dynamics under feature ablation. Due to stochasticity in both environment transitions and training,



(a) Early Episode (b) Mid Episode (c) Late Episode

Figure 3: *SHAP Observation:* $\dot{\theta}$ exhibit consistently higher contribution to the predicted action, while x and $\dot{x}$ remain close to baseline, reinforcing the policy's reliance on angular dynamics.



(a) Early Episode (b) Mid Episode (c) Late Episode

Figure 4: *LIME Observation:* The surrogate model assigns larger coefficients to θ and $\dot{\theta}$, while x and $\dot{x}$ contribute minimally, indicating local decision boundaries are primarily governed by angular features.



Figure 5: Training performance of the neural network under baseline and feature ablation settings. Removing the Ablated (Most) feature ($\dot{\theta}$) leads to significant performance degradation, while removing the Ablated (Least) feature (x) has minor effect. Curves correspond to a representative run.

these trajectories may vary across runs. All quantitative results reported in Table 2 are averaged over 5 independent training runs.

Each trained policy is evaluated over 100 episodes to compute the average episodic reward. Results are aggregated across 5 independent training runs and reported as mean ± standard deviation. The results support the interpretability analysis: removing the Ablated (Most) feature ($\dot{\theta}$) leads to a substantial degradation in performance, whereas removing the Ablated (Least) feature (x) results in only a marginal decrease. This validates that the identified feature importance reflects functional dependence of the learned policy rather than spurious correlations.

Table 2: CartPole Performance under Feature Ablation (Mean Reward ± Std Dev)

Model	Baseline	Ablated (Least)	Ablated (Most)
RF	159.58 ± 23.45	142.84 ± 28.17	132.16 ± 51.44
NN	104.02 ± 4.04	89.56 ± 9.41	71.57 ± 11.43
XGBoost	153.22 ± 32.95	140.83 ± 35.52	69.50 ± 34.58
ExtraTrees	175.50 ± 28.10	155.20 ± 40.50	138.80 ± 45.20
AdaBoost	158.40 ± 35.60	138.10 ± 22.30	75.20 ± 25.40

Overall, the experiments on the CartPole benchmark provide a controlled validation of the proposed UDRL-based interpretation pipeline. The fixed-state temporal analysis yields coherent and evolving insights into policy behavior, and these qualitative findings are supported by quantitative feature ablation experiments. While CartPole represents a fully observable and relatively simple environment, these results demonstrate the feasibility of combining UDRL with interpretability tools to analyze agent behavior in a structured setting. This serves as a foundational step toward examining more complex and safety-critical DTR environments in Part 2, which introduce additional challenges such as higher dimensionality and potential confounding effects.

4 Part 2: Interpretation in Dynamic Treatment Regimes

To evaluate the applicability of the proposed interpretability pipeline in more realistic and safety-critical settings, we extend the analysis to Dynamic Treatment Regime (DTR) environments. DTRs model sequential clinical decision-making problems in which treatment actions are adapted over time according to a patient's evolving physiological state. In these environments, the state corresponds to patient health indicators, actions represent treatment interventions (e.g., dosage adjustments or therapeutic strategies), and rewards encode clinically meaningful objectives such as recovery, survival, or long-term stability.

Compared to standard control benchmarks, DTR environments introduce higher-dimensional state spaces, stochastic patient dynamics, and clinically relevant trade-offs between treatment efficacy and risk. These properties make DTRs a particularly suitable testbed for interpretability analysis, since policy performance alone is insufficient for deployment in safety-critical domains. Beyond achieving favorable returns, learned policies must also exhibit coherent and clinically justifiable decision structure. In particular, understanding which patient variables influence treatment decisions is essential for evaluating policy reliability and identifying potentially unstable or non-robust behavior.

4.1 Pipeline Transfer from CartPole

We emphasize that this section does not introduce new interpretability methods. Instead, we directly apply the interpretation pipeline validated in Section 3. This pipeline consists of (i) qualitative feature importance estimation using multiple interpretability techniques, (ii) identification of the most and least important features based on consensus across methods, and (iii) quantitative validation via feature ablation.

In the CartPole setting, we demonstrated that this pipeline produces consistent and functionally meaningful feature importance estimates. The goal in the DTR setting is therefore to assess whether the same pipeline yields coherent and reliable insights in more complex environments.

In DTR environments, patient initial states are sampled stochastically from environment-defined distributions reflecting physiological heterogeneity. Unlike CartPole, where a single fixed initial state is clinically meaningless and a controlled starting point is reasonable, fixing one patient profile in a clinical environment would produce conclusions specific to that single patient rather than the policy's general behavior. We therefore adopt a population-level aggregation protocol: feature importance scores are computed locally at each timestep across $N = 100$ rollouts initialised from independently sampled patient states, and averaged across rollouts. This produces importance estimates that reflect the policy's reliance on state variables across the patient population rather than for a single trajectory. Formally, unlike the fixed-state CartPole analysis where all rollouts share the same initial condition, the DTR analysis aggregates explanations over stochastically sampled patient initializations:

$$I_f(t) = \frac{1}{N} \sum_{n=1}^{N} \phi_f^{(n)}(t), \quad s_0^{(n)} \sim p_0$$

where p_0 denotes the environment-specific initial state distribution and $\phi_f^{(n)}(t)$ is the local attribution score for feature f at timestep t in rollout n. Consequently, the resulting temporal importance profiles capture expected policy reliance across the patient population rather than for a single fixed trajectory.

The trade-off is that initial-state variability is no longer isolated from policy dynamics; however, this is the appropriate choice for clinical generalizability. For the feature ablation validation, the same stochastic initialization protocol is used, ensuring consistency between the attribution analysis and the ablation experiments.

While all interpretability methods described in Section 3 (MDI, Permutation Importance, SHAP, and LIME) are applied in these environments, we do not replicate the full set of temporal visualizations. Doing so across multiple models and environments would result in a large number of plots without proportionally improving clarity. Instead, we summarize their outcomes through a cross-method feature ranking procedure. Unlike CartPole, where multiple interpretability methods consistently identified pole angle and angular velocity as dominant variables across models, the DTR environments did not exhibit a single universally dominant feature across all policies. Instead, attribution patterns varied across models and environments, reflecting the higher-dimensional and heterogeneous nature of clinical state spaces. The consensus ranking therefore should not be interpreted as identifying globally optimal clinical variables, but rather as identifying features that multiple attribution methods consistently associated with policy behaviour within a given model and environment.

4.2 Experimental Setup

All models are trained using the same UDRL pipeline described in Section 3 to ensure consistency across environments. For DTR tasks, each policy is trained for 500 episodes with a different batch sizes. The action space is environment-specific (e.g., 8 actions for *OberstSepsisEnv*). All experiments are implemented within a unified codebase where only the environment and estimator class are varied.

To account for variability, all experiments are repeated across 5 independent runs with different random seeds. Each trained policy is evaluated over 100 episodes, and performance is reported as the average episodic reward (mean ± standard deviation) aggregated across runs. The same set of model classes is used, including NN, RF, ET, XGBoost, and AdaBoost, with identical implementation settings as described previously. This ensures that differences in performance and interpretability arise from the environment and learned policy structure rather than changes in methodology.

While large-scale benchmarks such as DTR-Bench employ up to 25,000 episodes per seed to distinguish marginal performance differences across 15+ algorithms, our evaluation uses 500 training episodes per run. This is a practical necessity: our retraining-based feature ablation requires training a complete UDRL policy from scratch for each removed feature, making full-scale episode budgets computationally prohibitive. Critically, UDRL's supervised formulation converges faster than standard RL methods since it avoids bootstrapped Bellman updates, making 500 episodes sufficient for stable policy learning in these structured environments. Our goal is verification-oriented auditing rather than marginal performance

ranking, for which the observed ablation effects are sufficiently large to detect at this scale.

4.3 DTR Environments

To test the real-world applicability of our framework, we consider three environments from the DTR-Bench suite: *AhnChemoEnv*, *GhaffariCancerEnv*, and *OberstSepsisEnv*. These environments provide mathematically-defined simulations of clinical treatment processes, including chemotherapy response, tumor progression, and sepsis management.

We restrict our analysis to the original environment configurations. More complex variants (e.g., PK/PD extensions, noise injection, and missing data settings) are left for future work. We exclude *SimGlucoseEnv* from our analysis due to its one-dimensional state space, which is not suitable for meaningful feature importance evaluation. The state spaces, action spaces, and reward structures of the selected environments are summarized in Table 3.

4.4 Quantitative Validation via Feature Ablation

Following the pipeline established in Section 3, feature importance is first estimated using multiple interpretability methods, and a consensus ranking is used to identify the most and least important features. Rather than presenting all intermediate visualizations, we focus on the final stage of the pipeline: quantitative validation via feature ablation.

For each environment, selected features are removed from the state representation, and new policies are trained from scratch using the same UDRL training procedure. The impact of each feature is then assessed by measuring the change in policy performance relative to the baseline.

This provides a direct test of whether the identified feature importance reflects functional dependence of the learned policy. The following tables summarize the performance of our UDRL agents under these conditions across the clinical benchmarks.

Across the cancer-focused environments, consistent architectural differences emerge. In AhnChemoEnv (Table 4), tree-based models such as RF and XGBoost achieve the higher baseline rewards among the other agents, while maintaining moderate dispersion relative to their reward scale. In contrast, the neural network baseline (88.47 ± 2.57) remains numerically stable but operates in a substantially lower reward regime. This gap, together with the consistent degradation observed under feature ablation, indicates that the neural model converges to a conservative or suboptimal policy rather than exhibiting instability.

A similar but more variable pattern was observed in the GhaffariCancerEnv (Table 5). Across multiple models, ablating the consensus-ranked most important features generally produced larger performance degradation than removing the least important features, indicating that the learned policies relied non-trivially on specific state variables. However, the magnitude of the observed standard deviations remained comparatively high across both baseline and ablated settings. This behaviour is consistent with the broader variability reported for several RL baselines in DTR-Bench [30], suggesting that the environment itself exhibits substantial stochasticity and sensitivity to trajectory evolution rather than indicating instability specific to the proposed pipeline. Consequently, while attribution patterns in GhaffariCancerEnv were less sharply separated than in AhnChemoEnv, the overall ablation trends still supported the presence of meaningful feature dependence.

The results for OberstSepsisEnv (Table 6) provide a useful negative control for the interpretability pipeline. In this environment, all tested models—regardless of architecture—converge to a similar mortality-based baseline (approximately -0.80). Unlike the cancer-focused environments, OberstSepsisEnv employs a highly sparse reward signal (+1 for discharge, -1 for death), which limits the ability of agents to learn structured policies due to the absence of informative intermediate reward signals.

Consistent with this, feature ablation (high vs. low importance) results in negligible changes in performance and minimal variance shifts across models. This indicates that the learned policies do not exhibit strong functional dependence on specific state variables. Importantly, the interpretability pipeline does not attribute artificial importance to specific features but instead correctly reflects limited and diffuse feature dependence under the sparse reward setting.

This behavior serves as a validation of the proposed framework. In environments where meaningful feature dependence exists (as in the cancer benchmarks), the pipeline identifies important variables and confirms their functional role through retraining-based ablation. Conversely, in settings where structured learning is limited (as in the sepsis environment), the framework yields near-null responses, reflecting the absence of strong feature dependence. This consistency across contrasting regimes demonstrates that the method functions as a validation mechanism, capable of distinguishing between genuine policy structure and cases where learned behavior lacks meaningful dependence on input variables.

5 Conclusion and Future Discussion

5.1 Conclusion

This work presented a structured interpretability and auditing pipeline for reinforcement learning policies trained using the Upside-Down Reinforcement Learning (UDRL) framework. Rather than introducing a new reinforcement learning algorithm, the primary contribution lies in demonstrating how temporal feature attribution and interventional validation can be systematically combined to analyse policy behaviour across both controlled control environments and clinically-inspired Dynamic Treatment Regime (DTR) benchmarks.

The proposed framework integrates multiple local explanation methods with temporal aggregation and retraining-based feature ablation. In the CartPole environment, where controlled fixed-state evaluation is meaningful, the resulting temporal explanations were consistent with known system dynamics and were supported by quantitative ablation experiments. These experiments showed that features identified as important by the attribution methods also produced measurable behavioural degradation when removed, providing evidence that the extracted explanations corresponded to functional policy dependence rather than visual artefacts alone.

When extended to DTR environments, the analysis highlighted substantially different interpretability characteristics across models and environments. Tree-based UDRL agents generally produced more stable and structurally interpretable behaviour functions,

Table 3: Comparison of the DTR environments and their state, action, and reward formulations. The environment definitions summarised here are those of the DTR-Bench suite [30].

Component	AhnChemoEnv	GhaffariCancerEnv	OberstSepsisEnv
State Space S	Continuous 3D vector: – $T(t)$: Tumor Population – $I(t)$: Immune Population – $B(t)$: Drug Concentration	Continuous 7D vector: – $T_p(t)$: Primary Tumor – $N_p(t)$: Primary NK Cells – $L_p(t)$: Primary T-Cells – $C(t)$: Lymphocytes – $T_s(t)$: Secondary Tumor – $N_s(t)$: Secondary NK Cells – $L_s(t)$: Secondary T-Cells	Continuous 6D vector: Normalized vital signs and clinical measurements including Heart Rate, Blood Pressure, Lactate, O_2 Saturation, and cumulative treatments.
Action Space A	Discrete(5): Maps to 5 distinct levels of a single chemotherapy agent's administration rate $u(t)$.	Discrete(25): A 5x5 grid mapping to combined dose levels of two independent treatments: Radiotherapy D and Chemotherapy $v_M(t)$.	Discrete(8): A 2x2x2 grid representing the binary (On/Off) application of three treatments: Antibiotics, Vasopressors, and Mechanical Ventilation.
Reward $r(t)$	$r = \frac{N}{N(0)} - \frac{T}{T(0)} + I - u$ A dense reward balancing normal cell health, tumor reduction, immune response, and drug toxicity.	$r = r_T(t) + r_o(t)$ A sum of a dense reward for tumor reduction (r_T) and a large terminal reward for cure (+100) or death (-100).	$r = +1$ (discharged), -1 (deceased), or 0. A sparse, terminal reward signal based only on the final patient outcome.

Table 4: AhnChemoEnv Performance under Feature Ablation

Model	Baseline	Ablated (Least)	Ablated (Most)
RF	202.12 ± 14.42	184.81 ± 35.94	165.25 ± 40.45
XGBoost	207.48 ± 23.51	196.93 ± 31.49	141.95 ± 63.57
NN	88.47 ± 2.57	84.78 ± 3.20	79.38 ± 5.17
ExtraTrees	182.45 ± 22.31	161.80 ± 38.15	118.25 ± 65.40
AdaBoost	178.90 ± 55.26	155.47 ± 32.16	128.62 ± 38.50

Table 5: GhaffariCancerEnv Performance under Feature Ablation

Model	Baseline	Ablated (Least)	Ablated (Most)
ExtraTrees	97.16 ± 41.17	81.40 ± 63.52	75.20 ± 51.81
RF	112.73 ± 34.94	86.97 ± 49.10	68.50 ± 64.23
XGBoost	103.36 ± 52.15	87.36 ± 78.50	65.45 ± 76.10
AdaBoost	94.45 ± 62.15	89.83 ± 61.26	82.66 ± 75.30
NN	87.83 ± 84.35	72.10 ± 80.59	57.41 ± 95.25

Table 6: OberstSepsisEnv Performance under Feature Ablation

Model	Baseline	Ablated (Least)	Ablated (Most)
ExtraTrees	−0.72 ± 0.61	−0.74 ± 0.63	−0.75 ± 0.62
RF	−0.78 ± 0.59	−0.75 ± 0.58	−0.79 ± 0.58
XGBoost	−0.78 ± 0.65	−0.80 ± 0.66	−0.81 ± 0.64
AdaBoost	−0.82 ± 0.60	−0.81 ± 0.62	−0.83 ± 0.59
NN	−0.79 ± 0.68	−0.80 ± 0.67	−0.82 ± 0.69

while neural policies often exhibited greater variability in both performance and attribution consistency. Importantly, the pipeline was also capable of identifying environments in which policies exhibited weak or unstable feature dependence. In the OberstSepsisEnv setting, uniformly limited ablation effects and inconsistent attribution patterns suggested that several policies did not develop strong reliance on individual state variables under the extremely sparse reward formulation. However, such flat ablation responses may also reflect redundancy or correlation between clinical variables rather than a complete absence of learned structure. This demonstrates that the framework can be used not only to interpret policy behaviour, but also to diagnose settings in which feature reliance remains diffuse or difficult to attribute reliably.

A central observation throughout the experiments is that reward performance alone provides an incomplete description of policy quality in safety-critical domains. Multiple policies achieved competitive reward regimes while exhibiting substantially different levels of attribution stability, feature reliance, and structural transparency. The results therefore support the broader view that policy interpretability and auditability should be evaluated alongside reward optimization, particularly in domains such as healthcare where decisions must remain inspectable and accountable.

Overall, this work establishes a reproducible pipeline for combining goal-conditioned policy learning, temporal interpretability analysis, and interventional validation within reinforcement learning. The framework is intentionally model-agnostic and is designed to serve as an auditing methodology rather than a task-specific learning solution. While the experiments are limited to simulation-based benchmarks, the results demonstrate that structured interpretability analysis can provide useful insight into both policy behaviour and policy failure modes in sequential decision-making systems.

5.2 Future Discussion

Several limitations define the scope of the present study and motivate future work.

First, the experiments intentionally employ relatively conservative model architectures and training configurations. Tree-based models use largely default hyperparameter settings, and neural policies are implemented using comparatively standard architectures without extensive tuning or large-scale optimization. This choice was made to reduce confounding variability during interpretability analysis and to keep attribution behaviour reproducible across environments. More sophisticated architectures or aggressive hyperparameter search may improve reward performance, but would also introduce additional complexity into the attribution analysis. Understanding how interpretability properties evolve with increasing model capacity remains an important open question.

To contextualise the obtained performance regimes, Table 7 compares selected DTR-Bench results [30] with the tree-based UDRL agents used in this study. Results marked with † are taken directly from the original DTR-Bench benchmark and therefore use different evaluation budgets and protocols. The comparison should consequently be interpreted as contextual rather than a strict benchmark ranking. Despite these differences, the proposed agents operate within reward ranges competitive with several standard RL baselines while additionally enabling trajectory-conditioned attribution and feature-ablation analysis.

Table 7: Comparison of Policy Performance across Medical Environments. The rows marked † are reproduced from the DTR-Bench benchmark [30]; the remaining rows are our own results.

Policy	AhnChemo	GhaffariCancer	OberstSepsis
π_b†	62.29 ± 12.62	−90.78 ± 2.96	−0.81 ± 0.58
DQN †	214.18 ± 141.64	134.24 ± 26.45	0.15 ± 0.73
dSAC †	114.06 ± 82.68	93.65 ± 73.98	0.16 ± 0.71
rDQN †	267.78 ± 62.65	63.66 ± 85.66	−0.11 ± 0.77
rC51 †	242.18 ± 82.29	−82.88 ± 16.37	−0.68 ± 0.51
RF	202.12 ± 14.42	112.73 ± 34.94	−0.78 ± 0.59
XGBoost	207.48 ± 23.51	103.36 ± 52.15	−0.78 ± 0.65
ExtraTrees	182.45 ± 22.31	97.16 ± 41.17	−0.72 ± 0.61

Several observations follow from this comparison. In AhnChemoEnv, the proposed tree-based UDRL agents achieve reward regimes comparable to several standard RL baselines, despite using comparatively simple model classes and no extensive hyperparameter tuning. In GhaffariCancerEnv, the ExtraTrees agent exhibits lower variance than many competing methods while remaining within a competitive reward regime. In OberstSepsisEnv, most methods — including standard RL baselines — converge to similarly weak reward regimes, reflecting the severe reward sparsity and difficulty of the environment itself rather than the limitations of a particular algorithm family.

Importantly, the purpose of this comparison is not to claim state-of-the-art reward performance, but to contextualise the interpretability-performance trade-off explored in this work. Many high-performing RL baselines remain effectively black-box policies whose decision structure cannot be directly audited. In contrast, the proposed tree-based UDRL agents permit explicit trajectory-conditioned analysis through the temporal attribution and ablation pipeline developed in Section 3.

The choice of UDRL is similarly motivated by interpretability considerations rather than reward maximization alone. Unlike standard value-based or policy-gradient methods, UDRL formulates policy learning as a direct goal-conditioned supervised mapping $f(s_t, d_r, d_t)$, enabling explanations to be conditioned on explicit desired outcomes. This structure provides a comparatively clean interface for temporal attribution analysis and avoids some of the interpretability indirectness associated with analysing implicit value functions or latent policy objectives. Nevertheless, this work does not claim that UDRL is uniquely compatible with interpretability analysis. Extending the proposed pipeline to alternative RL paradigms, including PPO, DQN, FQI, and offline RL approaches, remains an important direction for future study.

Several methodological limitations should also be noted regarding the interpretation pipeline itself. Feature ablation provides an interventional mechanism for testing whether policies functionally depend on particular variables, but performance degradation should not be interpreted as a direct measure of causal importance. In high-dimensional clinical environments, correlated or redundant variables may allow policies to partially compensate when certain features are removed. Consequently, limited degradation following ablation may indicate redundancy within the state representation rather than the absence of meaningful policy structure. Future work could incorporate counterfactual or causal interpretability methods to better distinguish policy invariance from feature redundancy.

The aggregation procedure used throughout the DTR experiments also reflects an important practical trade-off. Unlike CartPole, clinical environments employ stochastic patient initialization, and feature importance is therefore aggregated across heterogeneous patient trajectories rather than isolated fixed-state rollouts. This population-level evaluation improves clinical generalizability, but reduces the ability to disentangle initial-state variability from policy dynamics. Furthermore, trajectory lengths vary across episodes, meaning that temporal importance estimates at later timesteps are computed from progressively fewer surviving rollouts. While the aggregation procedure preserves meaningful temporal trends, these estimates should not be interpreted as fully stationary statistics over a fixed trajectory distribution.

In addition, the present work only evaluates the original DTR-Bench environment configurations and does not consider the extended variants involving pharmacokinetic/pharmacodynamic (PK/PD) modelling, missing-data mechanisms, or explicit noise injection. These variants introduce substantially greater realism and are known to alter policy learning dynamics. Evaluating the proposed interpretability pipeline under such conditions would provide a stronger assessment of robustness under uncertainty and partial observability.

Finally, all experiments are conducted in simulated environments with known dynamics. Real clinical deployment introduces additional challenges including latent confounding, distributional shift, irregular sampling, and limited observational coverage. Moreover,

while the current study evaluates whether explanations are structurally and functionally meaningful, it does not assess how such explanations are perceived or utilised by clinicians. Human- centred evaluation therefore remains essential before interpretability claims can translate into practical clinical decision support.

Taken together, these limitations reinforce a broader point: in high-stakes sequential decision-making, interpretability should not be treated purely as a post-hoc visualization layer applied after optimising reward. Instead, the ability to inspect, validate, and audit policy behaviour should be considered a core component of reinforcement learning system design itself.

Ethics Statement

All experiments reported in this work were carried out entirely in simulation, using the CartPole control benchmark and the in silico patient simulators distributed with DTR-Bench [30]. No human subjects were involved and no real patient data of any kind were accessed, processed, or generated; no ethics approval or informed consent was therefore required. The clinical environments used here are mathematical models and are not validated for clinical use. The policies and interpretability results reported in this paper are intended solely as a methodological study of policy auditing, and must not be interpreted as clinical guidance or used to inform the treatment of patients.

LLM Usage Disclosure

Large language models were not used to generate the research ideas, the experimental design, the analysis, or the results reported in this work. Their use was limited to routine assistance, namely language editing of author-written prose and general coding assistance during implementation. All scientific content, all experiments, and all conclusions are the authors' own, and the authors have verified every statement and every reference in this paper.

Acknowledgments

This research was conducted at the University of Liège (ULiège), Belgium. The authors acknowledge the academic environment and computational resources that supported this work.

References

[1] André Altmann, Laura Toloşi, Oliver Sander, and Thomas Lengauer. 2010. Permutation Importance: A Corrected Feature Importance Measure. *Bioinformatics* 26, 10 (2010), 1340–1347. doi:10.1093/bioinformatics/btq134

[2] Akanksha Atrey, Kaleigh Clary, and David Jensen. 2020. Exploratory Not Explanatory: Counterfactual Analysis of Saliency Maps for Deep Reinforcement Learning. In *International Conference on Learning Representations (ICLR)*.

[3] Andrew G. Barto, Richard S. Sutton, and Charles W. Anderson. 1983. Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems. *IEEE Transactions on Systems, Man, and Cybernetics* 13, 5 (1983), 834–846. doi:10.1109/TSMC.1983.6313077

[4] Leo Breiman. 2001. Random Forests. *Machine Learning* 45, 1 (2001), 5–32. doi:10.1023/a:1010933404324

[5] Juan Cardenas-Cartagena, Massimiliano Falzari, Marco Zullich, and Matthia Sabatelli. 2024. Upside-Down Reinforcement Learning for More Interpretable Optimal Control. *arXiv preprint arXiv:2411.11457* (2024).

[6] Bibhas Chakraborty and Susan A. Murphy. 2014. Dynamic Treatment Regimes. *Annual Review of Statistics and Its Application* 1 (2014), 447–464. doi:10.1146/annurev-statistics-022513-115553

[7] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. 2021. Decision Transformer: Reinforcement Learning via Sequence Modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 34. 15084–15097.

[8] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 785–794. doi:10.1145/2939672.2939785

[9] Cristian Drudi, Maximiliano Mollura, Li-wei H. Lehman, and Riccardo Barbieri. 2024. A Reinforcement Learning Model for Optimal Treatment Strategies in Intensive Care: Assessment of the Role of Cardiorespiratory Features. *IEEE Open Journal of Engineering in Medicine and Biology* 5 (2024), 806–815. doi:10.1109/OJEMB.2024.3367236

[10] Damien Ernst, Pierre Geurts, and Louis Wehenkel. 2005. Tree-Based Batch Mode Reinforcement Learning. *Journal of Machine Learning Research* 6 (2005), 503–556.

[11] Mehdi Fatemi, Taylor W. Killian, Jayakumar Subramanian, and Marzyeh Ghassemi. 2021. Medical Dead-Ends and Learning to Identify High-Risk States and Treatments. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 34. 4856–4870.

[12] Yoav Freund and Robert E. Schapire. 1996. Experiments with a New Boosting Algorithm. In *Proceedings of the 13th International Conference on Machine Learning (ICML)*. 148–156.

[13] Timothy C. Frommeyer, Michael M. Gilbert, Reid M. Fursmidt, Youngjun Park, John Paul Khouzam, Garrett V. Brittain, Daniel P. Frommeyer, Ean S. Bett, and Trevor J. Bihl. 2025. Reinforcement Learning and Its Clinical Applications Within Healthcare: A Systematic Review of Precision Medicine and Dynamic Treatment Regimes. *Healthcare* 13, 14 (2025), 1752. doi:10.3390/healthcare13141752

[14] Scott Fujimoto, David Meger, and Doina Precup. 2019. Off-Policy Deep Reinforcement Learning without Exploration. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*. 2052–2062. doi:10.48550/arXiv.1812.02900

[15] Pierre Geurts, Damien Ernst, and Louis Wehenkel. 2006. Extremely Randomized Trees. *Machine Learning* 63, 1 (2006), 3–42. doi:10.1007/s10994-006-6226-1

[16] Sam Greydanus, Anurag Koul, Jonathan Dodge, and Alan Fern. 2018. Visualizing and Understanding Atari Agents. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*. 1792–1801. doi:10.48550/arXiv.1711.00138

[17] Hong Han, Xiaoling Guo, and Hua Yu. 2016. Variable Selection Using Mean Decrease Accuracy and Mean Decrease Gini Based on Random Forest. In *Proceedings of the 7th IEEE International Conference on Software Engineering and Service Science (ICSESS)*. 219–224.

[18] Mason Hargrave, Alex Spaeth, and Logan Grosenick. 2024. EpiCare: A Reinforcement Learning Benchmark for Dynamic Treatment Regimes. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 37. 130536–130568.

[19] Rahul Iyer, Yuezhang Li, Huao Li, Michael Lewis, Ramitha Sundar, and Katia Sycara. 2018. Transparency and Explanation in Deep Reinforcement Learning Neural Networks. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (AIES)*. 144–150. doi:10.1145/3278721.3278776

[20] Michael Janner, Qiyang Li, and Sergey Levine. 2021. Offline Reinforcement Learning as One Big Sequence Modeling Problem. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 34. 1273–1286.

[21] Nan Jiang and Lihong Li. 2016. Doubly Robust Off-policy Value Evaluation for Reinforcement Learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, Vol. 48. 652–661. doi:10.48550/arXiv.1511.03722

[22] Zoe Juozapaitis, Anurag Koul, Alan Fern, Martin Erwig, and Finale Doshi-Velez. 2019. Explainable Reinforcement Learning via Reward Decomposition. In *Proceedings of the Pacific Rim International Conference on Artificial Intelligence (PRICAI)*. 87–98.

[23] Kia Khezeli, Scott Siegel, Benjamin Shickel, Tezcan Ozrazgat-Baslanti, Azra Bihorac, and Parisa Rashidi. 2023. Reinforcement Learning for Clinical Applications. *Clinical Journal of the American Society of Nephrology* 18, 4 (2023), 521–523. doi:10.2215/CJN.0000000000000084

[24] Taylor W. Killian, Sonali Parbhoo, and Marzyeh Ghassemi. 2023. Risk Sensitive Dead-end Identification in Safety-Critical Offline Reinforcement Learning. *arXiv preprint arXiv:2301.05664* (2023).

[25] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. 2020. Conservative Q-Learning for Offline Reinforcement Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 33. 1179–1191.

[26] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. 2020. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. *arXiv preprint arXiv:2005.01643* (2020).

[27] Ning Liu, Ying Liu, Brent Logan, Zhiyuan Xu, Jian Tang, and Yanzhi Wang. 2018. Deep Reinforcement Learning for Dynamic Treatment Regimes on Medical Registry Data. *arXiv preprint arXiv:1801.09271* (2018).

[28] Scott M Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 30. 4765–4774.

[29] Zhiyao Luo, Yangchen Pan, Peter Watkinson, and Tingting Zhu. 2024. Reinforcement Learning in Dynamic Treatment Regimes Needs Critical Reexamination. *arXiv preprint arXiv:2405.18556* (2024).

[30] Zhiyao Luo, Mingcheng Zhu, Fenglin Liu, Jiali Li, Yangchen Pan, Jiandong Zhou, and Tingting Zhu. 2024. DTR-Bench: An in silico Environment and Benchmark Platform for Reinforcement Learning Based Dynamic Treatment Regime. *arXiv preprint arXiv:2405.18610* (2024).

[31] Prashan Madumal, Tim Miller, Liz Sonenberg, and Frank Vetere. 2020. Explainable Reinforcement Learning Through a Causal Lens. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 2493–2500.

[32] S. A. Murphy. 2003. Optimal Dynamic Treatment Regimes. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 65, 2 (2003), 331–355. doi:10.1111/1467-9868.00389

[33] Erika Puiutta and Eric MSP Veith. 2020. Explainable Reinforcement Learning: A Survey. *arXiv preprint arXiv:2005.06247* (2020).

[34] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 1135–1144. doi:10.1145/2939672.2939778

[35] Zahra Sadeghi, Roohallah Alizadehsani, Mehmet Akif CIFCI, Samina Kausar, Rizwan Rehman, Priyakshi Mahanta, Pranjal Kumar Bora, Ammar Almasri, Rami S. Alkhawaldeh, Sadiq Hussain, Bilal Alatas, Afshin Shoeibi, Hossein Moosaei, Milan Hladík, Saeid Nahavandi, and Panos M. Pardalos. 2024. A Review of Explainable Artificial Intelligence in Healthcare. *Computers and Electrical Engineering* 118 (2024), 109370. doi:10.1016/j.compeleceng.2024.109370

[36] Jürgen Schmidhuber. 2019. Reinforcement Learning Upside Down: Don't Predict Rewards—Just Map Them to Actions. *arXiv preprint arXiv:1912.02875* (2019).

[37] Rupesh Kumar Srivastava, Pranav Shyam, Filipe Mutz, Wojciech Jaskowski, and Jürgen Schmidhuber. 2019. Training Agents using Upside-Down Reinforcement Learning. *arXiv preprint arXiv:1912.02877* (2019).

[38] Miroslav Štrupl, Francesco Faccio, Dylan R. Ashley, Jürgen Schmidhuber, and Rupesh Kumar Srivastava. 2022. Upside-Down Reinforcement Learning Can Diverge in Stochastic Environments With Episodic Resets. *arXiv preprint arXiv:2205.06595* (2022).

[39] Shengpu Tang and Jenna Wiens. 2021. Model Selection for Offline Reinforcement Learning: Practical Considerations for Healthcare Settings. *arXiv preprint arXiv:2107.11003* (2021).

[40] Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U. Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Hannah Tan, and Omar G. Younis. 2024. Gymnasium: A Standard Interface for Reinforcement Learning Environments. *arXiv preprint arXiv:2407.17032* (2024).

[41] Chunyu Wang and Brian DM Tom. 2025. A Tutorial on Optimal Dynamic Treatment Regimes. *arXiv preprint arXiv:2502.16988* (2025).

[42] Chao-Han Huck Yang, Zhengling Qi, Yifan Cui, and Pin-Yu Chen. 2021. Pessimistic Model Selection for Offline Deep Reinforcement Learning. *arXiv preprint arXiv:2111.14346* (2021).