Reinforcement Learning for Optimal Control of Adaptive Cell Populations

Josiah C. Kratz*
Computational Biology Department
Carnegie Mellon University
Pittsburgh, PA 15213
jkratz@andrew.cmu.edu

Jacob Adamczyk*
Department of Physics
University of Massachusetts Boston
IAIFI
Boston, MA 02125
jacob.adamczyk001@umb.edu

Abstract

Many organisms and cell types, from bacteria to cancer cells, exhibit a remarkable ability to adapt to fluctuating environments. Additionally, cells can leverage memory of past environments to better survive previously-encountered stressors. From a control perspective, this adaptability poses significant challenges in driving cell populations toward extinction, and is thus an open question with great clinical significance. In this work, we focus on drug dosing in cell populations exhibiting phenotypic plasticity. For specific dynamical models switching between resistant and susceptible states, exact solutions are known. However, when the underlying system parameters are unknown, and for complex memory-based systems, obtaining the optimal solution is currently intractable. To address this challenge, we apply reinforcement learning (RL) to identify informed dosing strategies to control cell populations evolving under novel non-Markovian dynamics. We find that model-free deep RL is able to recover exact solutions and control cell populations even in the presence of long-range temporal dynamics.

1 Introduction

In order to survive, organisms must adapt to unpredictable environmental stressors occurring over diverse timescales. As a result, biological systems display remarkable adaptive capabilities, making it exceptionally challenging to control them through environmental modulation—an open and significant question in the physics of living systems. Two examples of adaptive systems with great clinical importance are cancer cell resistance to chemotherapy [34], and bacterial resistance to antibiotics [4]. In both settings, constant application of a drug does not typically result in population extinction, as drug application also drives a certain fraction of the population to alter their phenotypic state to become drug-resistant. This new resistant state can persist even after drug removal and across cell lineages, thus encoding a memory of the stressful environment which allows populations to more quickly adapt if the drug is reapplied [13, 21, 3].

Previous work [25, 8, 10, 9] has studied temporal drug dosing protocols to slow or prevent adaptation in various cancer or bacterial models using different methods. However, these models fail to account for the variety of adaptive timescales present in real biological systems. The presence of such memory effects greatly complicates the control problem, making the study of more realistic models imperative. Thus, to study the control of such memory-based adaptive systems, in this work we introduce a novel population model exhibiting phenotypic plasticity with non-Markovian dynamics. We couple insights from control theory with deep reinforcement learning to discover interpretable treatment protocols which successfully prevent proliferation.

*Equal contribution. Author ordering determined by coin flip at Primanti Bros.

Machine Learning and the Physical Sciences Workshop, NeurIPS 2024.

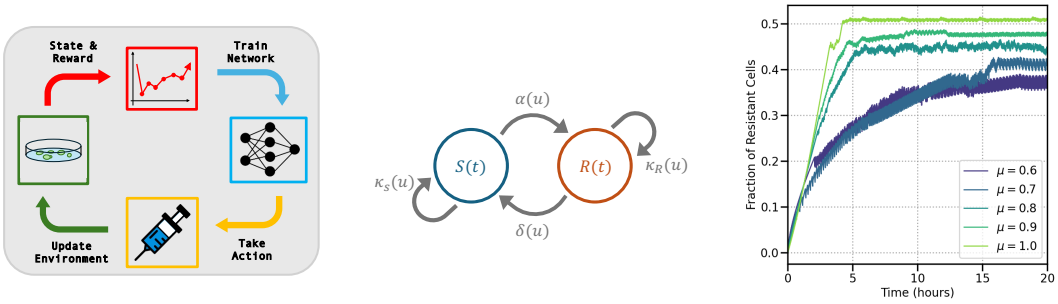

Figure 1: Left: Interaction loop between RL agent and environment (Sec. 2.2). Middle: depiction of the phenotypic switching model (Sec. 2.1). Right: Effect of learned policy on resistant fraction. for different memory strengths.

2 Proposed Model and Approach

2.1 Non-Markovian Phenotypic Switching Model

To model the treatment response of an adaptive cell population, we use a general phenotypic switching model which captures the time evolution of a susceptible subpopulation, with size $S(t)$, and a resistant subpopulation, with size $R(t)$. Phenotypic switching models have been successful in describing a wide variety of biological scenarios, including development of persister cells and resistant cells in bacteria, and development of drug resistance in cancer cells [2, 9, 35, 18, 17]. In our model, susceptible cells with a net growth rate $\kappa_S(u)$ ($\kappa_S(u) < 0$ corresponds to net cell death) switch to a resistant state at a rate $\alpha(u)$, where u is the drug concentration. Similarly, when the drug is removed, resistant cells with a net growth rate $\kappa_R(u)$ switch back to the susceptible state at a concentration-dependent rate $\delta(u)$ (Fig. 1, middle). All growth and switching rates are a function of drug concentration normalized by the maximum allowable dose, thus $u \in [0, 1]$. The time evolution of the size of each subpopulation, $\mathbf{x}(t) = [S(t), R(t)]^T$, is then given by the dynamical system:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), u(t)) = \mathbf{A}(u(t))\mathbf{x}(t) \tag{1}$$

where the state-transition matrix is given by:

$$\mathbf{A}(u) = \begin{bmatrix} \kappa_S(u) - \alpha(u) & \delta(u) \\ \alpha(u) & \kappa_R(u) - \delta(u) \end{bmatrix} . \tag{2}$$

We choose $\kappa_S(u)$ and $\kappa_R(u)$ to decrease and increase with drug concentration, respectively. This parametrization corresponds to "drug addiction" behavior in the resistant subpopulation, a robust phenomenon which has been observed not only in cell culture [29, 30, 23], but in animal models [6] and *in vivo* [27, 7]. Similarly, $\delta(u)$ decreases with u while $\alpha(u)$ increases with u, as drug application drives the population to become more resistant, while reduction in drug concentration causes the system to recover susceptibility (see Appendix A for complete model details).

Recently, cell populations of many types, including human cancer cell lines, yeast, and bacteria, have been shown to maintain a memory of past environments which facilitates adaptation to previously-seen stressors over many timescales [28, 13, 19, 36, 21]. To better capture this memory dependence on treatment response, we introduce a memory kernel into the previously-described dynamics (Eq. (1)), making them non-local in time. Specifically, we choose a fractional differential equation (FDE) formulation as a phenomenological way to introduce multiple timescales of adaptation, one which has been used successfully to model memory effects in other biological [20], ecological [15], and physical contexts [5]. With this addition the dynamics now become:

$$\dot{\mathbf{x}}(t) = \boldsymbol{F}(\mathbf{x}(t), u(t)) = \int_0^t \frac{(t-\tau)^{\mu-2}}{|\Gamma(\mu-1)|} \mathbf{f}(\mathbf{x}(\tau), u(\tau)) d\tau \, , \tag{3}$$

where $\Gamma(\cdot)$ denotes the Gamma function, $\mathbf{f}(\cdot)$ is given by Eq. (1), and here we introduce the parameter $\mu \in (0, 1]$ which controls memory strength. A value of $\mu = 1$ corresponds to the memoryless case (first order derivative), whereas smaller values of μ correspond to an increased influence of past states on the current dynamics (lower order fractional derivative).

We seek to obtain a temporal drug concentration protocol $u(t)$ which minimizes the growth of a population and thus choose the final cost as $C := \log N(T)/N(0)$ over the time interval T, where $N(t) = S(t) + R(t)$ represents the total population. Thus, we aim to solve the following control problem:

$$\min_u C(\mathbf{x}(0), \mathbf{x}(T; u(\cdot))) \text{ subject to } \dot{\mathbf{x}}(t) = \boldsymbol{F}(\mathbf{x}(t), u(t)),\ \mathbf{x}(0) = \mathbf{x}_0,\ u(t) \in [0, 1]\ , \tag{4}$$

where $\mathbf{x}(T; u(\cdot))$ denotes the state of $\mathbf{x}$ at terminal time T subject to control u from $0 \leq t \leq T$. Interestingly, in our minimal model (Eq. (3)), as long as $\kappa_S(u)$, $\kappa_R(u)$, $\alpha(u)$, and $\delta(u)$ are monotonic functions, we can show that the optimal control solution follows "bang-bang" control, regardless of model parameters and memory strength (see Appendix A for details). Thus, the continuous state-transition matrix of Eq. (2) can be simplified into a discrete one with binary control inputs, without altering the optimal solution (Appendix A). Despite this simplification, model control remains difficult, as the number of times and duration of drug application must be optimized. Constant application of the drug at the maximum dose ($u = 1$) results in cell adaptation and proliferation (Fig. 2): a highly suboptimal solution to Problem (4) and a catastrophic result in the clinical context. Previous work [9] has shown that in the memoryless case ($\mu = 1$), the optimal solution for this type of model requires an initial drug application phase, followed by pulsing between treatment and pause phases at a regular interval dependent on the model parameters. However, as seen in Fig. 2, we find that the addition of memory ($\mu < 1$) renders the control strategy of the memoryless case ineffective, as cells which have previously encountered treatment switch to the resistant state faster upon subsequent applications. Furthermore, in clinical or experimental setting, obtaining the values which parameterize Eq. (3) is usually not feasible, thus obtaining the appropriate switching frequency through direct computation via optimal control (OC) theory becomes impossible. As a result, we seek to learn the optimal policy directly through experience using reinforcement learning.

2.2 Reinforcement Learning

Reinforcement learning (RL) allows an agent to learn a drug protocol through experience (Fig. 1, left), despite the non-Markovian dynamics and without access to the underlying environment-specific model parameters. To formulate the RL problem, we define the relevant characteristics as follows: **State**: The state (s_t) of the agent's environment is a list of the last $K = 5$ estimates of the instantaneous growth rate, $c_t = \Delta^{-1} \log N_t / N_{t-\Delta}$, where Δ is the simulation time between actions. Thus, a history of past observations is encoded in the state vector, a crucial design choice if the agent is to learn control without a recurrent hidden state. **Action**: As motivated in 2.1, we choose a binary action space $u \in \{0, 1\}$ representing whether the drug is applied or not, as this is suitable to recover optimal control. **Reward**: As we seek to solve Problem (4), the reward is simply the negative growth rate, $r_t = -c_t$. Notice that with this choice of reward function, the sum of rewards across a trajectory simplifies as $R_{0:T} = \Delta^{-1} \log N_0 / N_T$, ensuring the agent's objective is aligned with a reduction in total cell population. **Dynamics:** The dynamics of the total cell population N_t are governed by the dynamical system described in 2.1, initialized to be fully susceptible ($\mathbf{x}_0 = [1000, 0]$). After an action is executed, the simulation (a numerical solution [2] of Eq. (3)) is computed with the action (dose) fixed for $\Delta = 0.01$ hours to compute the next state ($s_{t+\Delta}$).

We use our own implementation of Double DQN [33] based on open-source code for DQN [26] to train the agent in this environment. After an action is taken, it is stored in an experience replay buffer for later use. We parameterize the value function Q with a neural network (with parameters θ), and train via SGD by sampling mini-batches uniformly at random from the buffer. The loss function is defined as the Bellman residual loss – the squared difference between left and right hand sides of the following equation:

$$Q^*(s_t, u_t; \theta) = r(s_t, u_t) + \gamma \max_{u'} Q^*(s_{t+\Delta}, u'; \theta)\ . \tag{5}$$

As common in value-based algorithms, we use an additional target network and use exponentially annealed ε-greedy exploration. Further details on reinforcement learning can be found e.g. in [31, 14]. We tune over several hyperparameters (whose values we list in the Appendix) to optimize performance. To encourage the agent to reduce the cell population while decreasing runtime, we terminate the episode if the number of cells ever exceeds its initial amount, forcing the agent to initially apply a continuous dose. All code to reproduce our experimental results can be found at https://github.com/JacobHA/RL4Dosing.

[2] The numerical solution of fractional differential equations requires some care; cf. [11] for details.

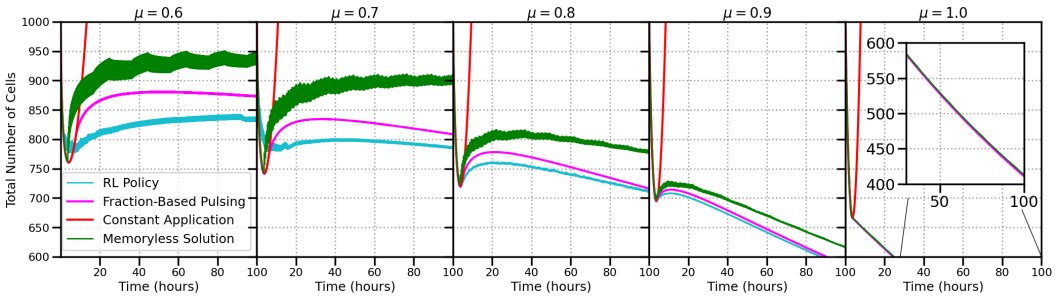

Figure 2: Performance comparison of constant drug application, solution for the memoryless case, resistant fraction-based pulsing technique, and policy learned by RL. For the fraction-based policy, an optimal lower and upper bound for resistant fractions are found through sweeping (Appendix A.1). The RL policy is capable of controlling the cell population better than any other scheme.

3 Results

We first test DQN in the memoryless case ($\mu = 1$), for which an optimal controller is known [9]. In this case, the optimal policy can be derived based only on two consecutive resistant fractions. However, given only the growth rates, the RL agent is able to reliably recover the optimal policy with only two frames and without any access to underlying model parameters. It does so by first applying the drug and then pulsing regularly, in agreement with the OC solution, as seen in Fig. 2 (rightmost plot). With confirmation that RL can find the optimal dosing strategy in the memoryless case, we turn to the more difficult memory-based dynamics ($\mu < 1$). We do not have a solution to Problem 4 in this regime, so we compare to two baselines: The memoryless protocol and a modified version in which switching times occur when the resistant fraction reaches a threshold value. We note that these baselines can have arbitrarily small switching times, which is practically infeasible. Remarkably, we find that using a small but bounded Δ, our learned policy outperforms both of these baselines (Fig. 2). In addition, our experiments show that decreasing Δ beyond a certain threshold does not considerably increase performance (Fig. 4). Interestingly, the learned policy reveals that the agent initially maintains constant drug application before transitioning to a memory-dependent pulsing protocol. The agent increases the frequency of pulsing throughout the trajectory until saturation at the maximum rate (Δ^{-1}) (Fig. 3). For cases with memory, this increase in dosing frequency can be understood as the result of faster cellular adaptation back to the resistant state after multiple drug encounters. Thus, to maintain a susceptible population, the policy's pulse frequency must continuously be increased to compensate for the memory-based adaptability. We also find that the agent maintains a lower average resistant fraction at higher memory values (Fig. 1, right).

4 Discussion

In this work, we study the control of a highly non-Markovian model of adaptive cellular growth dynamics using deep reinforcement learning. Although we focus here on a specific parameterization most relevant to cancer, we expect this methodology to be applied successfully to other scenarios, including resistance development in bacteria. We find that deep RL is capable of recovering the known optimal policy for the memoryless case and can successfully find a policy for memory-based systems which prevents proliferation, all without access to the underlying model parameters. We utilize frame-stacking [22] to encapsulate the history of the agent's trajectory, but in future work we will test the use of recurrent policies to capture more nuanced long-term effects and perhaps further improve performance. We also plan to extend our deterministic framework to the stochastic setting, as population heterogeneity is known to further complicate the control process. This work demonstrates the benefits of combining OC and RL, showing how their frameworks can be effectively integrated. This connection can be further developed by using the memoryless OC solution to enhance RL training through reward shaping [24, 1] or pre-training [32] techniques.

Experimentally measuring the growth rate of a pathogenic population is easier than determining its resistant fraction. Surprisingly, we find that RL can learn successful policies directly from the growth rate making it a promising method for use in clinical settings.

5 Acknowledgements

JA acknowledges funding support from the Topical Group on Data Science (GDS) of the American Physical Society; the NSF through Award No. PHY-2425180; the use of the supercomputing facilities managed by the Research Computing Department at UMass Boston; and fruitful discussions with Rahul V. Kulkarni and Stas Tiomkin. JCK acknowledges support from the NeurIPS ML4PS organizers, the Computational Biology Department at CMU, and from Shiladitya Banerjee in the Department of Physics at CMU. This project developed out of the hackathon at the 2024 Summer School hosted by the Institute for Artificial Intelligence and Fundamental Interactions (IAIFI).

References

[1] Jacob Adamczyk et al. "Utilizing prior solutions for reward shaping and composition in entropy-regularized reinforcement learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 6. 2023, pp. 6658–6665.

[2] Nathalie Q. Balaban et al. "Bacterial persistence as a phenotypic switch". In: *Science* 305.5690 (Sept. 2004), pp. 1622–1625. ISSN: 00368075.

[3] Shiladitya Banerjee et al. "Mechanical feedback promotes bacterial adaptation to antibiotics". In: *Nature Physics* 17.3 (Jan. 2021), pp. 403–409. ISSN: 1745-2481.

[4] Jessica M.A. Blair et al. "Molecular mechanisms of antibiotic resistance". In: *Nature Reviews Microbiology* 13.1 (2015), pp. 42–51. ISSN: 17401534.

[5] Alessandra Bonfanti et al. "Fractional viscoelastic models for power-law materials". In: *Soft Matter* 16.26 (2020), pp. 6002–6020.

[6] Meghna Das Thakur et al. "Modelling vemurafenib resistance in melanoma reveals a strategy to forestall drug resistance". In: *Nature* 494.7436 (Jan. 2013), pp. 251–255. ISSN: 1476-4687.

[7] Andrew J Dooley et al. "Ongoing Response in BRAF V600E-Mutant Melanoma After Cessation of Intermittent Vemurafenib Therapy: A Case Report". In: *Targeted Oncology* 11 (2016), pp. 557–563.

[8] Dalit Engelhardt. "Dynamic control of stochastic evolution: a deep reinforcement learning approach to adaptively targeting emergent drug resistance". In: *Journal of Machine Learning Research* 21.203 (2020), pp. 1–30.

[9] Matthias M Fischer and Nils Bluethgen. "On minimising tumoural growth under treatment resistance". In: *Journal of Theoretical Biology* 579 (2024), p. 111716.

[10] Kit Gallagher et al. "Mathematical Model-Driven Deep Learning Enables Personalized Adaptive Therapy". In: *Cancer Research* 84.11 (2024), pp. 1929–1941. ISSN: 15387445.

[11] Roberto Garrappa. "Numerical solution of fractional differential equations: A survey and a software tutorial". In: *Mathematics* 6.2 (2018), p. 16.

[12] M. I. Gomoyunov. "On the Relationship Between the Pontryagin Maximum Principle and the Hamilton–Jacobi–Bellman Equation in Optimal Control Problems for Fractional-Order Systems". In: *Differential Equations* 59.11 (2023), pp. 1520–1526. ISSN: 16083083.

[13] Guillaume Harmange et al. "Disrupting cellular memory to overcome drug resistance". In: *Nature Communications* 14 (2023).

[14] Matteo Hessel et al. "Rainbow: Combining improvements in deep reinforcement learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

[15] Moein Khalighi et al. "Quantifying the impact of ecological memory on the dynamics of interacting communities". In: *PLoS Computational Biology* 18.6 (2022), pp. 1–21. ISSN: 15537358.

[16] Diederik Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *International Conference on Learning Representations (ICLR)*. San Diego, CA, USA, 2015.

[17] Josiah C Kratz and Shiladitya Banerjee. "Gene Expression Tradeoffs Determine Bacterial Survival and Adaptation to Antibiotic Stress". In: *PRX Life* 2 (2024), pp. 1–15.

[18] Niraj Kumar et al. "Stochastic modeling of phenotypic switching and chemoresistance in cancer cell populations". In: *Scientific reports* 9.1 (2019), p. 10845.

[19] Ajay Larkin et al. "Mapping the dynamics of epigenetic adaptation in S . pombe during heterochromatin misregulation". In: *Developmental Cell* 59.16 (2024), pp. 2222–2238. ISSN: 1534-5807.

[20] Brian N. Lundstrom et al. “Fractional differentiation by neocortical pyramidal neurons”. In: *Nature Neuroscience* 11.11 (2008), pp. 1335–1342. ISSN: 15461726.

[21] Roland Mathis and Martin Ackermann. “Asymmetric cellular memory in bacteria exposed to antibiotics”. In: *BMC Evolutionary Biology* 17.1 (2017), pp. 1–14. ISSN: 14712148.

[22] Volodymyr Mnih et al. “Human-level control through deep reinforcement learning”. In: *Nature* 518.7540 (2015), pp. 529–533.

[23] Gatien Moriceau et al. “Tunable-Combinatorial Mechanisms of Acquired Resistance Limit the Efficacy of BRAF/MEK Cotargeting but Result in Melanoma Drug Addiction”. In: *Cancer Cell* 27.2 (Feb. 2015), pp. 240–256. ISSN: 1535-6108.

[24] Andrew Y Ng et al. “Policy invariance under reward transformations: Theory and application to reward shaping”. In: *International Conference on Machine Learning*. Vol. 99. 1999, pp. 278–287.

[25] Regina Padmanabhan et al. “Reinforcement learning-based control of drug dosing for cancer chemotherapy treatment”. In: *Mathematical biosciences* 293 (2017), pp. 11–20.

[26] Antonin Raffin et al. “Stable-Baselines3: Reliable Reinforcement Learning Implementations”. In: *Journal of Machine Learning Research* 22.268 (2021), pp. 1–8.

[27] Heike Seifert et al. “Prognostic markers and tumour growth kinetics in melanoma patients progressing on vemurafenib”. In: *Melanoma Research* 26.2 (2016), pp. 138–144.

[28] Sydney M. Shaffer et al. “Memory Sequencing Reveals Heritable Single-Cell Gene Expression Programs Associated with Distinct Cellular Behaviors”. In: *Cell* 182.4 (2020), pp. 947–959. ISSN: 10974172.

[29] Kenichi Suda et al. “Conversion from the “oncogene addiction” to “drug addiction” by intensive inhibition of the EGFR and MET in lung cancer with activating EGFR mutation”. In: *Lung Cancer* 76.3 (2012), pp. 292–299. ISSN: 0169-5002.

[30] Chong Sun et al. “Reversible and adaptive resistance to BRAF(V600E) inhibition in melanoma”. In: *Nature* 508.7494 (Mar. 2014), pp. 118–122. ISSN: 1476-4687.

[31] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[32] Ikechukwu Uchendu et al. “Jump-start reinforcement learning”. In: *International Conference on Machine Learning*. PMLR. 2023, pp. 34556–34583.

[33] Hado Van Hasselt et al. “Deep reinforcement learning with double Q-learning”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 30. 1. 2016.

[34] Neil Vasan et al. “A view on drug resistance in cancer”. In: *Nature* 575.7782 (2019), pp. 299–309. ISSN: 14764687.

[35] Christopher Witzany et al. “The pharmacokinetic-pharmacodynamic modeling framework as a tool to predict drug resistance evolution”. In: *Microbiology* (2023), pp. 1–37.

[36] Denise M. Wolf et al. “Memory in microbes: Quantifying history-dependent behavior in a bacterium”. In: *PLoS ONE* 3.2 (2008). ISSN: 19326203.

A Optimal control of the phenotypic switching model yields a bang-bang control solution.

We seek to find a temporal drug protocol $u(t)$ which solves the control problem

$$\min_u C(\mathbf{x}(0), \mathbf{x}(T; u(\cdot))) \text{ subject to } \dot{\mathbf{x}} = \boldsymbol{F}(\mathbf{x}(t), u(t)),\ \mathbf{x}(0) = \mathbf{x}_0,\ u(t) \in [0, 1]. \tag{6}$$

We choose the terminal cost to be the logarithm of the relative growth of the population throughout the course of treatment, $C(\mathbf{x}(0), \mathbf{x}(T; u(\cdot))) := \log N_T/N_0$, where $\mathbf{x}(T; u(\cdot))$ denotes the state of $\mathbf{x}$ at terminal time T subject to control u from $0 \leq t \leq T$, and where $N(t) = S(t) + R(t)$. Using Eq. (1) and the fact that $N(t) = S(t) + R(t)$, the model dynamics can be rewritten as a single fractional differential equation, namely:

$$D_0^\mu \phi(t) = f(\phi, u) = (\kappa_S(u) - \kappa_R(u))\phi^2 + (\kappa_R(u) - \kappa_S(u) - \delta(u) - \alpha(u))\phi + \alpha(u)\ , \tag{7}$$

where D_0^μ denotes the Caputo fractional derivative of order μ starting at $t = 0$ [11], and here we drop the explicit time dependence for notational clarity. This can equivalently be written as the continuous delay differential equation (we use this form in the main text):

$$\dot{\phi}(t) = \int_0^t \frac{(t-\tau)^{\mu-2}}{|\Gamma(\mu-1)|} f(\phi(\tau), u(\tau)) d\tau\ . \tag{8}$$

To relate the net growth rate to drug concentration for each subpopulation, we assume that both rates take the general form:

$$\kappa_S(u) = \kappa_S^{\max} - (\kappa_S^{\max} - \kappa_S^{\min})g(u)\ ,\quad \kappa_R(u) = \kappa_R^{\min} + (\kappa_R^{\max} - \kappa_R^{\min})g(u)\ , \tag{9}$$

where $g(u) \in [0, 1]$ is a monotonic dose-response function which relates drug dose to net growth rate, $\kappa_S^{\max} > 0$ denotes the maximum growth rate of the susceptible subpopulation in the absence of drug application, and where $\kappa_S^{\min} < 0$ is the maximum death rate of the susceptible subpopulation caused by application of the maximum drug dose ($u = 1$). Importantly, $\kappa_R^{\max} > 0$ denotes the maximum growth rate of the resistant subpopulation, which occurs in the *presence* of the maximum drug dose, and $\kappa_R^{\min} < 0$ corresponds to the maximum death rate of the resistant subpopulation, which occurs when the drug is removed. This parametrization corresponds to "drug addiction" behavior, a phenomenon observed in several types of cancers [29, 30, 23, 27, 6, 7]. Similarly, the switching rates can then be defined as:

$$\alpha(u) = \alpha_{\max} g(u) \text{ and } \delta(u) = \delta_{\max}(1 - g(u))\ , \tag{10}$$

where $\alpha_{\max}, \delta_{\max} > 0$ denote the phenotypic switching rates of cells switching from the susceptible state to the resistant state, and vice versa.

Given these definitions, the Hamiltonian associated with this control problem is then

$$H(\phi(t), u(t), \lambda(t)) = \lambda(t) f(\phi(t), u(t))\ , \tag{11}$$

where the trajectory of the Lagrangian multiplier $\lambda(t)$ is the solution to the costate equation [12]:

$$\lambda(t) = -\frac{\partial_\phi C(\mathbf{x}(0), \mathbf{x}(T; u(\cdot)))}{\Gamma(\mu)(T-t)^{1-\mu}} + \frac{1}{\Gamma(\mu)} \int_t^T \frac{\partial_\phi \lambda(\tau) f(\phi(\tau), u(\tau))}{(\tau - t)^{1-\mu}} \mathrm{d}\tau\ . \tag{12}$$

Applying Pontryagin's minimum principle, we obtain the resulting inequality

$$H(\phi^*(t), u^*(t), \lambda^*(t)) \leq H(\phi^*(t), u(t), \lambda^*(t))\ , \tag{13}$$

which along with Eqs. (7) and (11) can be used to obtain the optimal control policy:

$$u^*(t) = \arg\min_u g(u(t)) B(\phi^*(t), \lambda^*(t))\ , \tag{14}$$

where $B(\phi, \lambda) = \lambda\alpha_{\min} - \lambda(\kappa_S^{\max} - \kappa_S^{\min} + \kappa_R^{\max} - \kappa_R^{\min})\phi^2 + \lambda(\kappa_R^{\max} - \kappa_R^{\min} + \kappa_S^{\max} - \kappa_S^{\min} + \delta_{\min} - \alpha_{\min})\phi$. As long as $g(u)$ is a monotonically increasing function of u, then the resulting control is said to be "bang-bang", where $u(t)$ only takes extreme values. The switching times between maximum and minimum values of u is determined by $B(\phi(t), \lambda(t))$, the switching function, yielding the optimal control solution:

$$\begin{aligned} &u^*(t) = 0 \text{ if } B(\phi^*(t), \lambda^*(t)) > 0, \\ &u^*(t) = 1 \text{ if } B(\phi^*(t), \lambda^*(t)) < 0, \text{ and} \\ &u^*(t) \in [0, 1] \text{ otherwise.} \end{aligned}$$

In principle, the optimal control trajectory can be obtained through numerical integration of the model dynamics (Eq. (7)) along with the corresponding costate equation (Eq. (12)). In the context of bang-bang control on non-Markovian systems however, using this approach can be difficult, as it requires careful choice of integration technique and update rule. Thus, we turn to reinforcement learning.

As shown above, the optimal control solution to Problem (6) follows bang-bang control, regardless of model parameters. This allows the continuous model of Eq. (1) to be simplified to a discrete model with binary controls without altering the optimal solution. This yields:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), u(t)) = \begin{cases} \mathbf{T}\mathbf{x}(t) & \text{for} \quad u = 1 \text{ (Treatment Phase)} \\ \mathbf{P}\mathbf{x}(t) & \text{for} \quad u = 0 \text{ (Pause Phase)} \end{cases}, \tag{15}$$

where now there are two state-transition matrices, given by:

$$\mathbf{T} = \begin{bmatrix} \kappa_S^{\min} - \alpha_{\max} & 0 \\ \alpha_{\max} & \kappa_R^{\min} \end{bmatrix}, \qquad \mathbf{P} = \begin{bmatrix} \kappa_S^{\max} & \delta_{\max} \\ 0 & \kappa_R^{\min} - \delta_{\max} \end{bmatrix}. \tag{16}$$

We use this formulation of the environment when training the reinforcement learning agent.

A.1 Obtaining the optimal solution in the memoryless case

Previous work [9] has shown that the optimal pulsing protocol for the memoryless case requires an initial drug application phase until the resistant fraction reaches some upper threshold ϕ_h, followed by a pause phase in which the resistant fraction decreases to some lower bound ϕ_l. This is followed by repeated cycles of drug treatment and pause phases where the switching time occurs when the resistant fraction reaches ϕ_h and ϕ_l, respectively. To obtain the optimal values of ϕ_h and ϕ_l for our specific model parameterization, we swept over values of ϕ_h and ϕ_l between 0.1 and 0.9, with increments of 0.04, selecting the values which yielded the highest net death rate ($\phi_l = 0.48$ and $\phi_h = 0.52$).

B Reinforcement Learning Details

We adapted DQN from Stable-Baselines3 [26] with a Double DQN action selection rule [33]. We train the RL agent for 3×10^5 total environment steps. We limited the episodes to be of length 10^4 steps (corresponding to 100 hours in simulation). An MLP of fixed size (2 hidden layers with 64 dimensions each and ReLU activation) was used to parameterize the Q-function.

Regarding the environment and training, we list several key implementation choices: At the beginning of an episode, the state is zero-padded to always be of length $K = 5$. We experimented with adding a penalty for allowing the number of cells to increase beyond the initial amount upon termination, but found this was not necessary for successful training. Borrowing terminology from the literature on Atari environments [22], we stack K frames (previous cost values) to form the RL agent's state vector. Although initially we let K be a μ-dependent hyperparameter, we found a constant choice of $K = 5$ to work well across the values of μ studied.

We find exponentially decaying the exploration parameter ε to work better than the typical l inear annealing (with constant, positive final ε) scheduling. We conjecture that this is because non-greedy actions can be quite detrimental (causing the environment to terminate), and exponentially decaying ε ensures some exploration continues to occur but with increasingly fewer random actions. To ensure the agent is not overly myopic (especially for such long episodes) we found a large discount factor of $\gamma = 0.999$ (corresponding to an effective horizon of $H = (1-\gamma)^{-1} = 10^3$) to be helpful. When training the agent, we wait until the completion of one rollout episode, and take as many gradient steps as environment steps have occurred.

B.1 Hyperparameters

We find that sweeping over a range of hyperparameters (as shown in Table 2) did not have a significant effect on performance, though for reproducibility we list the final hyperparameters used (for $\mu = 0.7$) below in Table 1.

Table 1: Finetuned Hyperparameter Values for Double DQN

Hyperparameter	Finetuned Value
batch size	32
buffer size	100,000
exploration rate	0.05
frames stacked	5
gradient steps (UTD / RR)	1
learning rate	3.60×10^{-4}
target update interval	$1,000$
discount factor	0.999
learning starts	$10,000$

Table 2: Hyperparameter Sweep Ranges

Hyperparameter	Sweep Values
batch size	16, 32, 64
exploration rate	$0.01 - 0.2$
learning rate	$10^{-5} - 10^{-3}$
target update interval	1,000, 5,000, 10,000, 30,000
target Polyak averaging	0.95, 0.99, 0.995, 1.0

C Further Experimental Results

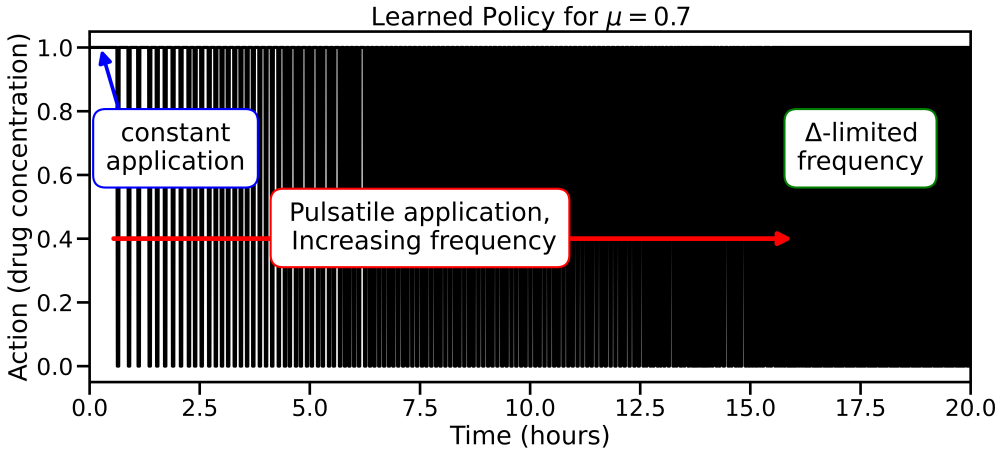

Figure 3: The learned policy shows a resemblance to the optimal memoryless strategy, with an initial constant application phase followed by a pulsatile phase. However, in the case of memory-based dynamics, the frequency of pulsing must be increased over time as discussed in Sec. 3. Since the policy is eventually limited by the simulation time, the pulsing frequency becomes bottlenecked by our choice of time discretization Δ after ≈ 20 hours. Despite this, the policy is still able to perform well with rapid pulsing.

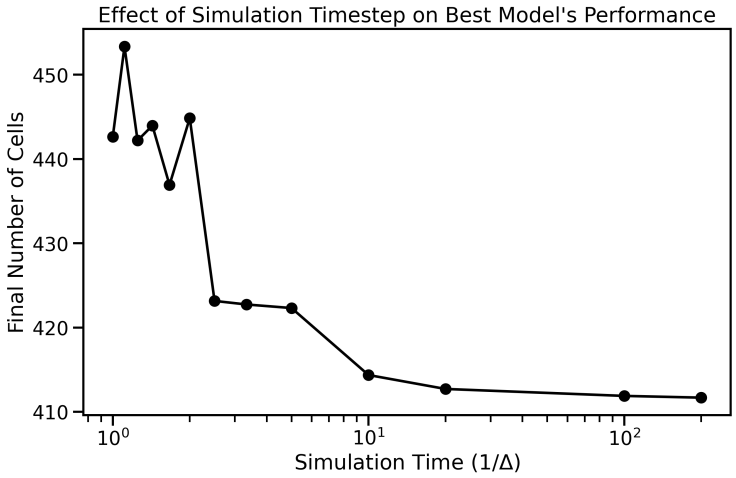

Figure 4: We find that the simulation time can have a significant effect on RL performance. For each choice of Δ, we run a random sweep of size 30 over various hyperparameters, selecting the highest-performing run for each Δ. Since smaller values of Δ require longer compute-times for simulations, there is a tradeoff between the amount of time (also, the inverse of max pulsing rate, which may be more relevant in clinical settings) and the best performance. We have chosen to use $\Delta = 0.01$ throughout.