CardioSim: Gymnasium Environments for Reinforcement Learning in Cardiac Electrophysiology

Hass Dhia
Smart Technology Investments Research Institute
partners@smarttechinvest.com

March 2026

Abstract

Reinforcement learning has demonstrated promise in sequential medical decision-making, yet standardized benchmarks for cardiac electrophysiology remain absent from the Gymnasium ecosystem. We present CardioSim, a Python package providing three Gymnasium-compatible environments for cardiac control: pacemaker rate optimization, antiarrhythmic drug dosing, and defibrillation timing. CardioSim integrates the FitzHugh-Nagumo and Aliev-Panfilov action potential models, a single-compartment pharmacokinetic/pharmacodynamic model, and a cardiac conduction system simulator. Each environment supports configurable difficulty tiers (easy, medium, hard) and ships with random, heuristic, and PPO baseline agents. PPO agents trained for 200,000–500,000 timesteps learn positive-reward policies on drug dosing (mean reward +129 vs. random −31), eliminate catastrophic failures on pacing control (−646 vs. random −369,322), while heuristic agents outperform PPO on both tasks. Defibrillation timing remains unsolved: PPO does not exceed random baselines on easy difficulty. These results establish a spectrum from learnable to open-challenge environments. The package includes 134 unit tests, is installable via PyPI, and is released under the MIT license to facilitate reproducible RL research in cardiology.

1 Introduction

Cardiac electrophysiology encompasses the study of electrical activity in the heart, including action potential generation, propagation, and the pathological states that arise when these processes fail. Clinical interventions such as pacemaker programming, antiarrhythmic drug titration, and defibrillation timing represent sequential decision-making problems where reinforcement learning (RL) may offer advantages over static protocols [McNamara et al., 2025, Jiang et al., 2024].

Despite growing interest in RL for healthcare [Liu et al., 2022, Tosca et al., 2024], the field lacks standardized, open-source simulation environments specifically designed for cardiac electrophysiology benchmarking. Existing cardiac simulators either focus on high-fidelity biophysical modeling without RL interfaces, or provide simplified toy problems that do not capture clinically relevant dynamics.

CardioSim addresses this gap by providing three Gymnasium-compatible [Towers et al., 2024] environments that balance physiological realism with computational tractability. Each environment wraps established cardiac models – the FitzHugh-Nagumo [FitzHugh, 1961, Nagumo et al., 1962] and Aliev-Panfilov [Aliev and Panfilov, 1996] models – with standardized observation and action spaces, shaped reward functions, and configurable difficulty tiers.

Our contributions are:

1. Three Gymnasium environments spanning distinct cardiac control paradigms: rate control, pharmacological intervention, and emergency defibrillation.
2. Integration of two established cardiac action potential models with a pharmacokinetic/pharmacodynamic (PK/PD) model and a cardiac conduction system simulator.
3. Baseline agents (random, heuristic, PPO) with documented performance across all environments and difficulty tiers.
4. An open-source Python package with 134 tests, available on PyPI for reproducible research.

2 Background

2.1 Cardiac Action Potential Models

The FitzHugh-Nagumo (FHN) model [FitzHugh, 1961] reduces the Hodgkin-Huxley equations [Hodgkin and Huxley, 1952] to a two-variable system capturing excitability:

$$\frac{dv}{dt} = v - \frac{v^3}{3} - w + I_{\text{ext}} \tag{1}$$

$$\frac{dw}{dt} = \epsilon(v + a - bw) \tag{2}$$

where v is the membrane potential, w is the recovery variable, I_{ext} is external stimulus current, and ϵ, a, b are model parameters.

The Aliev-Panfilov (AP) model [Aliev and Panfilov, 1996] provides cardiac-specific action potential dynamics:

$$\frac{du}{dt} = -ku(u-a)(u-1) - uv + I_{\text{ext}} \tag{3}$$

$$\frac{dv}{dt} = \left(\epsilon_0 + \frac{\mu_1 v}{\mu_2 + u}\right)(-v - ku(u-a-1)) \tag{4}$$

where u represents normalized membrane potential, v is the recovery variable, and k, a, b, ϵ_0, μ_1, μ_2 are tissue-specific parameters.

2.2 Pharmacokinetic Modeling

The single-compartment PK model describes drug concentration dynamics with first-order absorption:

$$\frac{dA_{\text{gut}}}{dt} = -k_a A_{\text{gut}} + D \tag{5}$$

$$\frac{dC}{dt} = \frac{k_a A_{\text{gut}}}{V_d} - k_e C \tag{6}$$

where A_{gut} is the gut compartment amount, C is plasma concentration, D is the administered dose, V_d is the volume of distribution, k_a is the absorption rate constant, and k_e is the elimination rate constant. Pharmacodynamic effects follow a sigmoid $E_{\max}$ model:

$$E = E_{\max}\frac{C^n}{EC_{50}^n + C^n} \tag{7}$$

2.3 Reinforcement Learning

We formulate each cardiac control task as a Markov Decision Process (MDP) $\langle S, A, T, R, \gamma\rangle$. At each timestep, the agent observes state $s \in S$, selects action $a \in A$, receives reward $r = R(s, a)$, and transitions to $s' \sim T(s'|s, a)$. We use Proximal Policy Optimization [Schulman et al., 2017] via Stable-Baselines3 [Raffin et al., 2021] as our primary RL algorithm.

3 Related Work

Reinforcement learning for cardiac device programming has gained attention with recent work on reward machine learning from demonstrations for pacemaker synthesis [Jiang et al., 2024]. McNamara et al. [McNamara et al., 2025] outline strategies for safe RL in cardiac implantable electronic devices, emphasizing the need for simulation-based training before clinical deployment.

In drug dosing, Tosca et al. [Tosca et al., 2024] demonstrate model-informed RL for precision dosing using PK/PD models, while Liu et al. [Liu et al., 2022] apply deep RL to warfarin dosing optimization. Hajialigol and Alikhani [Hajialigol and Alikhani, 2019] develop optimal adaptive control of drug dosing using integral reinforcement learning. Computational studies of antiarrhythmic mechanisms [Dutta et al., 2024] and ML-based dofetilide dose management [Levy et al., 2019] further establish the role of computational methods in cardiac pharmacology.

For defibrillation, machine learning approaches to predicting shock success have been systematically reviewed [Isasi et al., 2023], and AI's role in defibrillators has been surveyed [Haddadian et al., 2022, Krishnamurti and Mrutu, 2025]. However, none of these works provide standardized RL benchmarking environments.

The Gymnasium framework [Towers et al., 2024, Brockman et al., 2016] has become the standard interface for RL environments, yet cardiac electrophysiology remains unrepresented. CardioSim fills this gap by combining established biophysical models with the Gymnasium API, enabling direct comparison of RL algorithms on clinically motivated tasks.

4 Environments

CardioSim provides three environments, each targeting a distinct cardiac control paradigm. All environments follow the Gymnasium API with continuous observation and action spaces.

4.1 PacingControl-v0

Clinical motivation: Optimize pacemaker settings to restore normal sinus rhythm in bradycardia patients.

Observation space: $\mathbb{R}^6$ – membrane potential v, recovery variable w, current heart rate, target heart rate, heart rate error, and heart rate variability (HRV).

Action space: $\mathbb{R}^2$ – pacing rate adjustment and pacing amplitude.

Dynamics: The FHN model generates cardiac action potentials. A conduction system model (SA node, AV node) converts action potentials to heart rate. The agent adjusts pacing parameters to minimize the error between current and target heart rate while maintaining physiological HRV.

Reward: $r = -\alpha|HR - HR_{\text{target}}| - \beta|HRV - HRV_{\text{target}}| + \gamma \cdot \mathbb{1}[|HR - HR_{\text{target}}| < \delta]$

4.2 AntiarrhythmicDosing-v0

Clinical motivation: Maintain therapeutic drug concentration while suppressing arrhythmia and avoiding toxicity.

Observation space: $\mathbb{R}^6$ – membrane potential, recovery variable, drug concentration, arrhythmia indicator, drug efficacy, and normalized time.

Action space: $\mathbb{R}^1$ – drug dose (mg).

Dynamics: The FHN model simulates cardiac dynamics with drug-modulated parameters. A single-compartment PK model tracks drug concentration. The PD model modulates action potential parameters based on drug effect.

Reward: $r = r_{\text{therapeutic}} + r_{\text{arrhythmia}} + r_{\text{toxicity}}$, where agents are rewarded for maintaining concentration within the therapeutic window, penalized for arrhythmia persistence, and heavily penalized for toxic concentrations.

4.3 DefibrillationTiming-v0

Clinical motivation: Terminate ventricular fibrillation with minimal shock energy and count.

Observation space: $\mathbb{R}^6$ – membrane potential, recovery variable, fibrillation index, time in fibrillation, shock count, and cumulative energy.

Action space: $\mathbb{R}^2$ – shock decision (continuous, thresholded at 0.5) and shock energy (Joules).

Dynamics: The Aliev-Panfilov model operates in a chaotic regime simulating fibrillation. Shock delivery resets the membrane state probabilistically, with success probability increasing with energy and depending on the fibrillation phase.

Reward: $r = r_{\text{termination}} - \lambda_1 \cdot E_{\text{shock}} - \lambda_2 \cdot N_{\text{shocks}} - \lambda_3 \cdot t_{\text{fib}}$

4.4 Difficulty Tiers

Each environment supports three difficulty tiers that modify noise levels, parameter variability, episode length, and tolerance thresholds. Easy tier uses deterministic dynamics with wide tolerances; hard tier introduces stochastic parameter variation, observation noise, and narrow success criteria.

5 Implementation

CardioSim is implemented in Python with NumPy for numerical integration (custom fourth-order Runge-Kutta). The package follows a modular architecture:

- `cardiosim.models` – FHN, AP, PK/PD, and conduction system models with validated parameter ranges.
- `cardiosim.envs` – Three Gymnasium environments with difficulty tier support.
- `cardiosim.agents` – Random, heuristic, and PPO baseline agents.
- `cardiosim.training` – Evaluation utilities and reward ratio computation.
- `cardiosim.benchmarks` – Benchmark configuration and runner.

All domain models include `PARAMETER_RANGES` dictionaries with literature-backed value ranges, units, and source citations. Runtime validation enforces these ranges.

6 Baseline Agents

6.1 Random Agent

Samples uniformly from the action space at each timestep. Serves as the minimum performance baseline.

6.2 Heuristic Agents

HeuristicPacingAgent: Implements proportional control on heart rate error with gain scheduling. Adjusts pacing rate proportionally to the difference between current and target heart rate.

HeuristicDosingAgent: Implements rule-based dosing with loading dose, maintenance dose, and toxicity withholding logic based on drug concentration thresholds.

No heuristic agent is provided for defibrillation timing, as the optimal policy depends on stochastic fibrillation dynamics that resist simple rule-based approaches.

6.3 PPO Agent

We train PPO agents [Schulman et al., 2017] using Stable-Baselines3 [Raffin et al., 2021] with MLP policies. Training hyperparameters are environment-specific (Table 1).

Table 1: PPO hyperparameters per environment.

Parameter	PacingControl	Dosing	Defibrillation
Learning rate	3×10^{-4}	5×10^{-4}	3×10^{-4}
Steps per update	2048	1024	256
Batch size	64	64	64
Epochs	10	10	10
Entropy coeff.	0.01	0.02	0.10
Discount γ	0.99	0.99	0.99
Total timesteps	200,000	300,000	500,000

7 Results

We evaluate each agent over 20 episodes per environment (Table 2). Performance is measured by mean episodic reward and the ratio of PPO reward to random baseline reward.

Table 2: Baseline agent performance (mean ± std over 20 episodes).

Environment	Random	Heuristic	PPO
PacingControl	-369322 ± 1545175	129 ± 6	-646 ± 284
Dosing	-31 ± 5	169 ± 16	129 ± 32
Defibrillation	136 ± 22	–	128 ± 27

The results reveal a clear difficulty gradient across environments. On pacing control, PPO eliminates the catastrophic over-pacing failures seen in the random baseline (mean reward -646 vs. $-369{,}322$), but the heuristic proportional controller achieves substantially better performance ($+129$), suggesting that the smooth reward landscape favors classical control over model-free RL at this training budget. On antiarrhythmic dosing, PPO learns a positive-reward dosing policy ($+129$) compared to the random baseline (-31), though the heuristic rule-based agent again outperforms ($+169$), indicating room for improvement in PPO's handling of delayed pharmacokinetic effects.

The defibrillation timing environment, trained on easy difficulty ($p_{\text{success}} = 0.7$ base), presents the hardest challenge: PPO scores 128 compared to random's 136, failing to outperform random after 500,000 timesteps. The stochastic success dynamics mean that high-frequency random shocking is a competitive strategy, and PPO's learned caution about shock delivery is sometimes counterproductive. This environment remains an open challenge for RL algorithms.

Across all three environments, heuristic agents match or exceed PPO, suggesting these benchmarks are nontrivial for model-free RL while remaining solvable by domain-informed methods – a desirable property that rewards algorithm development incorporating domain structure.

8 Discussion

CardioSim demonstrates that simplified cardiac electrophysiology models can serve as effective RL benchmarks while maintaining physiological relevance. The FitzHugh-Nagumo and Aliev-

Panfilov models capture essential excitability dynamics without the computational cost of full ionic models (e.g., ten Tusscher-Panfilov).

Limitations. CardioSim uses reduced-order models that omit spatial propagation, three-dimensional geometry, and patient-specific parameterization. The pharmacokinetic model is single-compartment, lacking tissue distribution dynamics. These simplifications are deliberate – the goal is benchmarking RL algorithms, not clinical simulation. Our PPO baselines use a single set of hyperparameters per environment without sensitivity analysis, and defibrillation results are reported on easy difficulty; results on medium and hard tiers may differ substantially. Evaluation over 20 episodes with high variance (particularly on pacing control) limits statistical confidence in precise reward comparisons.

Future work. Extensions may include multi-lead ECG observation spaces, patient population generators for robust policy evaluation, and integration with higher-fidelity models (e.g., OpenCARP) for transfer learning studies.

9 Reproducibility

All code, trained models, and experimental results are available at `https://github.com/HassDhia/cardiosim`. The package is installable via `pip install cardiosim` and includes 134 automated tests. Training scripts reproduce all reported results with fixed random seeds.

10 Conclusion

We present CardioSim, an open-source Python package providing three Gymnasium-compatible reinforcement learning environments for cardiac electrophysiology. By integrating established biophysical models with the standard RL interface, CardioSim enables reproducible benchmarking of RL algorithms on clinically motivated cardiac control tasks. Our baseline results show that the three environments span a difficulty spectrum: PPO partially learns drug dosing and pacing control but does not match heuristic baselines, while defibrillation timing remains unsolved by model-free RL. This range from learnable to open-challenge makes CardioSim a useful benchmark suite for developing RL algorithms that incorporate domain structure.

References

Rubin R Aliev and Alexander V Panfilov. A simple two-variable model of cardiac excitation. *Chaos, Solitons & Fractals*, 7(3):293–301, 1996.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. *arXiv preprint arXiv:1606.01540*, 2016.

Sara Dutta et al. A simulation study on the antiarrhythmic mechanisms of established agents in myocardial ischemia and infarction. *PLOS Computational Biology*, 2024.

Richard FitzHugh. Impulses and physiological states in theoretical models of nerve membrane. *Biophysical Journal*, 1(6):445–466, 1961.

Behrouz Haddadian et al. Role of artificial intelligence in defibrillators: a narrative review. *Current Problems in Cardiology*, 47(11):101349, 2022.

Nooshin Hajialigol and Mostafa Alikhani. Optimal adaptive control of drug dosing using integral reinforcement learning. *Mathematical Biosciences*, 309:131–142, 2019.

Alan L Hodgkin and Andrew F Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117(4):500–544, 1952.

Iraia Isasi et al. The ability of machine learning algorithms to predict defibrillation success during cardiac arrest: A systematic review. *Resuscitation*, 185:109740, 2023.

John Jiang, Daniel Neider, and Ufuk Topcu. Show, don't tell: Learning reward machines from demonstrations for reinforcement learning-based cardiac pacemaker synthesis. *arXiv preprint arXiv:2411.01750*, 2024.

Chandrasekhar Krishnamurti and Salehe Mrutu. Artificial intelligence in cardiac emergencies: A review. *Global Heart Failure*, 2025.

Andrew E Levy et al. Applications of machine learning in decision analysis for dose management for dofetilide. In *PLOS ONE*, volume 14, page e0227324, 2019.

Yawei Liu et al. Optimizing warfarin dosing using deep reinforcement learning. *Journal of Biomedical Informatics*, 137:104267, 2022.

Conor McNamara et al. Strategies and considerations for safe reinforcement learning in programming cardiac implantable electronic devices. *PMC*, 2025.

Jin-ichi Nagumo, Suguru Arimoto, and Shuji Yoshizawa. An active pulse transmission line simulating nerve axon. *Proceedings of the IRE*, 50(10):2061–2070, 1962.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Elisa M Tosca et al. Model-informed reinforcement learning for enabling precision dosing via adaptive dosing. *Clinical Pharmacology & Therapeutics*, 2024.

Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Arjun KG, Markus Krimmel, et al. Gymnasium: A standard interface for reinforcement learning environments, 2024.