Bayes-Adaptive Impulse Control of Piecewise-Deterministic Markov Processes

Orlane Rossini, Meritxell Vinyals, Alice Cleynen, Benoîte de Saporta, Régis Sabbadin

Bayes-Adaptive Impulse Control of Piecewise-Deterministic Markov Processes

Orlane Rossini[1] **Meritxell Vinyals**[2] **Alice Cleynen**[3] **Benoîte de Saporta**[1] **Régis Sabbadin**[2]
[1]IMAG, Univ Montpellier, CNRS, Montpellier, France
[2]Univ Toulouse, INRAE-MIAT, Toulouse, France
[3]John Curtin School of Medical Research, Australian National University, Canberra, ACT, Australia

Abstract

We introduce a Bayes-adaptive framework for controlling Piecewise-Deterministic Markov Processes (PDMPs) under partial observability and uncertain parameters. PDMPs form a class of continuous-time Markov processes that capture hybrid (discrete–continuous) dynamics, allowing highly flexible modeling with a small number of interpretable parameters. We cast the problem as a hybrid state-space Bayes-Adaptive Partially Observable Markov Decision Process (BAPOMDP), which accounts for model uncertainty offline, without requiring prior interaction with the system. This BAPOMDP can be seen as a higher-dimensional hybrid Partially Observable Markov Decision Process (POMDP). Because computing optimal policies in such hybrid state-space POMDPs is intractable, we rely on simulation-based deep reinforcement learning algorithms to obtain effective solutions. The approach is demonstrated in a medical patient follow-up scenario, where numerical experiments highlight the feasibility of applying this framework in realistic settings with partial observability and uncertain dynamics.

1 Introduction

Piecewise-Deterministic Markov Processes (PDMPs), first introduced by [Davis, 1984], are a family of Markov processes characterized by deterministic motion interspersed with random jumps. This structure allows PDMPs to model not only the discrete, random events of a system but also the continuous, deterministic dynamics between these events. Due to this dual capability, PDMPs have been widely applied to describe phenomena in various fields, including biology, economics, and medicine [Anderson and Kurtz, 2011, Goan et al., 2023, Wang and Chen, 2023].

The problem of optimally controlling a fully observed PDMP via punctual decisions, also known as *impulse control of PDMPs*, was first formulated by [Costa and Davis, 1989] and further extended by subsequent works [Costa, 1991, Dempster and Ye, 1995, Dufour et al., 2016, de Saporta et al., 2017]. In cases of partial observations, this problem can be framed as a continuous-space Partially Observed Markov Decision Process (POMDP). While approximation methods like discretization [Cleynen and de Saporta, 2018, Cleynen and de Saporta, 2023] and Monte Carlo planning [de Saporta et al., 2024] have been proposed to solve these POMDPs, they all share a critical assumption: the underlying PDMP model is perfectly known.

In this article, we address the more challenging problem of controlling partially observed PDMPs with *ill-known parameters*. A key challenge is that the values of these parameters are generally uncertain at planning time. Although prior knowledge can come from experts or available data, and online learning methods may sometimes be applied, many critical applications, such as personalized medical treatment, do not permit extensive real-world experimentation. Therefore, there is a clear need for a planning approach that explicitly handles model uncertainty, a challenge we address using the *Bayes-Adaptive Partially Observed Markov Decision Processes (BAPOMDP)* framework [Duff, 2002, Pineau et al., 2008]. This framework is designed to manage model uncertainty a priori, without assuming prior interaction with the controlled system.

The classical BAPOMDP approach augments the orig-

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

inal POMDP's state space to include hyperparameters representing parameters uncertainty. For finite-horizon problems with finite states and observations, the resulting BAPOMDP also has a finite, though exponentially larger, state space [Pineau et al., 2008]. In our case, we demonstrate that modeling controlled PDMPs within a Bayes-adaptive framework results in an even more complex augmented POMDP with a continuous, multidimensional state space. Our contributions are therefore three-fold.

1. We propose a Bayes-adaptive framework to model and solve partially observed controlled PDMPs with ill-known parameter values. We motivate our approach with a medical patient follow-up application. Unlike standard BAPOMDPs, our resulting framework handles continuous state spaces and uses state transitions to generate bayesian estimates of the dynamics parameters.
2. We note that solving the resulting BAPOMDP is significantly more difficult than solving a POMDP derived from a controlled PDMP with known parameters. We therefore benchmark several state-of-the-art deep reinforcement learning (deep RL) algorithms to derive near-optimal policies for our BAPOMDP model of the medical patient follow-up problem.
3. Finally, we evaluate the robustness of the policies obtained from our Bayes-adaptive approach by comparing them against a deep RL baseline that handles uncertainty only as an initial distribution. Our experiments demonstrate that the Bayes-adaptive approach is computationally effective for controlling PDMPs with ill-known parameters.

The remainder of this paper is organized as follows. Section 2 provides an overview of impulse control of PDMPs. Section 3 introduces our main contribution: the Bayes-adaptive approach to model impulse-controlled PDMPs with uncertain parameters, and its motivating medical example. Finally, Section 4 presents, first, a benchmark of classical deep RL algorithms on the constructed BAPOMDP, and second, an empirical evaluation of the benefits of using an adaptive model compared to a non-adaptive approach.

2 Control of Piecewise Deterministic Markov Processes

2.1 Piecewise Deterministic Markov Process

We start with a formal definition of a Piecewise Deterministic Markov Process (PDMP), its state space and its local characteristics.

Definition 2.1 (PDMP). *A Piecewise Deterministic Markov Process $X = (X_t)_{t\geq 0}$ is defined by a tuple $\mathbf{P} = \langle E, \Phi, \lambda, Q\rangle$, where*

- *The* state space *E has the hybrid form*

$$E = \bigcup_{\mathsf{m}\in\mathsf{M}} \{\mathsf{m}\} \times \mathsf{E}_{\mathsf{m}},$$

for some finite mode set M, and where for all $\mathsf{m} \in \mathsf{M}$, E_{m} is some Borel subset of $\mathbb{R}^{d_{\mathsf{m}}}$. Let $\mathcal{B}(E)$ be its Borel σ-field. Denote $\bar{E}$ the closure of E.

A state will be denoted $x = (\mathsf{m}, \mathsf{x}) \in E$.

- *The* flow *Φ is a continuous function from $E \times \mathbb{R}_+$ onto E satisfying a semi-group property, i.e. $\Phi(\cdot, t+s) = \Phi(\Phi(\cdot, t), s)$ for all $t, s \in \mathbb{R}_+$ and leaving the mode unchanged. The flow prescribes the deterministic motion between jumps. We write $\Phi(x, t) = (\mathsf{m}, \Phi_{\mathsf{m}}(\mathsf{x}, t))$, for all $x = (\mathsf{m}, \mathsf{x}) \in E$.*

- *The* jump intensity*, also known as hazard rate or risk function λ is a measurable function from $\bar{E}$ onto $\mathbb{R}_+$ that determines the occurrence of random jumps and is such that for any x in E, there exists $\epsilon > 0$ such that*

$$\int_0^\epsilon \lambda(\Phi(x, t))dt < +\infty,$$

forbidding instantaneous jumps. We also write $\lambda(x) = \lambda_{\mathsf{m}}(\mathsf{x})$, for all $x = (\mathsf{m}, \mathsf{x}) \in E$.

- *The* jump kernel *Q is a Markov kernel on $(\mathcal{B}(E), \bar{E})$ that selects the new location of the process after each jump. The probability $Q(B|\mathsf{m}, \mathsf{x}) = Q_{\mathsf{m}}(B|\mathsf{x})$ is the probability to jump to $B \subset E$ when a jump occurs at $\mathsf{x} \in E_{\mathsf{m}}$. It satisfies $Q(\{x\}|x) = 0, \forall x \in E$ so that a jump has to change the state of the process.*

The PDMP dynamics can be described informally as follows: starting from some initial point $x \in E$, the motion of the process follows the deterministic flow $t \mapsto \Phi(x, t)$ until a first jump time T_1. Jumps may occur via two means: *random jumps* occur from the realization of the random clock with intensity λ, while *boundary jumps* occur when the process reaches the boundary ∂E of the state space. Thus, starting from x at time 0, the first jump time T_1 has the following distribution

$$\begin{aligned}\mathbb{P}_x(T_1 > t) &= \mathbb{P}(T_1 > t \mid X_0 = x) \\ &= \exp\left(-\int_0^t \lambda(\Phi(x, u))\, du\right) \mathbb{1}_{t<t^*(x)},\end{aligned}$$

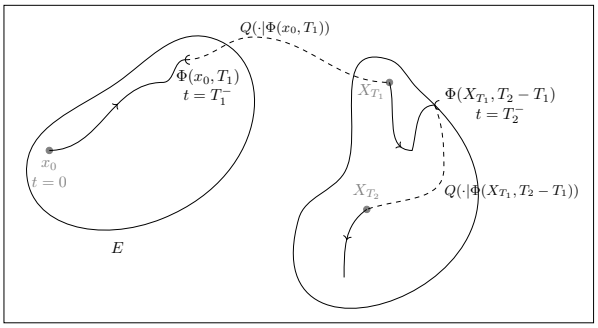


Figure 1: Trajectory of a generic PDMP. Starting from an initial value x_0 at time $t = 0$, the process follows a deterministic trajectory until a jump occurs, either at a random time (as at T_1) or because the process reaches the state boundary (as at T_2). At jump times, the process jumps to a new location drawn from kernel Q.

where $t^*(x)$ is the deterministic time the flow takes to reach the boundary ∂E of E when it starts from x:

$$t^*(x) = t^*_{\mathsf{m}}(\mathsf{x}) = \inf\{t > 0 : \Phi(x, t) \in \partial E\}.$$

At T_1 the process jumps to a new point $x' = X_{T_1}$ selected according to the distribution $Q(\cdot|\Phi(x, T_1))$ and the motion restarts from this new point as before. A generic representation of a PDMP is given in Figure 1. The flow Φ, the jump rate λ, and the Markov kernel Q are called the *local characteristics* of the PDMP $\langle E, \Phi, \lambda, Q\rangle$.

Note that the dynamics of a PDMP can be simulated easily when Φ and λ are explicitly given and when Q can be simulated.

2.2 Control of a PDMP

For a PDMP on a state space E, let $\{\langle\Phi, \lambda, Q\rangle^\ell : \ell \in L\}$ be the finite set of (user-chosen) available dynamics. Controlling the PDMP means that the decision-maker can switch dynamics after some delay $r \in \mathbb{T} = [0, T]$, with $0 < T < +\infty$.

We define action space $\mathbb{A} = L \times \mathbb{T}$, where an action $a = (\ell, r) \in \mathbb{A}$ consists of the chosen dynamic ℓ and a finite delay r before the next switching. The set of admissible actions in state $x \in E$ is denoted by $A(x) \subseteq \mathbb{A}$. A *policy* to control a PDMP is a function $\pi : E \to \mathbb{A}$ such that $\pi(x) = (\ell, r) \in A(x), \forall x \in E$. Given an arbitrary policy π, the PDMP $\mathbf{P} = \{\langle\Phi, \lambda, Q\rangle^\pi\}$ on state space E is augmented with deterministic jumps at decision times, where the active dynamic ℓ is updated and the current state remains unchanged.

The *expected cost* of a policy π applied at time $t_0 = 0$ in state x_0 is defined as follows. First, a *running cost* $c_R : E \to \mathbb{R}$ captures the cost accumulated along the trajectory $(X^\pi_t)_{t\in[0,T]}$. Then, an *impulse cost* is incurred whenever a dynamic change occurs: $c_I : E \times E \to \mathbb{R}$ captures the cost incurred when applying $\pi(x_{t_n}) = (\ell_n, r_n) \in A(x_{t_n})$ in state x_{t_n}.

The *expected cost* of an impulse policy π applied from x_0 at time $t = 0$ to time T, is then defined by:

$$V(\pi, x_0) = \mathbb{E}^\pi_{x_0}\left[\int_0^T c_R(X^\pi_t)dt + \sum_{n=0}^{\infty} c_I(X^\pi_{t_n}, X^\pi_{t_n^+})\right].$$

Solving a controlled PDMP amounts to compute a policy with the minimal *expected cost* up to a given precision. Such policies always exists [Costa and Davis, 1989].

2.3 Partially Observed Controlled PDMP

The state x of a controlled PDMP may not be fully observed. In particular, the mode m may be hidden, or the Euclidean part x observed with noise. Partial observability does not impact the PDMP transition model. However, modelling partially-observed controlled PDMP requires the addition of an *observation space*

$$\Omega = \bigcup_{\mathsf{m}\in\mathsf{M}} \Omega_{\mathsf{m}}, \text{ where } \Omega_{\mathsf{m}} \in \mathbb{R}^{d'_{\mathsf{m}}}.$$

Then, we define an *observation function* $O : E \times \mathbb{A} \times \Omega \to [0, 1]$. $O(x, a, \omega) \in [0, 1]$ is the *probability* of observing ω when action $a = (\ell, r)$ was previously applied and led to current state $x \in E$.

A *partially observed* controlled PDMP can be modelled as a *Partially Observed Markov Decision Process* (POMDP) [Cleynen et al., 2025]. Then, an optimal control policy shall take into account past decisions and observations to make new decisions. Thus, we define $\mathcal{H}$, the set of all possible *histories*. At any given decision time $t_n \in [0, T)$: $h_{t_n} = \langle a_{t_0}, \omega_{t_1}, \ldots, a_{t_n-1}, \omega_{t_n}\rangle$. And $\mathcal{H} = \{h_{t_n}, t_n \in [0, T]\}$.

A controlled POMDP *history-dependent* policy is a function $\pi : \mathcal{H} \to \mathbb{A}$. Such history dependent policy assigns a probability distribution over actions in $\mathbb{A}$ to any partial history $h_{t_n} \in \mathcal{H}$, at decision time t_n.

Note that the resulting POMDP has a hybrid state space and a transition kernel built from the PDMP's local characteristics. Unlike the simple structure of a standard POMDP, this transition kernel depends on actions (it changes depending on the chosen PDMP dynamics) and is considerably more complex to manage. Moreover, the underlying process evolves in continuous time rather than in discrete steps.

2.4 Motivating medical example

Consider a medical setting in which patients cured from cancer are monitored over a follow-up period of length T (e.g., 2400 days), starting from the onset of their first remission at time $t = 0$. During follow-up, cancer relapses may occur, which can be either curable or incurable, and patients may subsequently transition to new remission phases or to death. Hospital visits must be scheduled throughout the follow-up period, at which blood samples are checked for cancer marker presence. After each visit, based on marker levels, treatment decisions may be made, with prescribed therapies maintained until the subsequent visit.

We give here a short description of this case study, as a motivating example. The full details of the model are in the Appendix B.

Let us first describe the *controlled PDMP* model $\langle E, (\Phi, \lambda, Q)^\ell \rangle$ for the medical scenario.

The **state space** $E = \{E_{\mathsf{m}}\}_{\mathsf{m} \in \mathsf{M}}$ is defined as follows.

$\mathsf{M} = \{0, 1, 2, 3\} \times \{0, 1, \ldots, K\}$ represents the mode part. A mode $m = (h, k) \in M$ is composed of a *health status*, h ($h = 0$: remission, $h = 1$: curable relapse, $h = 2$: non curable relapse, $h = 3$: death) and of a number, $k \in \{0, ..., K\}$, of past applied treatments. We assume that an increase in the number of past treatments decreases treatment efficiency and increases the probability of a non-curable relapse.

The quantitative part of the state is described by a Euclidian variable, $\mathsf{x} = (\zeta, u, \tau, t) \in \bar{E} = [\zeta_0, D] \times [0, T]^3$ where ζ is some marker level that keeps track of the disease progression, u is the time that has been spent in the current health status, τ is the time since the current treatment has been applied, and t is the time since the beginning of the follow-up. A treatment has to be applied for at least 45 days.

Since ζ_0 is the marker level of a patient in remission and D is the marker level of a dead patient, we have $E_{\mathsf{m}} = (\zeta_0, D) \times [0, T]^3$ whenever $\mathsf{m} = (h, k)$ is such that $h \in \{1, 2\}$ (curable or non-curable relapse), $E_{\mathsf{m}} = \{\zeta_0\} \times [0, T]^3$ whenever $h = 0$ (remission) and $E_{\mathsf{m}} = \{\partial\}$ whenever $h = 3$ (death). State ∂ is an absorbing state.

The **decision space** is defined as follows. A full action is $a = (\ell, r)$, where $r \in \mathbb{T} = \{15, 30, 60\}$ is the delay until next visit, during which dynamic ℓ is applied. We consider two sets of dynamics $\{\langle \Phi, \Lambda, Q \rangle^\ell : \ell \in \{0, 1\}\}$ ($\ell = 0$: no treatment, $\ell = 1$: under treatment). Constraints $A(x)$ ensure treatment lasts at least 45 days: if at a visit: $\mathsf{x} = (\zeta, u, \tau)$ with $0 < \tau < 45$ the dynamic cannot be changed.

Flow, **Risk intensity** and **Jump kernel** are detailed in Appendix B.1. However, note that the flow corresponds to biomarker growth without treatment, and to biomarker decline under treatment during relapse: for $\mathsf{x} = (\zeta, u, \tau, t)$,

$$\Phi^1_{(1,k)}(\mathsf{x}, s) = (\zeta e^{-\frac{v_1}{k} s},\, u + s,\, \tau + s,\, t + s),$$
$$\Phi^0_{(1,k)}(\mathsf{x}, s) = (\zeta e^{v_1 s},\, u + s,\, 0,\, t + s),$$
$$\Phi^\ell_{(2,k)}(\mathsf{x}, s) = (\zeta e^{v_2 s},\, u + s,\, (\tau + s)\mathbb{1}_{\ell=1},\, t + s),$$
$$\Phi^\ell_{(0,k)}(\mathsf{x}, s) = (\zeta_0,\, u + s,\, (\tau + s)\mathbb{1}_{\ell=1},\, t + s).$$

In the next Section, we will consider that parameter v_1 is unknown.

In words, random jumps are possible from health state $h = 0$ (remission) to health state $h = 1$ (curable relapse) and to health state $h = 2$ (incurable relapse). Both jump intensities actually only depend on the number of relapses, k and the time spent in health status $h = 0$ (u). It is also possible to switch directly from a curable relapse to a non-curable one. It depends on the marker level ζ instead of the time spent in curable relapse, and on the curent dynamic ℓ. All other jumps are impossible, except jumps from $h \in \{1, 2\}$ to $h' = 3$ (death) which occur deterministically when the marker level reaches D.

Partial Observation. In the above controlled PDMP, marker measurements are intrinsically subject to variations independent of the medical condition. These fluctuations can be attributed to measurement errors, natural variations and external influences. The biomarker is thus observed through an additive noise. Let $y = \zeta + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 1)$ be the noisy biomarker. In addition, the patient's overall health h is hidden, together with the time u since the last change of health status, except when the patient is deceased, with $z = 1_{(h=3)}$ the death indicator. The time since the current treatment started, τ, and the number of treatments, k, are perfectly observed.

The POMDP corresponding to the controlled partially observed PDMP is presented in Appendix B.2. The transition kernel $P(x' \mid x, a)$ is derived from a combination of the PDMP's local characteristics. Although its expression is lengthy, its structure allows for simple simulation.

3 Bayes-Adaptive Control of PDMP

Controlled PDMPs are very general and flexible models for planning under uncertainty which generalize Markov and Semi-Markov Decision Processes [Hu and Yue, 2008]. In many applications of controlled PDMP, it is useful to consider a parameterized

version of Φ, λ and Q since, while the general structure of the flow, jump intensity and probability kernel are often known, the precise values of the parameters of the dynamics may be ill-known.

3.1 Context and state-of-the art of control under ill-known parameters

The controlled PDMP framework relies on a flow Φ, a jump intensity λ and a probability kernel Q which may be ill-known in practice.

Let us assume that Φ, λ, and Q are parameterized and that their parameters are unknown. The parameterized dynamics model is written $\langle \Phi_v, \lambda_w, Q_z \rangle$ with parameters v, w, and z following probability density functions $(f^{\Phi}_{v|\theta_\Phi}, f^{\lambda}_{w|\theta_\lambda}, f^{Q}_{z|\theta_Q})$, with parameters $\theta_\Phi, \theta_\lambda$ and θ_Q.

Updated knowledge about the parameters of the probability density functions may be acquired whenever observations of the controlled PDMP are acquired online. When only few on-line observations are available (because the PDMP is controlled through a limited number of interactions with the environment), the uncertainty about these parameters must be explicitly considered off-line in the planning framework, before any interaction with the real environment occurs.

Uncertainty on the **transition and observation functions** has been addressed in the sub-case of discrete-time POMDP (which is encompassed in PDMPs by assuming constant flow and specific jump intensities leading to discrete-time Markovian dynamics), with the introduction of BAPOMDP [Ross et al., 2011]. In BAPOMDP, the agent must not only make decisions based on partial observations of the environment, but also adapt to the current knowledge about the unknown parameters that affect the system dynamics. These parameters are treated as random variables with associated prior distributions, which are updated over time using Bayesian inference as new observations are acquired [Duff, 2002]. This allows the agent to maintain and refine their belief about the environment while interacting with the system, improving decision-making in the face of both partial observation and uncertainty about the system's underlying dynamics.

Uncertainty on the **jump intensity** has also been tackled in the simpler case of Semi-Markov Decision Processes by [Kohar, 2020]. In this case, time can be continuous, but the state space is discrete and the authors have proposed a Bayes-adaptive approach to handle uncertain continuous sojourn time duration (linked to jump intensity).

In the following sections, we focus on a scenario where the **parametric flow** is partially unknown, while other dynamics, as well as the observation function, are fully known. Handling full model uncertainty in flow, jump intensity and transition kernel would lead to a more complex BAPOMDP model, leveraging the contributions of [Ross et al., 2011], [Kohar, 2020] and of this paper. This is left for future research.

3.2 Modelling controlled PDMP with ill-known parameters as a BAPOMDP

Let $\{\langle E, (\Phi, \lambda, Q)^\ell \rangle : \ell \in L\}$ be a controlled PDMP under partial observations and parametric flow, where parameter vector v is unknown but supposed to have a probability density $f^{\Phi}_{v|\theta_\Phi}$ parameterized by a vector of hyper-parameters θ_Φ (which we will denote θ for short, in the following, since only Φ will be parameterized).

As established in [Cleynen et al., 2025] and described in Section 2.3, a partially observed controlled PDMP can be modelled as a *partially observed MDP* $\langle E, \mathbb{A} = L \times \mathbb{T}, P, R, \Omega, O \rangle$.

The transition kernel $P(\cdot|x, a)$ specifies the transitions of the POMDP. Choosing $a = (\ell, r)$ in state $x \in E$ specifies that decision dynamic ℓ will be applied from now on, until date $t' = t + r$.

The cost function c should, in full generality, be defined by

$$c(x, a) = \mathbb{E}\left[\int_0^r c_R(X_t^{\ell'})dt \middle| \; X_0 = x\right] + \tilde{c}_I(x, a),$$

where $\tilde{c}_I(x, a) = c_I(X^{\ell}_{t_n}, X^{\ell'}_{t_n})$ if $a = (\ell', r)$ and at time t_n, the decision is taken to change the dynamic ℓ to ℓ'. In practice, such a cost function is generally intractable to compute, and alternative cost functions close to c, depending only on the values of $\langle x, a, x' \rangle$ are considered instead.

Note that this POMDP has a finite action set, $\mathbb{A}$, and a finite number of decision steps, since $\mathbb{T}$ is finite, excludes 0, and the horizon T is finite. In contrast, the state space E is hybrid, combining discrete and continuous components: jumps may occur at any time between two consecutive decision steps, while the flow evolves continuously. The finiteness of $\mathbb{T}$ does not alter this property.

As a consequence, classical BAPOMDP methods (e.g., [Ross et al., 2011]) cannot be directly applied to this setting. In the following, we introduce a novel update function designed to address this Bayes-adaptive framework.

First, the state space is augmented to handle flow hyperparameters θ:

$$\tilde{E} = E \times \Theta.$$

Then, the following transition kernel $\tilde{P}(\tilde{x}', \theta' | \tilde{x} = (x, \theta), a = (\ell, r))$ specifies the transitions of the BAPOMDP.

$$\tilde{P}\big(\tilde{x}' = (x', \theta') \in B_E \times B_\Theta \mid \tilde{x} = (x, \theta), a\big)$$

$$= \int_{B_E} 1_{B_\Theta} \mathcal{U}(\theta, x, a, x') \times P(dx' \mid x, a, \theta).$$

for any Borel subsets B_E, B_Θ of E and Θ, where $\mathcal{U}(\theta, x, a, x')$ is the posterior update function.

Typically authors assume that the prior over the transition function follows a Dirichlet distribution [Duff, 2002, Ross et al., 2011]. In contrast, our setting places the prior on a specific parameter of the transition function, and this parameter has continuous support. As a result, the posterior update does not directly depend on the transition $\langle x, a, x' \rangle$, as in the classical case, but instead requires inferring the unknown flow parameter v from the observed transition:

$$\mathcal{U}(\theta, x, a, x') = \mathcal{W}(\theta, \mathcal{V}(x, a, x')),$$

where $\mathcal{V}(x, a, x')$ denotes the inference function for the unknown parameter θ, and $\mathcal{W}(\theta, \hat{v})$ represents the prior update function.

Example: Assume that the flow parameter follows a log-normal distribution. Consequently, one might model the prior distribution over the log-normal parameters using a Normal-Gamma prior [Fink, 1997], though other prior distributions could also be considered. Theorem 3.1 provides the update formulas for the Normal-Gamma distribution, which will be used in the next section.

Theorem 3.1. *Let $v = (v^1, \ldots, v^n)$ be an i.i.d. sample drawn from a log-normal distribution:*

$$v^i \mid \mu, \sigma^2 \sim \textit{Log-}\mathcal{N}(\mu, 1/\sigma^2),$$

where the mean μ and the variance σ^2 are unknown.

The conjugate prior for $(\mu, 1/\sigma^2)$ is assumed to be a normal-gamma distribution:

$$(\mu, 1/\sigma^2) \sim \mathcal{NT}(\mu_0, \kappa_0, \alpha_0, \beta_0),$$

where $(\mu_0, \kappa_0, \alpha_0, \beta_0)$ are the prior parameters.

Assume that, at time n, the posterior distribution of $(\mu, 1/\sigma^2)$ follows a normal-gamma distribution:

$$(\mu, 1/\sigma^2) \sim \mathcal{NT}(\mu_n, \kappa_n, \alpha_n, \beta_n).$$

Then, after a new observation v^{n+1} is obtained via $\mathcal{V}(x_n, a_n, x_{n+1})$, the posterior parameters are given by:

$$\mathcal{W}(\theta_{n+1}, v_{n+1}) = \begin{cases} \mu_{n+1} & = \frac{\kappa_n \mu_n + \log(v^{n+1})}{\kappa_n + 1} \\ \kappa_{n+1} & = \kappa_n + 1 \\ \alpha_{n+1} & = \alpha_n + 1/2 \\ \beta_{n+1} & = \beta_n + \frac{\kappa_n (\log(v^{n+1}) - \mu_n)^2}{2(\kappa_n + 1)} \end{cases}.$$

Proof. The proof is derived from [Murphy, 2007, Fink, 1997] and available in Appendix A. □

3.3 Medical example formalized as a BAPOMDP

In the following, it is assumed that all PDMP parameters are known, except in remission where we assume that $v_1 \sim \text{Log-}\mathcal{N}(\mu, 1/\sigma^2)$, where the mean μ and variance σ^2 are unknown.

We now formalize the medical example of Section 2.4 as a BAPOMDP $\langle \tilde{E}, \mathbb{A}, \Omega, A, \tilde{P}, O, c \rangle$. Its characteristics are defined as follows:

- The state space $\tilde{E} = E \times \Theta$ with E from the PDMP model in Section 2.4 and $\Theta \subset \mathbb{R}^4$ the space of hyperparameters. As explained in Section 2.4, model uncertainty only concerns the value of the flow parameter v_1, which follows a log-normal distribution with unknown parameters μ and σ. According to Theorem 3.1, a conjugate prior for the log-normal distribution is a normal-gamma distribution. Therefore, the set of hyperparameters Θ consists of the set of vectors $(\mu_n, \kappa_n, \alpha_n, \beta_n) \subseteq \mathbb{R}^4$.
- The action space $\mathbb{A}$, the observation space Ω, the available actions sets A, the observation kernel O and the cost function c are the same as those defined in Section 2.4 and in Appendix B.2.
- The transition kernel $\tilde{P}(\cdot | \tilde{x} = (x, \theta), a)$ is defined in Section 3.2:
  - at the start of a patient's trajectory, the unknown parameters μ and σ are assumed to follow a Normal-Gamma distribution with hyperparameters $(\mu_0, \kappa_0, \alpha_0, \beta_0)$,
  - the posterior update function $\mathcal{U}$ is defined in the example of Section 3.2, and
  - the transition kernel P remains the same as in Section 2.4, but now depends on the parameter vector $\theta = (\mu_n, \kappa_n, \alpha_n, \beta_n)$.

Unlike standard BAPOMDP cases where the transition update only depends on $\langle x, a, x' \rangle$, here the update of the function $\mathcal{U}$ is different because we estimate the distribution of the flow parameter v_1 with function

$\mathcal{V}(x, a, x')$ detailed in Table 1. This estimate allows us to update the posterior distribution of v_1 using the Bayesian update formulas for a normal-gamma prior, available in Theorem 3.1.

Table 1: Function $\mathcal{V}(x, a, x')$ to estimate the slope $\hat{v_1}$ from transition $\langle x, a, x' \rangle$.

ℓ	(h, h')	Expression
0	(0,1)	$\frac{1}{u'} \log\left(\frac{\zeta'}{\zeta_0}\right)$
0	(1,1)	$\frac{1}{r} \log\left(\frac{\zeta'}{\zeta}\right)$
0	(1,2)	$\frac{1}{r-u'} \left(\log\left(\frac{\zeta'}{\zeta}\right) - v_2 u'\right)$
1	(1,0)	$\frac{k}{r-u'} \log\left(\frac{\zeta}{\zeta'}\right)$
1	(1,1)	$\frac{k}{r} \log\left(\frac{\zeta}{\zeta'}\right)$
1	(1,2)	$\frac{k}{r-u'} \left(\log\left(\frac{\zeta}{\zeta'}\right) - v_2 u'\right)$

4 Benchmarking solution algorithms for BAPOMDP frameworks

This section details a practical implementation of our BAPOMDP framework using simulation-based deep reinforcement learning (RL) and evaluates its effectiveness in the medical patient follow-up case study. We first conduct an extensive benchmark to assess the ability of deep RL algorithms to solve the BAPOMDP. We then evaluate the resulting policies on the real model (i.e., instantiated with real parameters), and compare their performance against a non-adaptive baseline. Experiments' code is provided as supplementary material.

4.1 Solving a BAPOMDP

The BAPOMDP, formulated in Section 3.3, is an augmented finite-horizon POMDP with a continuous, multi-dimensional state space, rendering classical tabular RL algorithms impractical. We therefore turn to deep RL methods, which are well suited to such multidimensional continuous settings.

In simulation-based RL, the algorithm iteratively improves its policy by interacting with a simulator of the model—in this case, the BAPOMDP. At each step, the algorithm selects an action, which the simulator executes, returning the resulting observation transition ($\omega \to \omega'$) and reward, defined as the opposite of $r(x, a) = -c(x, a)$. The algorithm then updates its policy based on these experiences, gradually converging toward a near-optimal deep policy.

Given the wide variety of deep RL algorithms and configuration options, identifying the most effective for computing near-optimal policies for the BAPOMDP is a challenging task. To address this, we conduct an extensive benchmark of the main deep RL algorithms, options, and hyperparameters on our BAPOMDP. A detailed account of the study, including implementation, tuning procedures, and extended results, is provided in the Appendix C. Based on this analysis, we adopt for the remainder of our evaluation the deep DQN algorithm [Mnih et al., 2013], combined with action-masking to handle invalid actions and tuned hyperparameters as specified in the Appendix C.3.

4.2 Illustrative empirical Evaluation

We assess the performance of the policies computed from the BAPOMDP model at test time, i.e., when applied to the real model with the true distribution of the unknown parameter v_1.

To examine how the level of uncertainty in the prior affects performance, we consider three scenarios of increasing certainty about parameter θ that differ in the amount of information provided by the prior distribution. Table 2 reports the prior values associated with each scenario where v_1 follows a log-normal distribution with parameters $\mu = -6.40$ and $\sigma = 0.0$.

Table 2: Prior values for the three experimental scenarios where v_1 follows a log-normal distribution with parameters $\mu = -6.40$ and $\sigma = 0.0$.

Prior	$(\mu_0, \kappa_0, \alpha_0, \beta_0)$
θ_{weak}	(1, 0.001, 1.01, 1)
θ_{medium}	(−6.785, 5.001, 3.51, 1)
θ_{high}	(−6.23, 10, 6.01, 1)

As a baseline, we compare our approach with a non-adaptive POMDP. In this model, the unknown parameter v_1 is modeled as a fixed Normal-Gamma random variable. Thus, when solving the model via simulation-based RL, parameter v_1 is drawn from the Normal-Gamma prior at the start of each simulated patient trajectory.

For each scenario, we compute policies for both the BAPOMDP and the non-adaptive POMDP using simulation-based RL with the DQN algorithm, configured with the best-performing options and hyperparameters identified in the benchmarking analysis (Section 4.1). To account for training variability, this process is repeated five times, yielding five policies per model and scenario.

We then evaluate these policies on the real model by running 5,000 Monte Carlo simulations of complete trajectories for each policy. The performance of each model in each scenario is measured as the average tra-

jectory reward, aggregated over the $5 \times 5{,}000$ trajectories. Results, reported in Table 3, show comparable performance between the BAPOMDP and the non-adaptive POMDP, with outcomes depending on the informativeness of the prior. Specifically, under a weakly informative prior, the non-adaptive POMDP slightly outperforms the BAPOMDP (-5.82 ± 0.28 vs. -5.70 ± 0.23). In contrast, with a medium or strong prior, the BAPOMDP outperforms the non-adaptive approach (-5.74 ± 0.29 vs. -5.86 ± 0.13 and -5.77 ± 0.12 vs. -5.84 ± 0.11).

Table 3: Performance evaluation of BAPOMDP and non-adaptive POMDP policies. Reported values correspond to the average trajectory reward (with standard deviation) computed from 25,000 Monte Carlo simulations on the real model for each scenario.

Prior	BAPOMDP	Non-adaptive POMDP
θ_{weak}	-5.82 ± 0.28	-5.70 ± 0.23
θ_{medium}	-5.74 ± 0.29	-5.86 ± 0.13
θ_{high}	-5.77 ± 0.12	-5.84 ± 0.11

Overall, these results indicate that although the performance is close to that of standard POMDPs approaches, the BAPOMDP handles prior uncertainty more effectively, up to the point where the prior becomes entirely uninformative, in which case it provides no advantage over non-adaptive methods.

5 Concluding remarks

In this article, we have provided a framework for the optimal control of Piecewise-Deterministic Markov Processes (PDMPs) under partial state and model observability. We have considered the case where the parameters of the PDMP flow are ill-known and a probability distribution over their values is maintained. We have proposed a Bayes-adaptive method to address the learning-while-managing PDMP control problem. A key practical finding of this paper is to have demonstrated that the problem of optimally controlling PDMPs under parameter uncertainty and partial observability can be modeled as a hybrid-state Partially Observable Markov Decision Process (POMDP).

The BAPOMDP framework is not an online reinforcement learning framework. Instead, updates to the POMDP model based on future observations are predetermined before taking any action and are already incorporated into the BAPOMDP model. Thus, while the original POMDP model is ill-known, the constructed BAPOMDP model is fully-known at planning time.

Solving hybrid state POMDPs is highly intractable. Therefore, we propose and evaluate an implementation of our approach that approximates the solution of the BAPOMDP using simulation-based deep reinforcement learning. Numerical experiments in the medical patient follow-up application show that our approach achieves test-time performance comparable to currently used non-adaptive models, while offering the advantage of being a principled method grounded in the standard Bayesian inference framework.

As additional real-world data becomes available, the agent's knowledge can be refined by constructing a new BAPOMDP with a more accurate prior. The agent can then be retrained on this updated model, benefiting from the improved information. Over time, as this process is repeated with progressively richer data and increasingly adaptive priors, the framework could naturally evolve toward an online reinforcement learning paradigm, where model updates and policy improvements occur continuously during interaction.

The field of deep reinforcement learning (RL) research for hybrid-state POMDPs is experiencing rapid growth. Emerging algorithms, such as those based on transformer models [Vaswani et al., 2023], are being investigated for POMDP solutions. We believe that the unique characteristics of Bayes-adaptive control of PDMPs make this problem a valuable new benchmark for exploring the capabilities of cutting-edge deep RL methods. Our work paves the way for controlling more complex PDMPs, where future research could explore scenarios in which the jump intensity or Markov kernel also contain multiple unknown parametric components, going beyond the uncertainty of a single flow parameter considered in this study.

References

[Anderson and Kurtz, 2011] Anderson, D. F. and Kurtz, T. G. (2011). *Design and Analysis of Biomolecular Circuits: Engineering Approaches to Systems and Synthetic Biology*, chapter Continuous Time Markov Chain Models for Chemical Reaction Networks, pages 3–42. Springer New York, New York, NY.

[Cleynen and de Saporta, 2018] Cleynen, A. and de Saporta, B. (2018). Change-point detection for piecewise deterministic markov processes. *Automatica*, 97:234–247.

[Cleynen and de Saporta, 2023] Cleynen, A. and de Saporta, B. (2023). Numerical method to solve impulse control problems for partially observed piecewise deterministic markov processes.

[Cleynen et al., 2025] Cleynen, A., de Saporta, B., Rossini, O., Sabbadin, R., and Vernay, A. (2025).

Bridging impulse control of piecewise deterministic markov processes and markov decision processes: Frameworks, extensions, and open challenges.

[Costa, 1991] Costa, O. (1991). Impulse control of piecewise-deterministic processes via linear programming. *IEEE Transactions on Automatic Control*, 36(3):371–375.

[Costa and Davis, 1989] Costa, O. L. V. and Davis, M. H. A. (1989). Impulse control of piecewise-deterministic processes. *Mathematics of Control, Signals, and Systems*, 2(3):187–206.

[Davis, 1984] Davis, M. H. A. (1984). Piecewise-Deterministic Markov Processes: A General Class of Non-Diffusion Stochastic Models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 46(3):353–376.

[de Saporta et al., 2017] de Saporta, B., Dufour, F., and et al. (2017). Optimal strategies for impulse control of piecewise deterministic Markov processes. *Automatica*, 77:219–229.

[de Saporta et al., 2024] de Saporta, B., Thierry d'Argenlieu, A., Sabbadin, R., and Cleynen, A. (2024). A monte-carlo planning strategy for medical follow-up optimization: Illustration on multiple myeloma data. *PLOS ONE*, 19(12):1–23.

[Dempster and Ye, 1995] Dempster, M. A. H. and Ye, J. J. (1995). Impulse control of piecewise deterministic Markov processes. *The Annals of Applied Probability*, 5(2):399–423.

[Duff, 2002] Duff, M. (2002). *Optimal learning : Computational procedures for Bayes-adaptive Markov decision processes*. PhD thesis, University of Massachusetts Amherst.

[Dufour et al., 2016] Dufour, F., Horiguchi, M., and Piunovskiy, A. B. (2016). Optimal impulsive control of piecewise deterministic Markov processes. *Stochastics*, 88(7):1073–1098.

[Fink, 1997] Fink, D. (1997). A compendium of conjugate priors. Technical report, Environmental Statistics Group - Department of Biology - Montana State Univeristy.

[Goan et al., 2023] Goan, E., Perrin, D., Mengersen, K., and Fookes, C. (2023). Piecewise deterministic markov processes for bayesian neural networks.

[Hu and Yue, 2008] Hu, Q. and Yue, W. (2008). *Semi-Markov Decision Processes*, pages 105–120. Springer US, Boston, MA.

[Huang and Ontañón, 2020] Huang, S. and Ontañón, S. (2020). A closer look at invalid action masking in policy gradient algorithms. *arXiv preprint arXiv:2006.14171*.

[Kohar, 2020] Kohar, R. (2020). *BAYES-ADAPTIVE SEMI-MARKOV DECISION PROCESSES – A Pathway to Optimal Learning in Sequential Decision Making with Time under Uncertainty*. PhD thesis, Royal Military College of Canada.

[Mnih et al., 2013] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing Atari with Deep Reinforcement Learning. arXiv:1312.5602 [cs].

[Murphy, 2007] Murphy, K. (2007). Conjugate bayesian analysis of the gaussian distribution. Technical report, University of British Columbia.

[Pineau et al., 2008] Pineau, J., Ross, S., and Chaib-draa, B. (2008). Bayes-adaptive pomdps: A new perspective on the explore-exploit tradeoff in partially observable domains. *Journal of Machine Learning Research*, 12.

[Ross et al., 2011] Ross, S., Pineau, J., Chaib-draa, B., and Kreitmann, P. (2011). A Bayesian Approach for Learning and Planning in Partially Observable Markov Decision Processes. *The Journal of Machine Learning Research*, 12(null):1729–1770.

[Schulman et al., 2017] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms.

[Towers et al., 2024] Towers, M., Kwiatkowski, A., Terry, J., Balis, J. U., De Cola, G., Deleu, T., Goulão, M., Kallinteris, A., Krimmel, M., KG, A., et al. (2024). Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*.

[Vaswani et al., 2023] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2023). Attention is all you need.

[Wang and Chen, 2023] Wang, W. and Chen, X. (2023). Piecewise deterministic markov process for condition-based imperfect maintenance models. *Reliability Engineering & System Safety*, 236:109271.

[Wu et al., 2021] Wu, Z., Liang, E., Luo, M., Mika, S., Gonzalez, J. E., and Stoica, I. (2021). RLlib flow: Distributed reinforcement learning is a dataflow problem. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Bayes-Adaptive Impulse Control of Piecewise-Deterministic Markov Processes
Supplementary Material

A Proof of Theorem 3.1

Let $v = [v^1, \cdots, v^n]$ be an i.i.d. sample drawn from a log normal distribution Log-$\mathcal{N}(\mu, \lambda)$ where we assume that both the mean μ and the precision $\lambda = \sigma^{-2}$ are unknown. Here we show how to infer those parameters from a Bayesian model with conjugate priors.

A.1 Likelihood of a log normal distribution

The probability density function is defined as

$$\pi(\cdot|\mu, \lambda) = \frac{1}{\cdot}\left(\frac{\lambda}{2\pi}\right)^{\frac{1}{2}} \exp\left(-\frac{\lambda}{2}(\log(\cdot) - \mu)^2\right),$$

hence the likelihood of sample v is

$$\begin{aligned}
\pi(v|\mu, \lambda) &= \prod_{i=1}^{n} \pi(v^i|\mu, \lambda) \\
&\propto \prod_{i=1}^{n} \left(\frac{\lambda}{2\pi}\right)^{\frac{1}{2}} \exp\left(-\frac{\lambda}{2}(\log(v^i) - \mu)^2\right) \\
&= \left(\frac{\lambda}{2\pi}\right)^{\frac{n}{2}} \exp\left(-\frac{\lambda}{2}\sum_{i=1}^{n}(\log(v^i) - \mu)^2\right) \\
&= \left(\frac{\lambda}{2\pi}\right)^{\frac{n}{2}} \exp\left(-\frac{\lambda}{2}(n(\mu - \tilde{v})^2 + \sum_{i=1}^{n}(\log(v^i) - \tilde{v})^2\right),
\end{aligned}$$

where

$$\tilde{v} = \frac{1}{n}\sum_{i=1}^{n} \log(v^i).$$

A.2 The prior

The conjugate prior of the log normal distribution is the normal-Gamma distribution, with density function:

$$\pi(\mu, \lambda|\mu_0, \kappa_0, \alpha_0, \beta_0) = \frac{\beta_0^{\alpha_0}}{\Gamma(\alpha_0)}\left(\frac{2\pi}{\kappa_0}\right)^{-\frac{1}{2}} \lambda^{\alpha_0 + \frac{1}{2} - 1} \exp\left(-\frac{\lambda}{2}(\kappa_0(\mu - \mu_0)^2 + 2\beta_0)\right)$$

Note that it is possible to compute the marginals of such distributions in the following way:

$$\begin{aligned}\pi(\lambda) &= \int_{-\infty}^{\infty} \pi(\mu,\lambda)d\mu \\ &\propto \lambda^{\alpha_0+\frac{1}{2}-1}\exp\left(-\lambda\beta_0\right)\int_{-\infty}^{\infty}\exp\left(-\frac{\lambda\kappa_0}{2}(\mu-\mu_0)^2\right)d\mu \\ &\propto \lambda^{\alpha_0}\exp\left(-\lambda\beta_0\right)\end{aligned}$$

And therefore λ follows a Gamma distribution with parameters α_0+1, β_0. Similarly,

$$\begin{aligned}\pi(\mu) &= \int_{0}^{\infty} \pi(\mu,\lambda)d\lambda \\ &\propto \int_{0}^{\infty}\lambda^{\alpha_0+\frac{1}{2}-1}\exp\left(-\lambda(\beta_0+\frac{\kappa_0(\mu-\mu_0)^2}{2})\right)d\lambda\end{aligned}$$

which integrand corresponds to the density of an unnormalized Gamma distribution $G(a=\alpha_0+\frac{1}{2}, b=\beta_0+\frac{\kappa_0(\mu-\mu_0)^2}{2})$. We can therefore write:

$$\begin{aligned}\pi(\mu) &\propto \frac{\Gamma(a)}{b^a} \\ &\propto b^{-a} \\ &= (\beta_0+\frac{\kappa_0(\mu-\mu_0)^2}{2})^{-\alpha_0-\frac{1}{2}} \\ &= (1+\frac{1}{2\alpha_0}+\frac{\alpha_0\kappa_0(\mu-\mu_0)^2}{\beta_0})^{-\frac{2\alpha_0+1}{2}}\end{aligned}$$

That is a multivariate Student's t distribution with mean μ_0, scale matrix $\beta_0/(\alpha_0\kappa_0)$ and $2\alpha_0$ degrees of freedom.

A.3 The posterior distribution

The posterior is defined as follows.

$$\begin{aligned}\pi(\mu,\lambda|v) &\propto \pi(\mu,\lambda|\mu_0,\kappa_0,\alpha_0,\beta_0)\pi(v|\mu,\lambda) \\ &\propto \lambda^{\alpha_0+\frac{1}{2}-1}\exp\left(-\frac{\lambda}{2}(\kappa_0(\mu-\mu_0)^2+2\beta_0)\right)\times\lambda^{\frac{n}{2}}\exp\left(-\frac{\lambda}{2}\sum_{i=1}^{n}(\log(v^i)-\mu)^2\right) \\ &\propto \lambda^{\frac{1}{2}}\lambda^{\alpha_0+\frac{n}{2}-1}\exp\left(-\beta_0\lambda\right)\exp\left[-\frac{\lambda}{2}\left(\kappa_0(\mu-\mu_0)^2+\sum_{i=1}^{n}(\log(v^i)-\mu)^2\right)\right]\end{aligned}$$

On one side we have

$$\sum_{i=1}^{n}(\log(v^i)-\mu)^2 = n(\mu-\tilde{v})^2+\sum_{i=1}^{n}(\log(v^i)-\tilde{v})^2$$

and on the other side

$$\kappa_0(\mu-\mu_0)^2+n(\mu-\tilde{v})^2 = (\kappa_0+n)(\mu-\mu_n)^2+\frac{\kappa_0 n(\tilde{v}-\mu_0)^2}{\kappa_0+n}$$

where

$$\mu_n = \frac{\kappa_0\mu_0+n\tilde{v}}{\kappa_0+n}$$

.

Therefore,

$$\pi(\mu,\lambda|v) \propto \lambda^{\frac{1}{2}} \exp\left(-\frac{\lambda}{2}(\kappa_0+n)(\mu-\mu_n)^2\right) \\ \times \lambda^{\alpha_0+\frac{n}{2}-1} \exp\left(-\beta_0\lambda\right) \exp\left(-\frac{\lambda}{2}\sum_{i=1}^{n}(\log(v^i)-\tilde{v})^2)\right) \exp\left(-\frac{\lambda}{2}\frac{\kappa_0 n(\tilde{v}-\mu_0)^2}{\kappa_0+n}\right)$$

which is once again a normal-Gamma distribution with parameters

$$\begin{aligned}\mu_n &= \frac{\kappa_0\mu_0+n\tilde{v}}{\kappa_0+n}\\ \kappa_n &= \kappa_0+n\\ \alpha_n &= \alpha_0+n/2\\ \beta_n &= \beta_0+\frac{1}{2}\sum_{i=1}^{n}(\log(v^i)-\tilde{v})^2+\frac{\kappa_0 n(\tilde{v}-\mu_0)^2}{2(\kappa_0+n)}\end{aligned}$$

A.4 The posterior marginals distribution

From Section A.2 we have

$$\begin{aligned}\pi(\lambda|v) &= G(\lambda|\alpha_n,\beta_n)\\ \pi(\mu|v) &= T_{2\alpha_n}(\mu|\mu_n,\beta_n/(\alpha_n\kappa_n))\end{aligned}$$

Note that in the special case where $n=1$ (only one new observation), the updated parameters are simply

$$\begin{aligned}\mu_{n+1} &= \frac{\kappa_n\mu_n+\log(v_n+1)}{\kappa_n+1}\\ \kappa_{n+1} &= \kappa_n+1\\ \alpha_{n+1} &= \alpha_n+1/2\\ \beta_{n+1} &= \beta_n+\frac{\kappa_n(\log(v_n+1)-\mu_n)^2}{2(\kappa_n+1)}.\end{aligned}$$

B Medical Example Details

B.1 Medical example as a PDMP

Subsets E_{m} are defined as follows:

- $E_{\mathsf{m}=(0,k)} = \{\zeta_0\}\times[0,T]^3,\ \forall k$: a patient in remission (health status $h=0$) has a marker level of ζ_0, whatever his current treatment and number of past treatments.
- $E_{\mathsf{m}=(1,k)} = [\zeta_0,D)\times[0,T]^3,\ \forall k$: a patient suffering a relapse (neither dead nor in remission) has a marker level $\zeta_0\leq\zeta<D$, whatever the number of past treatments.
- $E_{\mathsf{m}=(2,k)} = [\zeta_0,D)\times[0,T]^3,\ \forall k$: a patient suffering an incurable relapse (neither dead nor in remission) has a marker level $\zeta_0\leq\zeta<D$, whatever the number of past treatments.

- $E_{\mathsf{m}=(3)} = \{\partial\}$: a patient dies when her marker level reaches D (and reaches an arbitrary absorbing state, ∂), whatever his number of past treatments.

Let $\{\langle \Phi, \lambda, Q\rangle^\ell : \ell \in \{0,1\}$ denote the finite collection of available dynamics for the PDMP on the state space E. The case $\ell = 0$ corresponds to the dynamics without treatment, while $\ell = 1$ corresponds to the dynamics under treatment.

Dynamics without treatment ($\ell = 0$)**.** The biomarker evolution follows an exponential pattern. Under remission ($h = 0$), it remains constant at ζ_0, while for a relapse ($h = 1$ and $h = 2$), it is increasing:

$$\begin{aligned}
\Phi^0_{(0,k)}(\mathsf{x} = (\zeta, u, \tau, t), s) &= (\zeta_0,\, u+s,\, 0, t+s),\\
\Phi^0_{(1,k)}(\mathsf{x} = (\zeta, u, \tau, t), s) &= (\zeta e^{v_1 s},\, u+s,\, 0, t+s),\\
\Phi^0_{(2,k)}(\mathsf{x} = (\zeta, u, \tau, t), s) &= (\zeta e^{v_2 s},\, u+s,\, 0, t+s).
\end{aligned}$$

The time required for the process to reach its boundary is infinite under remission, and, under relapses, correspond to the time required to reach the death biomarker level D:

$$\begin{aligned}
t^{0,\star}_{(0,k)}(\mathsf{x}) &= +\infty,\\
t^{0,\star}_{(1,k)}(\mathsf{x}) &= \frac{1}{v_1}\log(\frac{D}{\zeta}),\\
t^{0,\star}_{(2,k)}(\mathsf{x}) &= \frac{1}{v_2}\log(\frac{D}{\zeta}).
\end{aligned}$$

From remission, the probability of relapse occurrence increases with the duration of time spent in remission and the number of previous treatments. From standard relapse, the risk of relapses associated with therapeutic escape is influenced by the biomarker level and the number of previous treatments. In light of these considerations, we choose Weibull distributions of the form: $\mu^0_{\mathsf{m}\to(h',k)}(u) = (\alpha^0_{\mathsf{m}\to(h',k)}u)^{\beta^0_{\mathsf{m}\to(h',k)}}$ and $\mu^0_{\mathsf{m}\to(2,k)}(\zeta) = (\alpha^0_k\zeta)^{\beta^0}$:

$$\begin{aligned}
\lambda^0_{(0,k)}(\mathsf{x}) &= (\mu^0_{\mathsf{m}\to(1,k)} + \mu^0_{\mathsf{m}\to(2,k)})(u),\\
\lambda^0_{(1,k)}(\mathsf{x}) &= \mu^0_{\mathsf{m}\to(2,k)}(\zeta),\\
\lambda^0_{(2,k)}(\mathsf{x}) &= 0.
\end{aligned}$$

In remission, the patient may transition to either a standard relapse or an incurable relapse. In the case of relapse or therapeutic escape, the biomarker increases to a critical value D, leading to the patient's death. A therapeutic escape may occur at any time. We define the Markov kernel Q (case $h = 3$ is omitted as no jumps are allowed when patients are dead) for all real-valued bounded measurable functions g on E :

$$\begin{aligned}
Q^0_{(0,k)}g(x) &= g(1,k,\zeta_0,0,0,t)\frac{\mu^0_{\mathsf{m}\to(1,k)}(u)}{(\mu^0_{\mathsf{m}\to(1,k)} + \mu^0_{\mathsf{m}\to(2,k)})(u)}\\
&\quad + g(2,k,\zeta_0,0,0,t)\frac{\mu^0_{\mathsf{m}\to(2,k)}(u)}{(\mu^0_{\mathsf{m}\to(1,k)} + \mu^0_{\mathsf{m}\to(2,k)})(u)},\\
Q^0_{(1,k)}g(x) &= g(2,k,\zeta,0,0,t)\mathbb{1}_{D>\zeta} + h(3,k,\partial)\mathbb{1}_{\zeta=D},\\
Q^0_{(2,k)}g(x) &= g(3,k,\partial)\mathbb{1}_{\zeta=D}.
\end{aligned}$$

Dynamics with treatment ($\ell = 1$)**.** The biomarker evolution follows an exponential pattern. Under remission ($h = 0$), it remains constant at ζ_0, while for a relapse $h = 1$ it is decreasing, and for incurable relapse ($h = 2$), it is increasing:

$$\begin{aligned}
\Phi^1_{(0,k)}(\mathsf{x} = (\zeta, u, \tau, t), s) &= (\zeta_0,\, u+s,\, \tau+s, t+s),\\
\Phi^1_{(1,k)}(\mathsf{x} = (\zeta, u, \tau, t), s) &= (\zeta e^{-\frac{v_1}{k}s},\, u+s,\, \tau+s, t+s),\\
\Phi^1_{(2,k)}(\mathsf{x} = (\zeta, u, \tau, t), s) &= (\zeta e^{v_2 s},\, u+s,\, \tau+s, t+s).
\end{aligned}$$

The time required for the process to reach its boundary is infinite under remission, and, under relapse, corresponds to the time required to reach the nominal value ζ_0 while under incurable relapse it is the time required to reach the death biomarker level D:

$$\begin{aligned} t^{1,\star}_{(0,k)}(\mathsf{x}) &= +\infty, \\ t^{1,\star}_{(1,k)}(\mathsf{x}) &= \frac{k}{v_1}\log(\frac{\zeta}{\zeta_0}), \\ t^{1,\star}_{(2,k)}(\mathsf{x}) &= \frac{1}{v_2}\log(\frac{D}{\zeta}). \end{aligned}$$

From remission, the probability of incurable relapse occurrence increases with the duration of time spent in remission and the number of previous treatments. From standard relapse, the risk of relapses associated with therapeutic escape is influenced by the biomarker level and the number of previous treatments. In light of these considerations, we, once again, choose Weibull distributions of the form: $\mu^1_{(0,k)\to(h',k)}(u) = (\alpha^1_{(0,k)\to(h',k)}u)^{\beta^1_{(0,k)\to(h',k)}}$ and $\mu^1_{(1,k)\to(2,k)}(\zeta) = (\alpha^1_k\zeta)^{\beta^1}$:

$$\begin{aligned} \lambda^1_{(0,k)}(\mathsf{x}) &= \mu^1_{(0,k)\to(2,k)}(u), \\ \lambda^1_{(1,k)}(\mathsf{x}) &= \mu^1_{(1,k)\to(2,k)}(\zeta), \\ \lambda^1_{(2,k)}(\mathsf{x}) &= 0. \end{aligned}$$

In remission, the patient may transition to an incurable relapse. In the case of relapse the biomarker decreases to ζ_0 and returns to remission. Therapeutic escape may occur at any time. In the case of therapeutic escape, the biomarker increases toward the D threshold, ultimately resulting in the patient's death. We define the Markov kernel Q (case $h = 3$ is omitted as no jumps are allowed when patients are dead) for all real-valued bounded measurable functions g on E :

$$\begin{aligned} Q^1_{(0,k)}g(x) &= g(2,k,\zeta_0,0,\tau,t), \\ Q^1_{(1,k)}g(x) &= g(2,k,\zeta,0,\tau,t)\mathbb{1}_{\zeta>\zeta_0} + g(0,k,\zeta_0,0,\tau,t)\mathbb{1}_{\zeta=\zeta_0}, \\ Q^1_{(2,k)}g(x) &= g(3,k,\partial)\mathbb{1}_{\zeta=D}. \end{aligned}$$

Under both dynamics, one could write the transition kernel $\mathcal{P}\ell$ as a combination of the deterministic flow, the jump intensity and the Markov kernel such that for a time period r and all real-valued bounded measurable functions g on E, $\mathcal{P}^\ell g(x) = \mathbb{E}[g(X_r)|X_0 = x, \ell]$.

The transition kernel $\mathcal{P}$ can be written as a combination of the deterministic flow, the jump intensity and the Markov kernel such that for a time period r and all real-valued bounded measurable functions g on E, $\mathcal{P}g(x) = \mathbb{E}[g(X_r)|X_0 = x]$. It is assumed that in the event of a relapse from the remission state ($m = 0$), the patient takes more than 60 days to die ($t^{0,\star}_{(0,k)}(\mathsf{x}) > 60$ and $t^{\ell,\star}_{(2,k)}(\mathsf{x}) > 60$), this implies $v_1 < 0.061$ and $v_2 < 0.061$. As this form of $\mathcal{P}^\ell$ is complex, we work case by case for $x = (\mathsf{m} = (h,k), \mathsf{x} = (\zeta, u, \tau, t)) \in E$.

- If $\mathsf{m} = (3)$, then $\mathcal{P}^\ell g(x) = g(3,k,\partial)$, the patient is dead.
- If $\mathsf{m} = (2,k)$, over a time step r the patient may either die or remain in therapeutic escape.
$$\mathcal{P}^\ell g(x) = g(3,k,\partial)\mathbb{1}_{t^{\ell,\star}_{\mathsf{m}}(\zeta)\le r} + g(\mathsf{m}, \Phi^\ell_{\mathsf{m}}(\mathsf{x}, r))\mathbb{1}_{t^{\ell,\star}_{\mathsf{m}}(\zeta)>r}.$$
- If $\mathsf{m} = (1,k)$, and no treatment is applied ($\ell = 0$), the biomarker increases. Over a time step r, the patient may die, remain in relapse, transition to a therapeutic escape and stay there, or transition to a therapeutic escape and die.

$$\begin{aligned} \mathcal{P}^0 g(x) = {} & g(3,k,\partial)e^{-\Lambda^\ell_{\mathsf{m}}\left(\mathsf{x}, t^{\ell,\star}_{\mathsf{m}}(\zeta)\right)}\mathbb{1}_{t^{\ell,\star}_{\mathsf{m}}(\zeta)\le r} + g(\mathsf{m}, \Phi^\ell_{\mathsf{m}}(\mathsf{x}, r))e^{-\Lambda^\ell_{\mathsf{m}}(\mathsf{x},r)}\mathbb{1}_{t^{\ell,\star}_{\mathsf{m}}(\zeta)>r} \\ & + \int_0^r g\left(2,k,\Phi^\ell_{(2,0,k)}\left(\Phi^\ell_{\mathsf{m}}(\zeta,s), r-s\right), r-s, 0, t+r\right) \\ & \quad \times \mu^\ell_{\mathsf{m}\to(2,k)}(\Phi^\ell_{\mathsf{m}}(\zeta,s))e^{-\Lambda^\ell_{\mathsf{m}}(\mathsf{x},s)}\mathbb{1}_{t^{\ell,\star}_{(2,k)}(\Phi^\ell_{\mathsf{m}}(\zeta,s))>r-s}\mathbb{1}_{t^{\ell,\star}_{\mathsf{m}}(\zeta)>s}\,ds \\ & + \int_0^r g(3,k,\partial)\mu^\ell_{\mathsf{m}\to(2,k)}(\Phi^\ell_{\mathsf{m}}(\zeta,s))e^{-\Lambda^\ell_{\mathsf{m}}(\mathsf{x},s)}\mathbb{1}_{t^{\ell,\star}_{(2,k)}(\Phi^\ell_{\mathsf{m}}(\zeta,s))\le r-s}\,ds \end{aligned}$$

- If $\mathsf{m} = (1, k)$, and treatment is applied ($\ell = 1$), the biomarker decreases. Over a time step r, the patient may fall in remission and remain there, remain in relapse, transition to a therapeutic escape and remain there, fall into remission and transition to a therapeutic escape, transition to a therapeutic escape and die.

$$
\begin{aligned}
\mathcal{P}^1 g(x) = {} & g\left(0, k, \zeta_0, r - t^{\ell,\star}_{\mathsf{m}}(\zeta), \tau + r, t + r\right) e^{-\Lambda^{\ell}_{\mathsf{m}}(\mathsf{x}, t^{\ell,\star}_{\mathsf{m}}(\zeta))} \mathbb{1}_{t^{\ell,\star}_{\mathsf{m}}(\zeta) \leq r} \\
& + g(1, k, \Phi^{\ell}_{\mathsf{m}}(\zeta, r), u + r, \tau + r, t + r) e^{-\Lambda^{\ell}_{\mathsf{m}}(\mathsf{x}, r)} \mathbb{1}_{t^{\ell,\star}_{\mathsf{m}}(\zeta) > r} \\
& + \int_0^r g\left(2, k, \Phi^{\ell}_{(2,k)}\left(\Phi^{\ell}_{\mathsf{m}}(\zeta, s), r - s\right), r - s, \tau + r, t + r\right) \\
& \quad \times \mu^{\ell}_{\mathsf{m} \to (2,k)}(\Phi^{\ell}_{\mathsf{m}}(\zeta, s)) e^{-\Lambda^{\ell}_{\mathsf{m}}(\mathsf{x}, s)} \mathbb{1}_{t^{\ell,\star}_{\mathsf{m}}(\zeta) > s} \mathbb{1}_{t^{\ell,\star}_{(2,k)}(\Phi^{\ell}_{\mathsf{m}}(\zeta, s)) > r - s} \, ds \\
& + \int_0^{r - t^{\ell,\star}_{\mathsf{m}}(\zeta)} g(2, k, \Phi^{\ell}_{(2,k)}\left(\zeta_0, r - s - t^{\ell,\star}_{\mathsf{m}}(\zeta)\right), r - s - t^{\ell,\star}_{\mathsf{m}}(\zeta), \tau + r, t + r) \\
& \quad \times \mu^{\ell}_{(0,k) \to (2,k)}(s) e^{-\Lambda^{\ell}_{\mathsf{m}}(\mathsf{x}, t^{\star}_{\mathsf{m}}(\zeta))} e^{-\Lambda^{\ell}_{(0,k)}(\mathsf{x}, s)} \mathbb{1}_{t^{\ell,\star}_{\mathsf{m}}(\zeta) \leq r} ds \\
& + \int_0^r g(3, k, \partial) \mu^{\ell}_{\mathsf{m} \to (2,k)}(\Phi^{\ell}_{\mathsf{m}}(\zeta, s)) e^{-\Lambda^{\ell}_{\mathsf{m}}(\mathsf{x}, s)} \mathbb{1}_{t^{\ell,\star}_{\mathsf{m}}(\zeta) > s} \mathbb{1}_{t^{\ell,\star}_{(2,k)}(\Phi^{\ell}_{\mathsf{m}}(\zeta, s)) \leq r - s} \, ds
\end{aligned}
$$

- If $\mathsf{m} = (0, k)$, and no treatment is applied ($\ell = 0$), the biomarker remains at its nominal value. Over a time step r, the patient may stay in remission, transition to a relapse, transition to a therapeutic escape or transition to a relapse before transitioning to a therapeutic escape.

$$
\begin{aligned}
\mathcal{P}^0 g(x) = {} & g\left(\mathsf{m}, \Phi^{\ell}_{(0,k)}(\mathsf{x}, r)\right) e^{-\Lambda^{\ell}_{\mathsf{m}}(\mathsf{x}, r)} \\
& + \int_0^r g\left(1, k, \Phi^{\ell}_{(1,k)}(\zeta_0, r - s), r - s, 0, t + r\right) \mu^{\ell}_{\mathsf{m} \to (1,k)}(u + s) \\
& \quad \times e^{-\Lambda^{\ell}_{\mathsf{m}}(\mathsf{x}, s)} e^{-\Lambda^{\ell}_{(1,k)}\left(\Phi^{\ell}_{(1,k)}(\zeta_0, r - s), r - s\right)} \, ds \\
& + \int_0^r g\left(2, k, \Phi^{\ell}_{(2,k)}(\zeta_0, r - s), r - s, 0, t + r\right) \mu^{\ell}_{\mathsf{m} \to (2,k)}(u + s) e^{-\Lambda^{\ell}_{\mathsf{m}}(\mathsf{x}, s)} \, ds \\
& + \int_0^r \int_0^{r - s} g\left(2, k, \Phi^{\ell}_{(2,k)}\left(\Phi^{\ell}_{(1,k)}\left(\zeta_0, s'\right), r - s' - s\right), r - s' - s, 0, t + r\right) \\
& \quad \times \mu^{\ell}_{\mathsf{m} \to (1,k)}(u + s) e^{-\Lambda^{\ell}_{\mathsf{m}}(\mathsf{x}, s)} \mu^{\ell}_{(1,k) \to (2,k)}\left(\Phi^{\ell}_{(1,k)}(\zeta_0, s')\right) e^{-\Lambda^{\ell}_{(1,k)}\left((\zeta_0, 0), s'\right)} \, ds' ds
\end{aligned}
$$

- If $\mathsf{m} = (0, k)$, and treatment is applied ($\ell = 1$), the biomarker remains at its nominal value. Over a time step r, the patient may stay in remission or transition to a therapeutic escape.

$$
\begin{aligned}
\mathcal{P}^1 g(x) = {} & g\left(\mathsf{m}, \Phi^{\ell}_{(\mathsf{m})}(\mathsf{x}, r)\right) e^{-\Lambda^{\ell}_{\mathsf{m}}(\mathsf{x}, r)} \\
& + \int_0^r g\left(2, k, \Phi^{\ell}_{(2,k)}\left(\zeta_0, r - s\right), r - s, \tau + r, t + r\right) \mu^{\ell}_{\mathsf{m} \to (2,k)}(u + s) e^{-\Lambda^{\ell}_{\mathsf{m}}(\mathsf{x}, s)} \, ds
\end{aligned}
$$

Although the PDMP transition kernel appears lengthy to write out, its structure allows for straightforward simulation.

B.2 Medical example as a POMDP

The associated POMDP $\langle E, \mathbb{A}, \Omega, A, P, O, c \rangle$, is characterized as follows :

- The state space E corresponds to the state of a patient $x \in E_{\mathsf{m}}$ and the absorbing state $\partial \in E_{(3,k)}$.
- The action space $\mathbb{A}$ collects the decision pairs $a = (\ell, r)$, where $r \in \mathbb{T}$ is the delay until the next visit and $\ell \in L = \{0, 1\}$ is the dynamics choice.

- The observation space Ω corresponds to the observation of a patient $\omega = (k, z, y, \tau, t)$, with $y = \zeta + \epsilon$ where ϵ are real-valued independent and identically distributed random variables with density f independent from the controlled PDMP. Here f correspond to the Gaussian distribution. $z = \mathbb{1}_{h=3}$ is the death indicator.
- The available actions sets are defined by

$$A(\omega) = \begin{cases} \emptyset & \text{if } z = 1, \\ \{1\} \times (\mathbb{T} \cap [0, T-t]) & \text{if } 0 < \tau < 45, \\ \{0,1\} \times (\mathbb{T} \cap [0, T-t]) & \text{otherwise.} \end{cases}$$

- The transition kernel P can be written from the PDMP transition kernel $\mathcal{P}^\ell$, defined above. For all real-valued bounded measurable function g on E, let $Pg(x) = \mathbb{E}[g(X_r)|X_0 = x]$. Then one has

$$Pg(x,a) = \begin{cases} g(\partial) & \text{if } x = \partial \\ \mathcal{P}^\ell g(h, k, \zeta, u, \tau, t+r) & \text{else.} \end{cases}$$

- The observation function does not rely on the action a. For all real-valued bounded measurable function q for $x' = (\mathsf{m}' = (h', k'), \mathsf{x}') \in E_{\mathsf{m}}$

$$Oq(x', \omega) = \begin{cases} q(k, 1, D, t) & \text{if } x' = \partial, \\ \displaystyle\int q(k, 0, y, \tau, t) f(y - \zeta')\, dy & \text{otherwise.} \end{cases}$$

- The cost function c is $c(x, a, x') = r\ell C_\ell + C_V + C_D r \mathbb{1}_{h=3}$ where $a = (\ell, r)$, *i.e.* the cost is composed of a fixed visit cost C_V, a cost of treatment $r\ell C_\ell$ proportional to the treatment duration and a death cost rC_D proportional to the time spent in the death mode.

C Benchmarking simulation-based reinforcement learning on solving BAPOMDPs

This section presents the benchmarking of deep RL algorithm on solving the BAPOMDP in our our medical patient follow-up application. We begin by detailing the experimental setup, including the baseline algorithms and their variants that will be benchmarked on the BAPOMDP model. Next, we describe the tuning of hyperparameters performed for each selected algorithm and finally we analyse the results of the numerical experiments, focusing on the performance of the trained policies.

C.1 Benchmark description

Next we describe the algorithms selected for benchmarking as listed in Table 4. The selection criteria were based on three dimensions: the type of algorithm (value-based or policy-based), the use of memory, and the method for handling invalid actions.

Value-based vs. policy-based. To compare value-based and policy-based RL algorithms, we include one state-of-the-art representative from each family: DQN [Mnih et al., 2013] for value-based methods and PPO [Schulman et al., 2017] for policy-based methods.

memory. Optimal policies in POMDPs generally depend on the history of past observations rather than only on the current observation of the system. Deep RL algorithms typically incorporate recurrent structures to encode this history. To assess the impact of memory, our benchmark includes a variant of PPO equipped with a Long Short-Term Memory (LSTM) module.

Handling invalid actions. In our medical application, two types of invalid actions can occur: (i) actions exceeding the treatment horizon; and (ii) actions that terminate the treatment prematurely. We benchmark the two main approaches [Huang and Ontañón, 2020] proposed in the literature to handle such invalid actions: action penalties and action masking, knowing that action masking has been shown to outperform in general

Table 4: List of benchmarked algorithms.

PPO
PPO with action masking
PPO-LSTM
DQN
DQN with action masking

action penalty. In invalid action penalty, the invalid actions receive large costs so that the agent learns to minimize costs by not executing any invalid actions. In our medical patient-follow up application this translates to constraints that are directly implemented in the cost function with prohibitive cost as follows:

$$c_K(x, a, x') = c(x, a, x') + C_H \mathbb{1}_{t>H} + C_T \mathbb{1}_{\ell=0} \mathbb{1}_{0<\tau<45},$$

where C_H is a cost for exceeding the horizon and C_T penalizes stopping treatment too early. Instead in action masking, invalid actions are masked out and then the algorithm samples just from those actions that are valid. In Table 4, unless otherwise specified, the algorithms handle invalid actions using an action penalty; cases where action masking is applied are explicitly indicated.

Regarding the environment, Table 5 show the parameters of the PDMP (the Patient model) and BAPOMDP models used in the experiments. Both models were implemented using the Gymnasium environment [Towers et al., 2024] to ensure compatibility with the main RL libraries. For each algorithm listed in Table 4 we use in experiments the implementation provided in the RLlib framework [Wu et al., 2021]. Finally, RL algorithms optimize rewards rather than costs. Since our reward is defined as the negation of the cost $r(x, a, x') = -c(x, a, x')$, maximizing reward is equivalent to minimizing cost, ensuring our results remain consistent with the original objective.

C.2 Hyperparameter Tuning

For each algorithm listed in Table 4, we performed a hyperparameter tuning process using the Ray Tune functionality from the RLlib library to identify the best-performing configuration. The set of hyperparameters considered in this tuning is reported in Table 6. For each algorithm, 1 000 configurations were randomly sampled from the ranges specified in Table 6 and trained in the training environment (the RL simulator of the BAPOMDP model). Training for each configuration was terminated once it reached either 100 000 simulation timesteps or 1 000 training iterations, whichever occurred first. The best configuration for each algorithm was defined as the one achieving the highest average reward among all tested configurations. Table 7 summarizes the selected hyperparameters for each algorithm.

C.3 Evaluation

For the evaluation, each algorithm was trained using the best hyperparameter configuration identified in the previous section. To account for randomness, five independent training runs were performed per algorithm. Training was terminated once it reached either 100,000 simulation timesteps or 1,000 training iterations, whichever occurred first. During training, the current policy was evaluated every five iterations in the environment without exploration noise. Each evaluation consisted of 10 episodes, and the performance metric was the average reward per episode under the current policy.

We report the performance of each algorithm, using its best configuration, by plotting the mean episode reward as a function of the number of training interactions. The results are shown in Figures 2 and are analyzed and discussed below.

Impact of action masking. Figures 2a and 2b compares handling invalid actions with action masking versus action penalties in DQN and PPO, respectively. In Figure 2a, we observe that DQN with an action penalty requires significantly more training interactions to learn to avoid invalid actions, whereas action masking enables the agent to incorporate this constraint directly and learn faster. In Figure 2b, the difference for PPO is less pronounced, but the action-masking variant still adapts to invalid actions more quickly than the action-penalty

Table 5: PDMP and BAPOMDP parameter values used in experiments, where $v_1 \sim \text{Log-}\mathcal{N}(\mu, \sigma)$ and (μ, σ) are unknown

Name	Notation	Value
Boundaries		
Biomarker nominal level	ζ_0	1
Biomarker death level	D	40
Horizon	T	2400
Flow		
Slope parameter in relapse	v_1	$\text{Log-}\mathcal{N}(-6.40, 0)$
Slope parameter in non curable relapse	v_2	$U[0.0001, 0.06]$
Jump function		
Scale parameter of $\mu^{\ell}_{(0,k)\to(1,k)}$	$\alpha^{\ell}_{(0,k)\to(1,k)}$	$0.0002/k$
Shape parameter of $\mu^{\ell}_{(0,k)\to(1,k)}$	$\beta^{k}_{(0,k)\to(1,k)}$	3.5
Scale parameter of $\mu^{\ell}_{(0,k)\to(2,k)}$	$\alpha^{k}_{(0,k)\to(2,k)}$	0.00001
Shape parameter of $\mu^{\ell}_{(0,k)\to(2,k)}$	$\beta^{k}_{(0,k)\to(2,k)}$	3.5
Scale parameter of $\mu^{0}_{(1,k)\to(2,k)}$	α^{0}_{k}	$1.116e^{-20}k/4$
Shape parameter of $\mu^{0}_{(1,k)\to(2,k)}$	β^{0}_{k}	4.5
Scale parameter of $\mu^{1}_{(1,k)\to(2,k)}$	α^{1}_{k}	$5.58e^{-7}k$
Shape parameter of $\mu^{1}_{(1,k)\to(2,k)}$	β^{1}_{k}	-0.8
BAPOMDP		
Noise	$\epsilon \sim \mathcal{N}(\mu, \sigma)$	$(\mu = 0, \sigma = 1)$
Initial prior	θ_1	$(-6.785, 5.001, 3.51, 1)$
Cost function		
Treatment cost	C_ℓ	0.1
Visit cost	C_v	1
Death cost	C_D	300
Horizon constraint cost	C_H	1000
Treatment constraint cost	C_T	1000

counterpart.

Impact of memory in PPO. Figure 2c examines the impact of adding memory (LSTM) to PPO. Incorporating LSTM does not significantly alter the overall learning curve compared to standard PPO, as both ultimately achieve similar reward levels. However, PPO-LSTM exhibits substantially lower performance during the initial training iterations before converging to comparable results.

Based on the above results, the remainder of our evaluation focuses on PPO with action masking and DQN with action masking, while excluding the PPO variant with memory and the action-penalty approach.

Impact of algorithm type (value-based vs. policy-based). Figure 2d compares DQN and PPO, both using action masking, to assess the effect of algorithm type. Both algorithms show performance degradation after approximately 15,000 training iterations, indicating overfitting or training collapse; training should therefore be stopped at this point. Up to this stage, their performances are comparable. However, PPO with action masking exhibits high variability, with large reward fluctuations and limited stability, whereas DQN with action masking demonstrates a more stable training process.

Conclusion. From these results, we retain DQN with action masking as the deep RL algorithm of choice for computing near-optimal policies for the BAPOMDP model in our medical application.

References

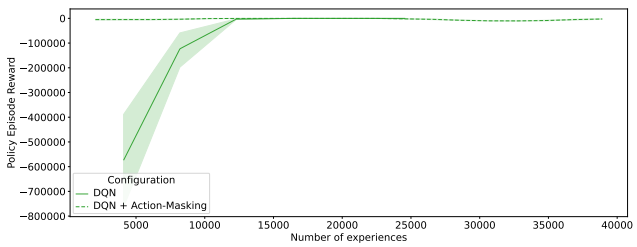
[Anderson and Kurtz, 2011] Anderson, D. F. and Kurtz, T. G. (2011). *Design and Analysis of Biomolecular Circuits: Engineering Approaches to Systems and Synthetic Biology*, chapter Continuous Time Markov Chain

Table 6: Summary of tuned parameters and their possible values.

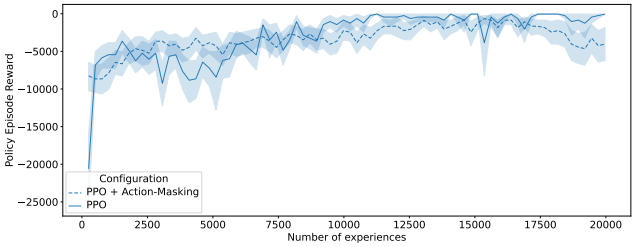
Parameter	**Possible Values**
Shared Hyperparameters	
Discount Factor (γ)	{0.95, 0.97, 0.99, 0.999}
Training Batch Size	{32, 65, 256, 512, 1024, 2048, 4096}
Training Batch Mode	{TruncateEpisodes, CompleteEpisodes}
Learning Rate	{3e-5, 0.0001, 0.0003, 0.001}
Fully Connected Layer Sizes	{[32], [32,32], [64], [64,64]}
Activation Function	{linear, relu, tanh}
Observation Filter	{meanStdFilter, NoFilter}
DQN	
TD Error Loss	{Huber, MSE}
Number of Step	{1, 2, 3}
Double Q-learning	{True, False}
Dueling Architecture	{True, False}
Replay Buffer Capacity	{50000, 100000, 500000, 1000000}
Prioritized Replay Alpha	{0.5, 0.6}
Prioritized Replay Beta	{0.4, 0.5}
Prioritized Replay Epsilon	{1e-6, 3e-6}
Epsilon Timestep	{2, 10000, 50000, 100000, 200000}
Initial Epsilon	{0.9, 1.0, 1.5}
Final Epsilon	{0.0, 0.01, 0.02}
Value Range Min	{-60 000, -50 000, -40 000}
Value Range Max	{-500, -10, 0, 10}
Noisy Nets	{True, False}
Hidden Layers Size	{[64], [128], [256], [512]}
Training Intensity	{1, 4, 16, 32}
Target Network Update Frequency	{500, 1000, 5000, 10000, 20000}
Steps Before Learning Starts	{1000, 10000, 20000}
PPO Without LSTM	
Number of SGD Iterations	{10, 20, 30}
SGD Minibatch Size	{64, 128, 256}
Clipping Parameter	{0.1, 0.2, 0.25, 0.3, 0.4, 0.5}
Value Function Shared Layers	{False, True}
Gradient Clipping	{None, 0.5, 1.0, 2.0, 5.0}
Use Generalized Advantage Estimation (GAE)	{True, False}
Value Function Loss Coefficient	{0.001, 0.01, 0.1, 0.5, 1.0}
PPO With LSTM	
LSTM Cell Size	{64, 256, 512, 1024}
Use Previous Action in LSTM	{False}
Use Previous Reward in LSTM	{False}

Table 7: Best configuration for each algorithm trained with BAPOMDP framework.

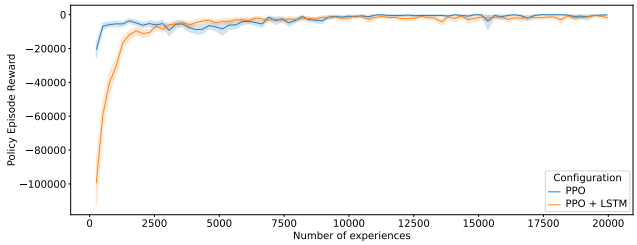
Parameter	**Without Action-Masking**	**with Action-Masking**
DQN		
TD Error Loss	MSE	MSE
Discount Factor (γ)	0.99	0.99
Number of Step	3	3
Double Q-learning	True	False
Dueling Architecture	False	False
Replay Buffer Capacity	500 000	500 000
Prioritized Replay Alpha	0.6	0.6
Prioritized Replay Beta	0.4	0.4
Prioritized Replay Epsilon	3×10^{-6}	3×10^{-6}
Epsilon Timestep	2	50000
Initial Epsilon	1.5	1.5
Final Epsilon	0.0	0.0
Value Range Min	$-50\,000$	$-40\,000$
Value Range Max	10	-500
Training Batch Size	4 096	2 048
Training Batch Mode	CompleteEpisodes	TruncateEpisodes
Learning Rate	0.0003	0.001
Fully Connected Layer Sizes	[64,64]	[64,64]
Activation Function	relu	tanh
Observation Filter	NoFilter	NoFilter
Noisy Nets	False	False
Hidden Layers Size	[512]	[]
Training Intensity	16	16
Target Network Update Frequency	500	500
Steps Before Learning Starts	10 000	20 000
PPO		
Discount Factor (γ)	0.95	0.97
Number of SGD Iterations	20	20
SGD Minibatch Size	256	128
Training Batch Size	256	256
Training Batch Mode	TruncateEpisodes	TruncateEpisodes
Learning Rate	0.001	0.0003
Clipping Parameter	0.25	0.5
Fully Connected Layer Sizes	[64,64]	[64,64]
Activation Function	tanh	linear
Value Function Shared Layers	True	False
Observation Filter	MeanStdFilter	NoFilter
Gradient Clipping	2.0	0.5
Use GAE	True	True
Value Function Loss Coefficient	0.5	0.01
PPO with LSTM		
Discount Factor (γ)	0.97	
Number of SGD Iterations	10	
SGD Minibatch Size	64	
Training Batch Size	256	
Training Batch Mode	TruncateEpisodes	
Learning Rate	3×10^{-5}	
Clipping Parameter	0.3	
Fully Connected Layer Sizes	[64,64]	
Activation Function	linear	
Value Function Shared Layers	True	
Observation Filter	MeanStdFilter	
Gradient Clipping	2.0	
Use GAE	True	
Value Function Loss Coefficient	0.001	
LSTM Cell Size	[64]	
Use Previous Action in LSTM	False	
Use Previous Reward in LSTM	False	



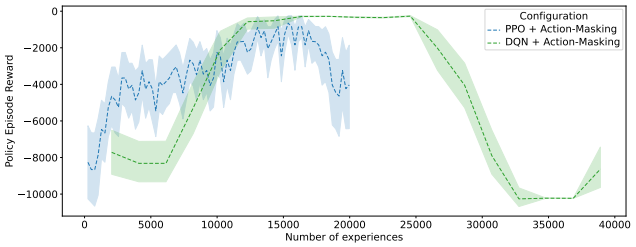
(a) Impact of action-making for DQN algorithm.



(b) Impact of action-masking for PPO algorithm.



(c) Impact of LSTM for PPO algorithm.



(d) Impact of PPO with action-masking or DQN with action-masking algorithm.

Figure 2: Impact of algorithm choice on the performance in terms of mean episode reward per episode with respect to the number of training experiences used on BAPOMDP model.

Models for Chemical Reaction Networks, pages 3–42. Springer New York, New York, NY.

[Cleynen and de Saporta, 2018] Cleynen, A. and de Saporta, B. (2018). Change-point detection for piecewise deterministic markov processes. *Automatica*, 97:234–247.

[Cleynen and de Saporta, 2023] Cleynen, A. and de Saporta, B. (2023). Numerical method to solve impulse control problems for partially observed piecewise deterministic markov processes.

[Cleynen et al., 2025] Cleynen, A., de Saporta, B., Rossini, O., Sabbadin, R., and Vernay, A. (2025). Bridging impulse control of piecewise deterministic markov processes and markov decision processes: Frameworks, extensions, and open challenges.

[Costa, 1991] Costa, O. (1991). Impulse control of piecewise-deterministic processes via linear programming. *IEEE Transactions on Automatic Control*, 36(3):371–375.

[Costa and Davis, 1989] Costa, O. L. V. and Davis, M. H. A. (1989). Impulse control of piecewise-deterministic processes. *Mathematics of Control, Signals, and Systems*, 2(3):187–206.

[Davis, 1984] Davis, M. H. A. (1984). Piecewise-Deterministic Markov Processes: A General Class of Non-Diffusion Stochastic Models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 46(3):353–376.

[de Saporta et al., 2017] de Saporta, B., Dufour, F., and et al. (2017). Optimal strategies for impulse control of piecewise deterministic Markov processes. *Automatica*, 77:219–229.

[de Saporta et al., 2024] de Saporta, B., Thierry d'Argenlieu, A., Sabbadin, R., and Cleynen, A. (2024). A monte-carlo planning strategy for medical follow-up optimization: Illustration on multiple myeloma data. *PLOS ONE*, 19(12):1–23.

[Dempster and Ye, 1995] Dempster, M. A. H. and Ye, J. J. (1995). Impulse control of piecewise deterministic Markov processes. *The Annals of Applied Probability*, 5(2):399–423.

[Duff, 2002] Duff, M. (2002). *Optimal learning : Computational procedures for Bayes-adaptive Markov decision processes*. PhD thesis, University of Massachusetts Amherst.

[Dufour et al., 2016] Dufour, F., Horiguchi, M., and Piunovskiy, A. B. (2016). Optimal impulsive control of piecewise deterministic Markov processes. *Stochastics*, 88(7):1073–1098.

[Fink, 1997] Fink, D. (1997). A compendium of conjugate priors. Technical report, Environmental Statistics Group - Department of Biology - Montana State Univeristy.

[Goan et al., 2023] Goan, E., Perrin, D., Mengersen, K., and Fookes, C. (2023). Piecewise deterministic markov processes for bayesian neural networks.

[Hu and Yue, 2008] Hu, Q. and Yue, W. (2008). *Semi-Markov Decision Processes*, pages 105–120. Springer US, Boston, MA.

[Huang and Ontañón, 2020] Huang, S. and Ontañón, S. (2020). A closer look at invalid action masking in policy gradient algorithms. *arXiv preprint arXiv:2006.14171*.

[Kohar, 2020] Kohar, R. (2020). *BAYES-ADAPTIVE SEMI-MARKOV DECISION PROCESSES – A Pathway to Optimal Learning in Sequential Decision Making with Time under Uncertainty*. PhD thesis, Royal Military College of Canada.

[Mnih et al., 2013] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing Atari with Deep Reinforcement Learning. arXiv:1312.5602 [cs].

[Murphy, 2007] Murphy, K. (2007). Conjugate bayesian analysis of the gaussian distribution. Technical report, University of British Columbia.

[Pineau et al., 2008] Pineau, J., Ross, S., and Chaib-draa, B. (2008). Bayes-adaptive pomdps: A new perspective on the explore-exploit tradeoff in partially observable domains. *Journal of Machine Learning Research*, 12.

[Ross et al., 2011] Ross, S., Pineau, J., Chaib-draa, B., and Kreitmann, P. (2011). A Bayesian Approach for Learning and Planning in Partially Observable Markov Decision Processes. *The Journal of Machine Learning Research*, 12(null):1729–1770.

[Schulman et al., 2017] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms.

[Towers et al., 2024] Towers, M., Kwiatkowski, A., Terry, J., Balis, J. U., De Cola, G., Deleu, T., Goulão, M., Kallinteris, A., Krimmel, M., KG, A., et al. (2024). Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*.

[Vaswani et al., 2023] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2023). Attention is all you need.

[Wang and Chen, 2023] Wang, W. and Chen, X. (2023). Piecewise deterministic markov process for condition-based imperfect maintenance models. *Reliability Engineering & System Safety*, 236:109271.

[Wu et al., 2021] Wu, Z., Liang, E., Luo, M., Mika, S., Gonzalez, J. E., and Stoica, I. (2021). RLlib flow: Distributed reinforcement learning is a dataflow problem. In *Conference on Neural Information Processing Systems (NeurIPS)*.