Received: 30 September 2022 | Revised: 11 April 2023 | Accepted: 17 April 2023

DOI: 10.1111/exsy.13324

ORIGINAL ARTICLE

Expert Systems WILEY

Model-based offline reinforcement learning for sustainable fishery management

Jun Ju[1] | Hanna Kurniawati[2] | Dirk Kroese[1] | Nan Ye[1]

[1]School of Mathematics and Physics, The University of Queensland, St Lucia, Queensland, Australia

[2]School of Computing, Australian National University, Canberra, Australian Capital Territory, Australia

Correspondence
Jun Ju and Nan Ye, School of Mathematics and Physics, The University of Queensland, St Lucia, QLD 4072, Australia.
Email: jun.ju@uq.net.au; nan.ye@uq.edu.au

Funding information
ARC Centre of Excellence for Mathematical and Statistical Frontiers, Grant/Award Number: CE140100049; Australian Research Council, Grant/Award Number: 200101049

Abstract

Fisheries, as indispensable natural resources for human, need to be managed with both short-term economical benefits and long-term sustainability in consideration. This has remained a challenge, because the population and catch dynamics of the fisheries are complex and noisy, while the data available is often scarce and only provides partial information on the dynamics. To address these challenges, we formulate the population and catch dynamics as a Partially Observable Markov Decision Process (POMDP), and propose a model-based offline reinforcement learning approach to learn an optimal management policy. Our approach allows learning fishery management policies from possibly incomplete fishery data generated by a stochastic fishery system. This involves first learning a POMDP fishery model using a novel least squares approach, and then computing the optimal policy for the learned POMDP. The learned fishery dynamics model is useful for explaining the resulting policy's performance. We perform systematic and comprehensive simulation study to quantify the effects of stochasticity in fishery dynamics, proliferation rates, missing values in fishery data, dynamics model misspecification, and variability of effort (e.g., the number of boat days). When the effort is sufficiently variable and the noise is moderate, our method can produce a competitive policy that achieves 85% of the optimal value, even for the hardest case of noisy incomplete data and a misspecified model. Interestingly, the learned policies seem to be robust in the presence of model learning errors. However, non-identifiability kicks in if there is insufficient variability in the effort level and the fishery system is stochastic. This often results in poor policies, highlighting the need for sufficiently informative data. We also provide a theoretical analysis on model misspecification and discuss the tendency of a Schaefer model to overfit compared with a Beverton–Holt model.

KEYWORDS
Beverton–Holt model, fishery management, incomplete data, model misspecification, offline reinforcement learning, POMDP, Schaefer model

1 | INTRODUCTION

Fisheries play a crucial role in human society by serving as a critical source of food, creating numerous job opportunities, contributing to international trade, and providing the basis for recreational fishing. They form an integral part of the intricate ecosystem which we lives in, but the intensive exploitation of the fisheries over the last few decades has caused concerns over the sustainability of the fisheries.

Sustainable fishery management has remained an actively researched area due to various challenges. First, multiple uncertainties need to be carefully considered, including the stochastic nature of stock dynamics, uncertainty about the state of nature, and a lack of knowledge about the fishery system (Charles, 1998; Sethi et al., 2005). Second, even though it helps learning about the fishery system by testing the system's response to different actions, such learning needs to avoid dangerous actions that could lead to the collapse of the fishery system. Third, fishery data may contain many missing values (Damasio et al., 2015; Matsuzaki & Kadoya, 2015; Rudd & Branch, 2017), presenting significant methodological challenges in using such data.

In this paper, we propose a model-based offline reinforcement learning approach (e.g., see Levine et al. [2020]) for sustainable fishery management. Our approach builds on recent advances in the powerful Partially Observable Markov Decision Process (POMDP) framework for decision-making under uncertainty and neural network learning, allowing to automatically learn the fishery dynamics model and corresponding optimal policy directly from historical fishery data. This involves first learning a POMDP fishery model using a novel least squares approach, and then computing the optimal policy for the learned POMDP. Our approach has been instantiated for the simplest setting in a previous preliminary study (Ju et al., 2021), where the environment is deterministic, the data contains no missing data for catches and efforts, and the model is well-specified. This paper extends our prior work so as to handle stochastic environments, incomplete data, and misspecified models.

We highlight two contributions of this paper. First, we propose a novel algorithm for learning fishery dynamics from possibly incomplete data. As compared to Bayesian inference (Ellison, 2004; Punt & Hilborn, 1997) or maximum likelihood estimation (Blamey et al., 2022), our algorithm is conceptually and algorithmically simpler and still capable of learning a good model. It defines a simple sum of squared error objective function for measuring model quality, and learns a model by optimizing the objective function using a stochastic version of the L-BFGS algorithm (Liu & Nocedal, 1989), which is a popular and very efficient quasi-Newton method that can effectively exploit the surface geometry of the objective function using the gradient only. Empirically, our algorithm outperforms the Bayesian method using a noninformative prior. Second, we advocate taking model learning error into account in evaluating algorithms for fishery management and present a comprehensive simulation study on the effect of model learning error in our POMDP-based approach. Several works (Filar et al., 2019; Memarzadeh et al., 2019) have leveraged recent advances in POMDP solver to fully explore the range of possible management strategies, instead of considering only a small set of candidate management strategies considered elicited from the experts, as typically done in practice (Punt et al., 2016). However, these previous works do not consider the effect of model learning error on the quality of the chosen policy: the quality of the chosen policy is evaluated in the estimated or assumed model in these works, while we evaluate the chosen policy with respect to the ground-truth model in our simulation study.

In the remainder of this paper, Section 2 discusses the existing approaches and Section 3 reviews some background materials needed for our approach. Section 4 describes our proposed algorithm. Section 5 introduces simulation study settings and presents the results and analysis, with a theoretical analysis on model misspecification and a discussion on the tendency of a Schaefer model to overfit compared with a Beverton–Holt model. Section 6 concludes the paper.

2 | RELATED WORK

This section briefly reviews research on sustainably managing fisheries for achieving maximum intermediate profits without causing over-fishing in the long term. We focus on discussing how our work relates to or differs from works in the learning of fishery models, management strategy evaluation, and applications of POMDPs in fisheries.

2.1 | Learning of fishery models

Fishery models form an integral part for fisheries management. Beverton and Holt (1957)'s classical work laid the theoretical foundation for population and catch dynamics models. These fishery models and their extensions have become an essential component in optimal management of fisheries. However, focusing on fishery modelling alone can result in accepting models that are inadequate for decision-making.

Fishery models are often learned from fishery data using the Bayesian method (Ellison, 2004; Punt & Hilborn, 1997) and maximum likelihood estimation (MLE) (Blamey et al., 2022). Performing MLE for a fishery model involves computing its likelihood, which is often very complex and difficult to compute because the model involve latent variables (i.e., the unobserved population biomasses). Bayesian method is a popular alternative for estimating the parameters of fishery models. It performs inference by combining the likelihood and a prior distribution on the model parameters using the Bayes rule. This allows quantifying the uncertainties in the parameter estimates in a simple way, but specifying a good prior is often involved.

We propose a novel least squares method for learning fishery models. It is conceptually and algorithmically simpler than both MLE and the Bayesian method. It does not require a prior as in the Bayesian method, and it has a much simpler objective function as compared to MLE. Our model learning algorithm provides more accurate estimates as compared to the Bayesian method in our experiments.

2.2 | Management strategy evaluation

Management strategy evaluation is a holistic management framework that evaluates the management system in its entirety, through using simulation to evaluate the effectiveness of all decisions that leads to management actions. This was initially pursued by two parallel lines of works in 1970s and 1980s: 'adaptive management', and 'comprehensive assessment and management procedure evaluation'. Adaptive management (Hilborn & Sibert, 1988; Walters, 2007; Walters & Hilborn, 1976) emphasizes on the importance of management strategies that can adapt to available experience and data. Comprehensive assessment and management procedure evaluation (Donovan, 1989; Magnusson & Stefánsson, 1989) emphasizes on systematically evaluating different management strategies. The management strategies are typically elicited from the experts. Both initiatives are conceptually the same and the framework has later been called management strategy evaluation (MSE) by some authors (Sainsbury et al., 2000). This approach has found success in various cases (Bunnefeld et al., 2011), and MSE has become a dominant framework for sustainable fishery management, with various extensions and refinements, such as handling of multiple criteria (De Lara & Martinet, 2009) and extreme environmental events (Blamey et al., 2022).

Our work complements MSE with its ability to search for an optimal management strategies among all possible management strategies, rather than just those elicited from the experts. This is done by computing an optimal adaptive policy for a learned fishery model, which is made possible by recent advances in developing efficient POMDP solvers.

2.3 | Applications of POMDPs in fisheries

POMDPs provide a general mathematical framework for decision making under uncertainty, but the use of POMDPs for practical problems was rather limited because of the lack of efficient solution algorithms. The situation has changed in the last two decades with the development of a few efficient POMDP solvers (Kurniawati et al., 2008; Silver & Veness, 2010; Ye et al., 2017).

Several works have explored the use of POMDPs in fisheries. Back in 1989, Lane (1989) developed a small POMDP model for fishermen to make fishing decisions. The use of POMDPs for practical problems was rather limited because of the lack of efficient solution algorithms, but the situation has changed in the last two decades with the development of a few efficient algorithms that can scale up to large POMDPs (Kurniawati et al., 2008; Silver & Veness, 2010; Ye et al., 2017). This allows a few more recent works exploring the POMDP framework for fishery management (Filar et al., 2019; Memarzadeh et al., 2019).

Our work differs from previous works on the POMDP-based approach in two important aspects. First, while previous works on the POMDP-based approach do not consider model learning errors (Filar et al., 2019; Memarzadeh et al., 2019), we present a comprehensive study on the effectiveness of the POMDP-based approach in the presence of model learning errors. Second, we propose a novel least squares algorithm for learning a model directly from catch and effort data. In constrast, Filar et al. (2019) assumes a correct model in their simulation study, and Memarzadeh et al. (2019) considers a simpler model learning problem where Bayesian inference is used to learn a model using catch and biomass data. The data was from the RAM Legacy Stock Assessment Database (Ricard et al., 2012), where the biomass values were results of previous stock assessments.

3 | BACKGROUND

We review some background materials in this section. We briefly describe the fishery population and catch models used in this work in Section 3.1, Partially Observable Markov Decision Processes (POMDPs) in Section 3.2, and reinforcement learning (RL) in Section 3.3. For the convenience of the readers, we provide in Table 1 a summary of the notations used in this paper.

TABLE 1 Table of notations.

Notations for fishery models	
B_t	The population biomass at a discrete time t
ρ	The proliferation rate in the Beverton–Holt model
K	The carrying capacity of the environment
q	The catchability constant in the catch model
r	The intrinsic rate in the Pella-Tomlinson model
m	The shape parameter in the Pella-Tomlinson model
Notations for data	
c_t	The amount of catch at a discrete time t
e_t	The amount of effort at a discrete time t
$T_{missing}$	Set of missing years in a dataset
T	Length of the dataset
$c_{\max}, e_{\max}$	Maximum catch and maximum effort in the data respectively
$\widetilde{c}_t, \widetilde{e}_t$	Normalized values of c_t and e_t
Notations for POMDP	
s,a,z	State, action, and observation respectively
$\mathcal{S},\mathcal{A},\mathcal{Z}$	State space, action space, and observation space respectively
$\mathcal{T},\mathcal{O},\mathcal{R}$	Transition model, observation model, and reward model respectively
b	Belief
π	Policy
γ	Discount factor
Notations for model discretization	
L_s	Length of states for discretization
L_a	Length of actions for discretization
N	Number of states sampled from a discrete state

3.1 | Fishery population and catch models

We focus on the classical discrete–time dynamics model proposed by Beverton and Holt (1957) in this work. This will be used to model the true environment in our simulation. In the Beverton–Holt model, the natural growth of the population biomass is described by

$$B_{t+1} = f(B_t;K,\rho) = \frac{\rho K B_t}{(\rho-1)B_t + K}, \tag{1}$$

where, B_t is the biomass at a discrete time t, ρ is a species-dependent constant known as the proliferation rate, and K is the carrying capacity or the maximum population biomass that the environment can sustain.

When fishing is taken into account, the amount of catch c_t at a discrete time t is determined by current population biomass B_t and the effort e_t (e.g., number of boat days) via

$$c_t = g(B_t,e_t;q) = qe_tB_t, \tag{2}$$

where, q is a constant called the catchability constant. The population biomass then evolves according to $B_{t+1} = f(B_t;K,\rho) - g(B_t,e_t;q)$.

Our simulation study considers both the well-specified setting and misspecified setting. In the well-specified setting, the true model is assumed to be in the model class. In reality, this assumption is generally not satisfied. That is, the true model is not in the model class, and the model class is said to be misspecified. We thus also investigate the effect of model misspecification by learning a non-Beverton–Holt model. Various alternative population dynamics models have been proposed and studied in the literature. The surplus production models (Hilborn & Walters, 1992) are a popular class of models that aggregate all aspects of production (recruitment, growth, and mortality) into a single term representing the increase in population biomass. A commonly used surplus production model is the Pella–Tomlinson model (Pella & Tomlinson, 1969)

$$B_{t+1} = f(B_t; K, r) = B_t + \frac{r}{m-1} B_t \left(1 - \left(\frac{B_t}{K}\right)^{m-1}\right), \tag{3}$$

where, r is called the intrinsic growth rate, and m is a shape parameter with larger m associated with faster growth. Typically, m is chosen to be 2 in fisheries studies and the resulting model is called the Schaefer model (Schaefer, 1954, 1957). We thus focus on the Schaefer model in our simulation study.

3.2 | Partially observable Markov decision processes

Partially Observable Markov Decision Processes (POMDPs) provide a mathematical framework for modelling uncertain environments. A POMDP is a 7-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{Z}, T, O, R, \gamma)$. Here $\mathcal{S}$ is a set of states, $\mathcal{A}$ a set of actions, $\mathcal{Z}$ a set of observations, and $\gamma \in [0,1)$ is a discount factor. At each time step, the agent executes an action $a \in \mathcal{A}$, and the current state $s \in \mathcal{S}$ transitions to the next state $s' \in \mathcal{S}$ according to the transition model $T(s' \mid s,a)$. The agent cannot observe the state directly, but receives an observation $z \in \mathcal{Z}$ according to the observation model $O(z|s',a)$. In addition, it receives a reward in accordance with the reward function $R(s,a)$ at the same time. See Figure 1 for a schematic illustration of the process.

The agent keeps track of the information about the current state by maintaining a belief b, which is a probability distribution on the state space. After taking an action a and receiving an observation z, the belief can be updated as

$$\tau(b,a,z)(s') = \frac{O(z|s',a)p(s' \mid a,b)}{p(z \mid a,b)}, \tag{4}$$

where, $p(s' \mid a,b) = \sum_{s \in S} T(s' \mid s,a)b(s)$, and $p(z \mid a,b) = \sum_{s' \in S} O(z|s',a)p(s' \mid a,b)$.

In a POMDP, an agent acts according to a policy π, which is a mapping from a belief to a distribution on the action space. That is, if the current belief is b, then the agent executes an action sampled from $\pi(b)$. The performance of a policy π is often measured using the value function $V_\pi(b_0)$, which is the expected total discounted reward obtained by following the policy π starting from an initial state s_0 sampled from the initial belief b_0:

$$V_\pi(b_0) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) | b_0\right], \tag{5}$$

where, s_t is the state at time step t with s_0 randomly drawn from b_0, a_t is the action taken by π at this time step, and $\gamma \in (0,1)$ is the discount factor. The value function satisfies the Bellman equation (Smallwood & Sondik, 1973):

$$V_\pi(b) = \sum_a \pi(a|b) \sum_z \left(\sum_s b(s)R(s,a) + \gamma p(z \mid b,a) V_\pi(\tau(b,a,z))\right), \tag{6}$$

where, $\pi(a|b)$ is the probability that action a is taken by π at a belief b.

The aim is to find an optimal policy π^*, which yields maximum value for each belief. In other words, for each belief b, $V_{\pi^*}(b) = \max_\pi V_\pi(b)$. The optimal policy can be obtained by maximizing the value function, that is, $\pi^* = \arg\max_\pi V_\pi(b_0)$. The optimal value function $V^*(b)$ satisfies the Bellman optimality equation

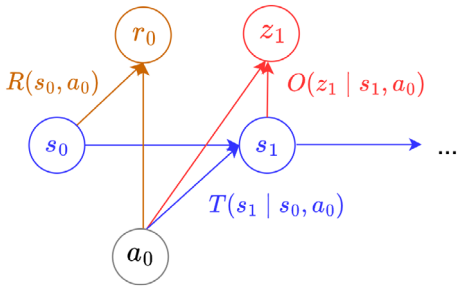


FIGURE 1 Schematic illustration of a POMDP.

$$V^*(b) = \max_a \sum_z \left(\sum_s b(s)R(s,a) + \gamma p(z \mid b,a) V^*(\tau(b,a,z)) \right). \quad (7)$$

In general, there are no closed-form or simple formula for the optimal policy π^*, and various algorithms have been designed to compute the optimal policy. However, for a long time, POMDP solution algorithms could only solve toy problems with a few states, and the potential of the POMDP framework for decision-making under uncertainty was not fully realized. Fortunately, in the last two decade, researchers have developed a few efficient algorithms that can scale up to very large POMDPs (Kurniawati et al., 2008; Silver & Veness, 2010; Ye et al., 2017).

The solution algorithms can be categorized into offline solvers and online solvers. An offline solver searches for a complete optimal policy first before executing it in the real environment. On the other hand, an online solver interleaves policy search and policy execution. It computes a policy on the fly when the agent interacts with the real environment: at each time step, the solver computes the optimal action for the current belief, executes it in the real environment, receives an observation and a reward, then update the belief, and moves on to the next time step. The online approach has been more successful at scaling up to large POMDPs because it does not need to compute a complete policy. We use an efficient online solver (Ye et al., 2017) in this paper.

3.3 | Reinforcement learning

In reinforcement learning (RL), an agent aims to find an optimal policy by interacting with the environment and exploiting the feedback from the interactions. RL provides a powerful general framework for learning to make decisions, and it has recently been successfully applied to many challenging applications, such as game playing (Mnih et al., 2015; Silver et al., 2017) and automated guided vehicle (Sierra-Garcia & Santos, 2022).

RL research has mostly focused on the online setting, where the agent is able to interact with the environment directly and learn the policy according to the immediate feedback. However, in some disciplines such as healthcare (Shortreed et al., 2011) and natural management, it is dangerous, expensive or impractical to learn decisions by direct interactions. For these situations, it is more appropriate to use offline RL (Levine et al., 2020), where the agent learns an optimal policy solely based on historical interactions.

RL algorithms can be classified as model-free algorithms or model-based algorithms, depending on whether an environment model is learned. Model-free algorithms do not learn an environment model, and they are often simple, but require a large number of interactions to learn a good policy (Mnih et al., 2015; Silver et al., 2017). On the other hand, model-based algorithms learn an environment model, and learn an optimal policy for the learned environment model, at the same time. Model-based algorithms are more complex because an environment model needs to be learned too, but it has recently become popular because of its data efficiency, stability and explainability (Moerland et al., 2020). In addition, an environment model can be used to improve the sample efficiency of RL by using it to generate additional experiences (Andersen et al., 2021; Sutton, 1990).

We take a model-based offline RL approach for sustainable fishery management in this paper. Our algorithm is relatively straightforward, but we would like to highlight that many interesting works have been done on model-based offline RL in the AI community for Markov Decision Processes (MDP) (Levine et al., 2020), which are the fully observable analogue of POMDPs where states are observed. For example, one recent work considers learning a pessimistic MDP and computing a near-optimal in the learned MDP so as to achieve robust performance (Kidambi et al., 2020). It will be interesting to explore extending such works for MDPs to POMDPs.

4 | MOOR FOR SUSTAINABLE FISHERY MANAGEMENT

The overall POMDP-based framework used in our study is shown in Figure 2, which include steps for learning a policy from catch and effort data, and steps of evaluating a policy in the ground truth model. We use an online solver as mentioned in Section 3.2, thus we do not explicitly compute a complete policy first and then simulate it to compute its value. Instead, we interleave policy search and policy evaluation as shown in the Policy Search and Evaluation box in the figure. We present our policy learning algorithm in the remainder of this section, and present more details on performance evaluation in Section 5.

4.1 | Overview of MOOR

We describe our model-based offline reinforcement learning approach in this section. An overview of our algorithm is shown in Algorithm 1. Our algorithm is called MOOR, which stands for Model-based Offline Reinforcement learning algorithm for sustainable fishery management. In the first step, we impute missing effort values if any, as detailed in Section 4.3. In the second step, we learn a fishery POMDP $POMDP(\theta)$ with

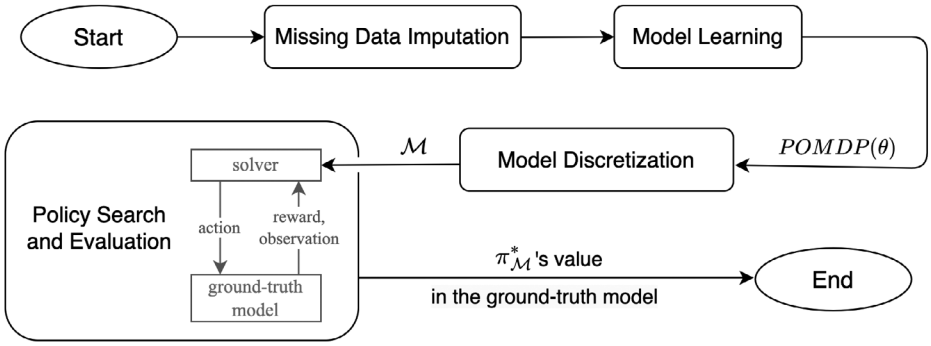


FIGURE 2 The overall POMDP-based solution and evaluation framework used in this paper. A discrete POMDP fishery model $\mathcal{M}$ is learned from the catch and effort data, then an online solver is used to compute an optimal policy $\pi^*_{\mathcal{M}}$ for $\mathcal{M}$, and the policy is evaluated in the ground-truth environment. Note that $\pi^*_{\mathcal{M}}$ is computed on the fly when it is being evaluated, as explained in detail in the main text.

ALGORITHM 1 MOOR

Input: Catch data c_t, effort data e_t, $t = 1,2,...T$, and $t \notin T_{missing}$

1 Impute e_t for each $t \in T_{\text{missing}}$

2 $POMDP(\theta) \leftarrow \text{ModelLearning}(c_t, e_t)$

3 $\mathcal{M} \leftarrow \text{ModelDiscretization}(POMDP(\theta))$

4 Find an optimal policy $\pi^*_{\mathcal{M}}$ for $\mathcal{M}$

parameters θ using the imputed data, where $\theta = (\rho, K, B_0, q)$ if the population dynamics is modelled using a Beverton–Holt model, and $\theta = (r, K, B_0, q)$ if a Schaefer model is used (see details in Section 4.2). Our learning algorithm, detailed in Section 4.4, is designed so that it can handle missing catch values in the imputed data. The learned POMDP model has continuous states, actions and observations, and is hard to solve. In the third step, we discretize the POMDP model learned in the second step (Section 4.5), so that we can exploit efficient solvers for discrete POMDPs in the last step.

At a high level, our approach above follows the same model-based offline reinforcement learning approach proposed in our preliminary study (Ju et al., 2021). However, there are a few important differences. First, we introduce an imputation scheme that aims to minimize the amount of imputation needed by imputing missing effort values only. Second, while our previous model learning objective is defined for learning deterministic models using complete data, we have extended the learning objective to handle stochastic models and incomplete data. Third, in the presence of model misspecification, we need to handle the discretization of both the groundtruth model and the learned model carefully to ensure that they share the same action space and observation space. This is necessary for ensuring that we can evaluate the performance of the learned policy with respect to the groundtruth model, because the action computed based on the learned model needs to be executable in the groundtruth model, and the observation produced by the groundtruth model needs to be a defined observation in the learned model.

4.2 | POMDPs for sustainable fishery management

We describe how POMDPs provide a natural framework for sustainable fishery management. Our POMDP model takes into account uncertainties such as unknown biomass, stochastic population dynamics, and stochastic catch process.

Specifically, the state is the unknown population biomass B at each time step, the action is the fishing effort e (e.g., fleet capacity or the numbers of vessel-days), and both the observation and the reward are the amount of catches c. The transition model combines a natural growth model f and catch model g

$$B_{t+1} = h(B_t, e_t, \varepsilon_t; K, \rho, q) = (f(B_t; K, \rho) - g(B_t, e_t; q))e^{\varepsilon_t}, \tag{8}$$

where the random noise $\varepsilon_t \sim N(0, \sigma^2)$ is introduced to model stochasticity in the population dynamics. The observation model and the reward model are both the catch model Equation (2).

Our POMDP model will be called a BH-POMDP if the dynamics model f is a Beverton–Holt model, and an SP-POMDP if the dynamics model f is a surplus production model. The key parameters of a BH-POMDP are (ρ, K, B_0, q), and those for an SP-POMDP are (r, K, B_0, q).

4.3 | Missing data imputation

We consider the case when both catches and efforts are missing for certain years $T_{missing}$. Naturally, we may try to impute both missing catches and efforts. However, as we shall see in Section 4.4, we can sidestep imputing missing catches, and still learn a model using such incomplete data.

Various data imputation approach for sequential data can be used to impute the missing values, such as imputation by regression, moving average of neighbours, and median or average of the trajectory (Zhang, 2016). We use a very simple imputation approach in this work: for each missing effort value, impute it as the average of its nearby available values. In practice, more sophisticated imputation methods may be needed. For our simulation study, this simple imputation scheme suffices for providing accurate estimates for the missing effort values, because our simulated effort values do not change abruptly, thus averages of the nearby values are close to the true values.

4.4 | Model learning

We describe how we can learn a fishery POMDP model using the imputed fishery data. We focus on learning a BH-POMDP with parameters (ρ, K, B_0, q). The same approach can be used to learn a SP-POMDP with parameters (r, K, B_0, q).

4.4.1 | Model learning objective

We define the following sum of squared error (SSE) loss for measuring how well the parameters (ρ, K, B_0, q) agree with the catch and effort data:

$$L(\rho, K, B_0, q) = \mathbb{E}\left[\sum_{t \in [T] \setminus T_{missing}}^{T} \left(g(B_t, e_t; q) - c_t\right)^2\right], \tag{9}$$

where, $[T]$ is a shorthand notation for $\{1, \ldots, T\}$, $T_{missing}$ is the set of time steps with missing data, and expectation is taken wrt the stochasticity in the evolution of B_t, as prescribed in the population dynamics model. Intuitively, the loss is the expected total squared difference between the predicted catches and the observed catches.

There are a few important things to note about the loss function. First, we do not calculate the difference for time steps with missing data, thus we do not need to impute the missing catch values. However, we still need to impute the missing effort values, because they are needed for computing the biomasses at the following time steps. In addition, the loss cannot be computed exactly in general, making it a hard objective to minimize.

4.4.2 | Model learning algorithm

We can learn the parameters by performing stochastic gradient descent (SGD) to avoid the need for computing the exact objective function values. We first sample a few biomass trajectories $B_1^{(i)}, \ldots, B_T^{(i)}$ for $i = 1, \ldots, N$, by simulating the fishery dynamics model with the given effort data. We can then compute an unbiased Monte Carlo estimate for the objective function as

$$\widehat{L}(\rho, K, B_0, q) = \frac{1}{N}\sum_{i=1}^{N}\left[\sum_{t \in [T] \setminus T_{missing}}^{T} \left(g\left(B_t^{(i)}, e_t; q\right) - c_t\right)^2\right]. \tag{10}$$

Now, SGD requires the gradient of $\widehat{L}$. While deriving an expression for the gradient is hard, we can compute it easily using automatic differentiation. This has been incorporated as a standard tool in several software that have been popular in ecology research, such as the AD Model Builder (ADMB) (Fournier et al., 2012) and its improvement Template Model Builder (TMB) (Kristensen et al., 2015). Automatic different is also a standard component in various deep learning libraries, such as PyTorch (Paszke et al., 2017), which we used in our implementation.

We note that our fishery POMDPs can be viewed as recurrent neural networks (RNN), as shown in Figure 3. The inputs are effort data, the hidden states are population biomasses and the outputs are catches. Hidden states are updated by the transition model (Equation 8) and outputs are predicted by the observation model (Equation 2).

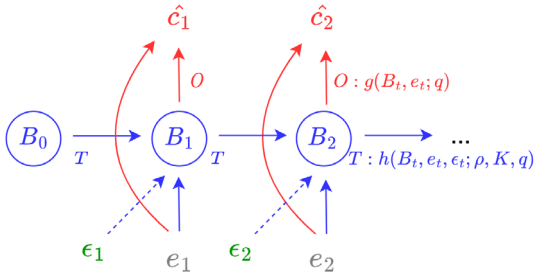


FIGURE 3 Representation of a fishery POMDP as an RNN.

Unfortunately, our experiments show that SGD does not work due to two challenges. First, the four parameters have completely different scales in practice, so it is difficult to optimize all four parameters at the same time using the same hyperparameters (e.g., learning rates) as their derivatives will be of different sizes. Second, the final solution is sensitive to the initial solution, and SGD is often trapped in a poor local minimum.

To deal with these challenges, we turn to the L-BFGS (Liu & Nocedal, 1989) and combine it with a few tricks. L-BFGS is chosen for two reasons: it is a quasi-Newton method capable of exploiting the surface geometry of the objective function, as in the case of Newton's method; at the same time, it only requires the computation of gradients but not Hessians, which are more costly to compute. Note that in our implementation, we are in fact using a stochastic version of L-BFGS. To be specific, our stochastic L-BFGS algorithm differs from the original L-BFGS algorithm only in one aspect: the original L-BFGS algorithm computes the exact gradient of the parameters in the loss function Equation (9), which is difficult, while we use the gradient of the estimated loss function (Equation 10) as an efficient stochastic approximation of the exact gradient. Readers interested in the details of L-BFGS are referred to the excellent description in the book of Nocedal and Wright (2006). Our first trick addresses the difficulty caused by different scales of the parameters. To this end, we compute normalized effort $\widetilde{e}_i = e_i / e_{\max}$ and normalized catch $\widetilde{c}_i = c_i / c_{\max}$ to $[0,1]$, where $e_{\max}$ and $c_{\max}$ are the maximum effort and catch respectively. We then work with the objective defined for the normalized data: $\widetilde{L}(\rho, K, B_0, q) = \mathbb{E}\left[\sum_{t \in [T] \setminus T_{missing}}^{T} \left(g(B_t, \widetilde{e}_t; q) - \widetilde{c}_t\right)^2\right]$. With the catch and effort data on the same scale, we find that a good initialization strategy for optimizing the objective $\widetilde{L}$ is to start with all parameters having values close to 1. A minimizer $\left(\overline{\rho}^*, \overline{K}^*, \overline{B}_0^*, \overline{q}^*\right)$ for $\widetilde{L}$ can be converted to a minimizer $\left(\rho^*, K^*, B_0^*, q^*\right)$ for the original objective L using

$$\rho^* = \overline{\rho}^*, K^* = \overline{K}^* c_{\max}, B_0^* = \overline{B}_0^* c_{\max}, q^* = \overline{q}^* / e_{\max}. \tag{11}$$

Our second trick aims to avoid poor local minimum. We simply use the standard trick of performing multiple runs with different random initializations around 1. We choose a model whose SSE is the smallest.

4.5 | Model discretization

Finding an optimal policy for a continuous POMDP model is generally difficult. We thus discretize the learned continuous POMDPs so as to exploit recent state-of-the-art solvers for discrete POMDPs. Our discretization algorithm is shown in Algorithm 2, with details explained below.

We first discretize state, observation and action spaces. Each discrete state represents a biomass interval of length L_s. Each discrete action represents the midpoint of an effort interval of length L_a. The discrete observation space is the same as the discrete state space.

We then discretize transition, observation and reward models using Monte Carlo simulation method proposed by Filar et al. (2019). Specifically, we first perform simulations to obtain a dataset D of (s, a, s', z) tuples as follows: sample N biomass state for each discrete state $\bar{s}$, then for each sampled s and each discrete action a, simulate the transition model (Equation 8) and the observation model (Equation 2) to obtain the next state s' and the observation/reward z. Using the dataset D, we can then calculate $T(\bar{s}'|\bar{s}, a)$ and $O(\bar{z}|\bar{s}', a)$ as their empirical probabilities for each discrete states $\bar{s}$, $\bar{s}'$, discrete observation $\bar{z}$ and discrete action a. The reward function $R(\bar{s}, a)$ is calculated as the average of rewards for (s, a) such that $s \in \bar{s}$.

5 | SIMULATION STUDY

We perform systematic simulation study to assess our approach's performance by quantitatively assessing the effects of stochasticity in fishery dynamics, population growth rate, missing values in fishery data, model misspecification, and effort variability. We describe our experimental

ALGORITHM 2 Discretization of continuous fishery POMDP models

Input: Model parameters (θ), number of samples N, transition model (Equation 8), observation model (Equation 2)

1 Determine discretized state, observation and action spaces

2 Uniformly sample N states from each discretized state intervals

3 For each sampled continuous state s and each discrete action a, generate the tuple (s,a,s',z) where s' is the next state sampled according to Equation (8), and z is computed by Equation (2)

4 Calculate $T(\bar{s}'|\bar{s},a)$, $O(\bar{z}|\bar{s}',a)$ and $R(\bar{s},a)$ using the dataset D of generated (s,a,s',z) tuples:

$T(\bar{s}'|\bar{s},a) = |\{(s,a,s',z) \in D : s \in \bar{s}, s' \in \bar{s}'\}| / n$

$O(\bar{z}|\bar{s}',a) \propto |\{(s,a,s',z) \in D : s \in \bar{s}, s' \in \bar{s}', z \in \bar{z}\}|$

$R(\bar{s},a) = \text{averageof}\{z : (s,a,s',z) \in D, s \in \bar{s}\}$

setup in Section 5.1 and present the results in Section 5.2. In addition, we present a comparison between our model learning algorithm and the Bayesian method in Section 5.3.

5.1 | Experiment settings

This section present the details for the individual steps in our policy learning and evaluation framework shown in Figure 2.

5.1.1 | Data generation

We implement our own simulator to generate the effort and catch data. We perform our experiments on three types of data: (a) deterministic data, which is generated by a deterministic model; (b) stochastic complete data, which is generated by a stochastic model but contains no missing values; and (c) stochastic incomplete data, which is generated by a stochastic model but contains missing values.

We choose Beverton–Holt model as the natural growth model in the ground truth models (GT-POMDP). Three types of environments are considered: low ($\rho = 1.3$), medium ($\rho = 2.0$) and high ($\rho = 3.0$) proliferation rate. We choose $K = 10{,}000$, $B_0 = 5000$ and $q = 0.005$ for all ground truth models, and use $\sigma = 0$ for deterministic datasets and $\sigma = 0.1$ for stochastic datasets. For each environment, we evaluate our algorithm's performance on 3 random effort and catch datasets, so as to assess the sensitivity of our algorithm's performance on the particular dataset used.

Each dataset spans $T = 50$ time steps. The effort time series is first generated according to a piecewise function which first increases up to a certain time step t_a and then stays at a constant level:

$$e_t = \begin{cases} \left(\frac{\beta - \alpha}{t_a} t + \alpha\right) e^{\phi_t} & \text{if } 0 < t \le t_a \\ \beta e^{\phi_t} & \text{if } t_a < t \le T \end{cases}, \tag{12}$$

where, $\phi_t \sim N(0, 0.05)$, $t_a = 0.7T$, $\alpha = 0.05/q$, and $\beta = 0.9/q$, in our experiments. Note that the maximum effort is $\frac{1}{q}$, which corresponds to 100% harvest rate and causes extinction of the species. Thus α corresponds to 5% harvest rate and β corresponds to a 90% harvest rate. Figure 4a displays an example of generated effort value series. After generating the effort values, we generate the catch values by simulating the fishery population and catch dynamics as described in Section 4.2.

We use more variable effort values as compared to our prior work (Ju et al., 2021), which uses effort values sampled from $N(10, 3)$ (equivalently, a harvest rate fluctuating around 5%), as illustrated in Figure 4b. This is because the effort values need to be sufficiently variable for successful model and policy learning in a stochastic environment, as we shall see in Section 5.2.5.

Incomplete data is generated by removing the catches and efforts for a moderate number of 10 randomly chosen years from a complete dataset for 50 years. This allows us to quantify how missing data degrades the performance of our algorithm. We note that in reality, data may be missing for consecutive years, and a more sophisticated imputation method may be needed for our method to perform well.

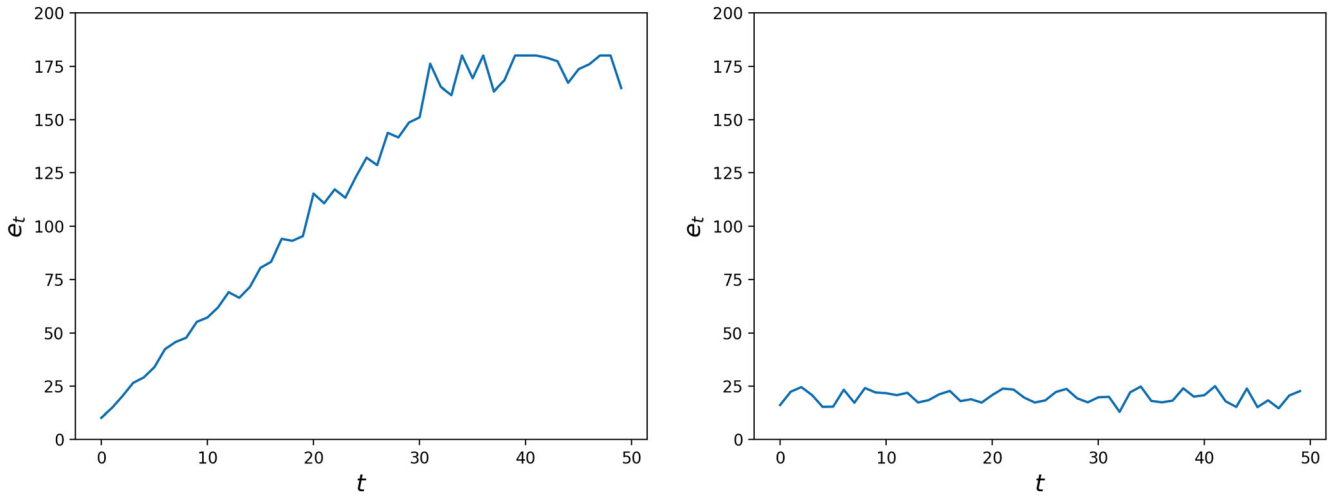


FIGURE 4 Effort values of different levels.

5.1.2 | Hyperparameters for model learning

We try L-BFGS with all combinations of learning rates {0.05, 0.1, 0.2, 0.3, 0.4, 0.5} and number of iterations {5, 10, 15, 20}, running the algorithm 30 times starting from different random initial parameters for each combination. We pick the model with the minimum SSE from all these trials.

For each dataset, we learn both deterministic and stochastic versions of well-specified and mis-specified models. For stochastic models, $\sigma = 0.1$ and $N = 5$ sampled biomass trajectories are used for computing the stochastic gradient. For deterministic models, $\sigma = 0$ and just $N = 1$ sampled biomass trajectory is needed for computing the exact gradient as there is no randomness.

5.1.3 | Hyperparameters for model discretization

Each model is discretized using Algorithm 4.5 with $L_s = 1000$, $L_a = 15$, and $N = 1000$.

5.1.4 | Hyperparameters for policy search and policy evaluation

We use DESPOT (Ye et al., 2017) as the POMDP solver to search for the optimal policy for learned models. We run DESPOT for 500 simulations of 100 steps with $\gamma = 0.95$. The action for each step is computed within 0.1 s for the current belief and the learned model, and then executed in the ground-truth model. The average total discounted reward over the 500 simulations is reported as the estimated value of the optimal policy for the learned model when it is executed in the ground-truth model.

5.2 | Results and analysis

We first present our model and policy learning results on different datasets, and then discuss how they are affected by factors including stochasticity of fishery dynamics, proliferation rates, missing values, model misspecification and effort variability.

Our model and policy learning results are shown in Figures 5–7, which shows the results on deterministic data, stochastic data, and stochastic incomplete data respectively. Each figure displays the performance of the models and policies learned on the corresponding dataset. For model learning, we compare the learned population dynamics with the groundtruth by plotting their natural growth curves of $B_{t+1} = f(B_t; K, \cdot)$ against B_t, and we compare the learned parameters against the true parameters. In addition, we plot the predicted expected catches and the observed catches to assess how the learned model fits the training data. The predicted expected catch series is the average of 1000 predicted catch series. For policy learning, we report policy values for groundtruth models and learned models. The policy value is the expected total discounted reward of a 100-time-step period over 500 simulations. We also draw a threshold line for 85% of optimal policy values

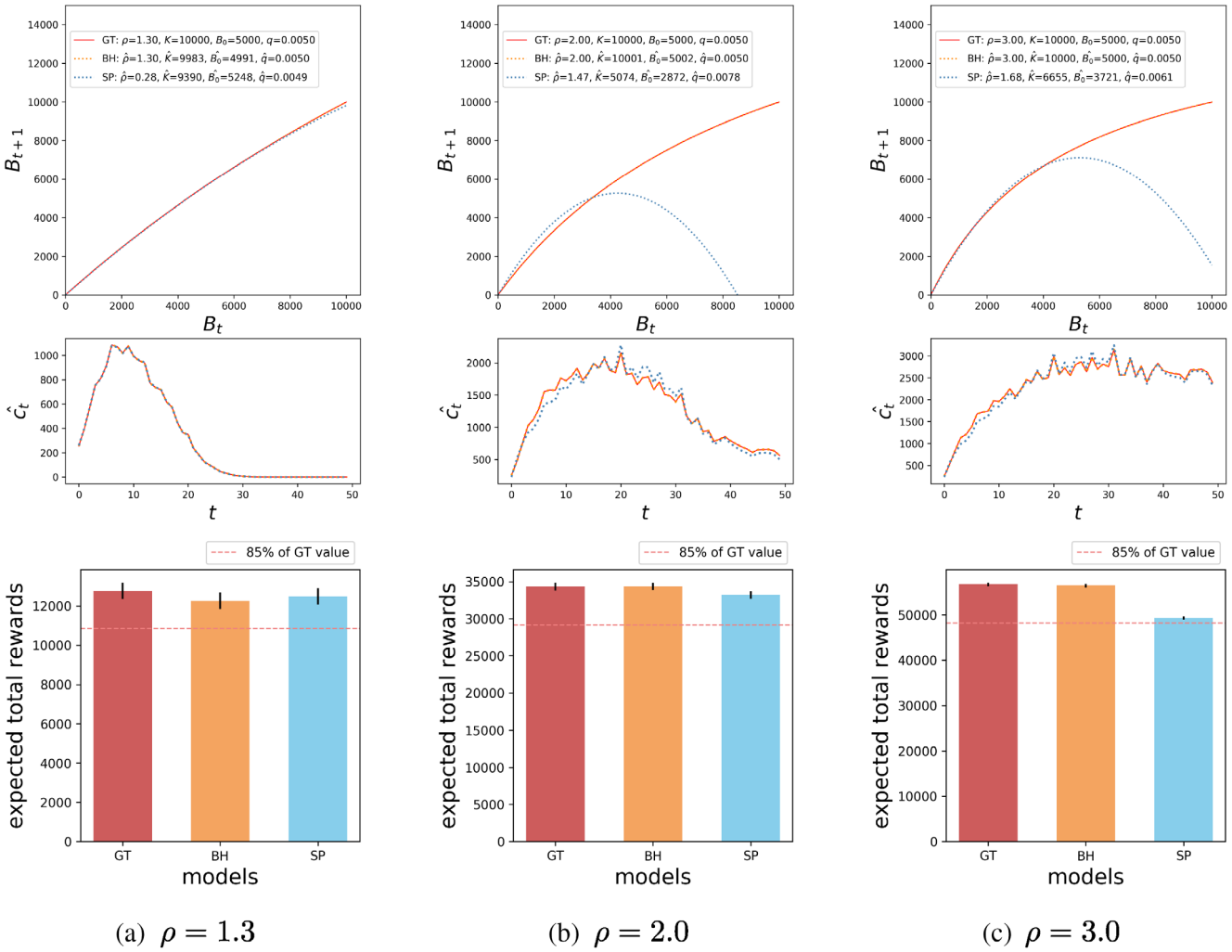


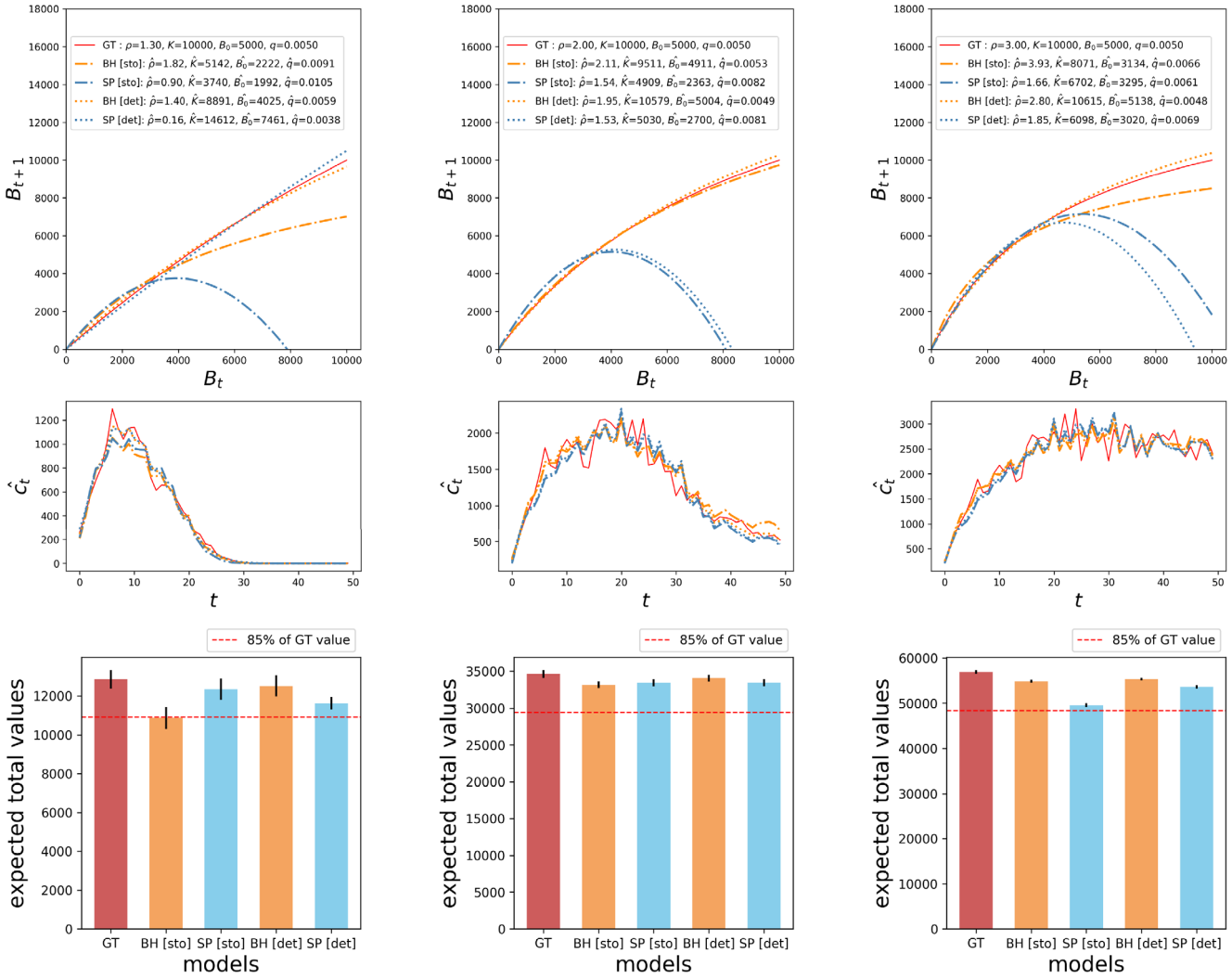
FIGURE 5 Model and policy learning results on deterministic datasets. Top: $B_{t+1} = f(B_t; K; \cdot)$ against B_t for groundtruth and learned models. Middle: true catches and predicted catches using learned learned models. Bottom: values of learned policies. GT, BH, and SP refer to GT-POMDP, learned BH-POMDP and learned SP-POMDP models respectively. det is used to highlight that the models are deterministic.

for groundtruth models to compare with the policies for learned models in each environment. We focus on the results on one random dataset in this section, but present additional results on different random datasets in Appendix A. Note that the conclusions are qualitatively the same though.

As a baseline, we first consider the case of learning well-specified models when the ground truth model is deterministic. From Figure 5, we can see that our algorithm learns very accurate well-specified models (the BH-POMDP models) and near-optimal policies. The parameters of the learned BH-POMDP models are almost equal to the true values in GT-POMDP. This leads to near identical natural population dynamics curves for the learned BH-POMDP models to the corresponding GT-POMDP models, and the predicted catches are nearly the same as the observed catches.

We also make some brief comments on learning a misspecified model. It is interesting to observe that from Figure 5, we can still fit the effort-catch data well and we can obtain very good policies using the learned misspecified SP-POMDP models. In addition, when $\rho = 1.3$, the population dynamics of SP-POMDP and the ground truth model are nearly identical, and the parameters (K, B_0, q) are correctly identified. When $\rho = 2.0$ and $\rho = 3.0$, the learned values of (K, B_0, q) and the learned population dynamics appears to be very different from the true one. It may appear puzzling why a large difference in the population dynamics, a misspecified model still produces very good policies. This is because only the lower biomass region of the population dynamics is relevant, and the learned dynamics is functionally similar to the true one in that region. More discussions on the effect of model-misspecification are in Section 5.2.4.

We analyse the effects of various factors on model and policy learning below.



(a) $\rho = 1.3$ (b) $\rho = 2.0$ (c) $\rho = 3.0$

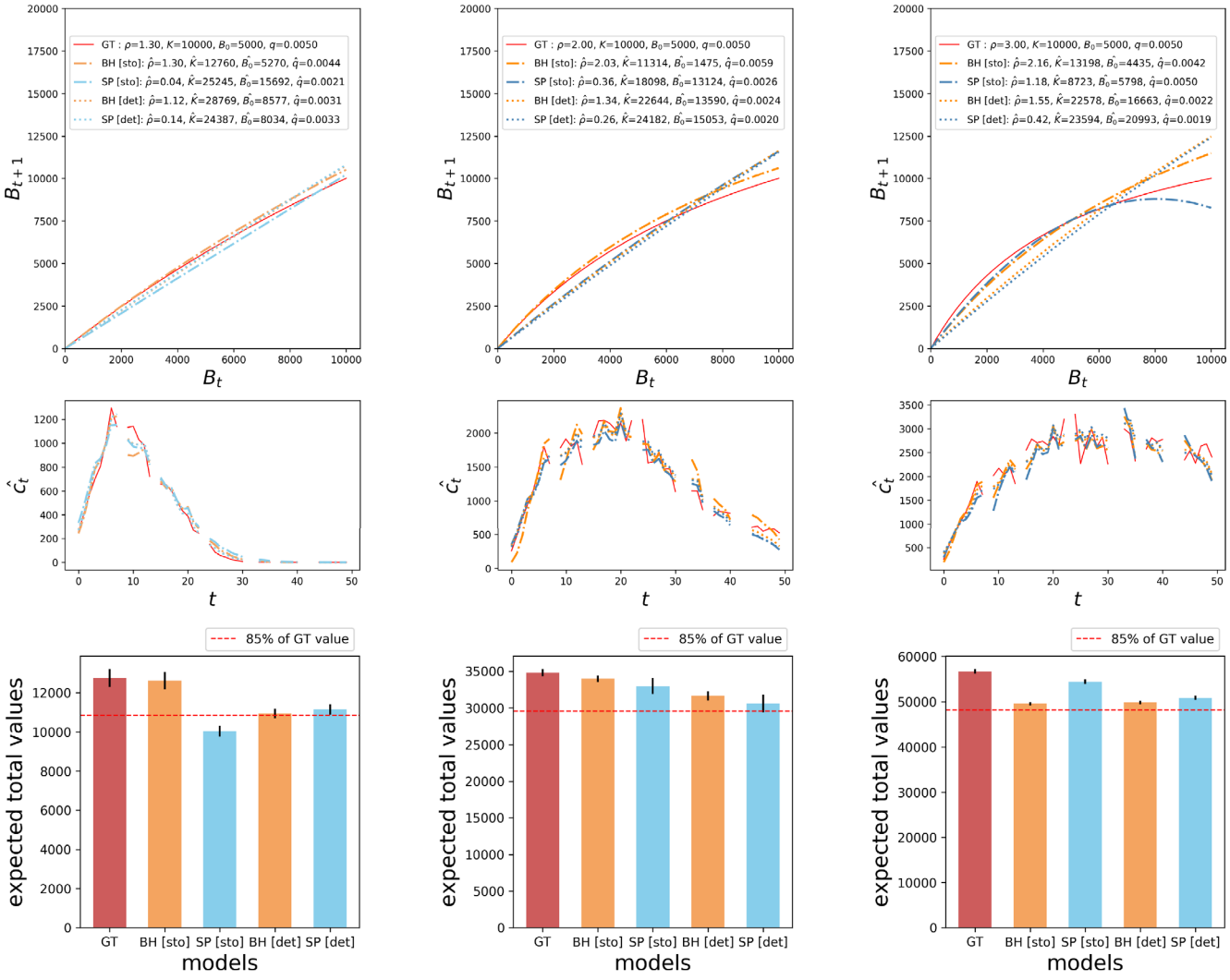
FIGURE 6 Model and policy learning results on stochastic complete datasets. Top: $B_{t+1} = f(B_t; K; \cdot)$ against B_t for groundtruth and learned models. Middle: true catches and predicted catches using learned learned models. Bottom: values of learned policies. GT, BH, and SP refer to GT-POMDP, learned BH-POMDP and learned SP-POMDP models respectively. sto indicates a stochastic model, and det a deterministic one.

5.2.1 | Effects of stochasticity in fishery dynamics

To understand how stochasticity in fishery system affect our approach, we compare the results for deterministic fishery systems in Figure 5 and the results for stochastic fishery systems in Figure 6.

Unsurprisingly, the quality of the learned models are more different from the ground truth models in terms of both parameter values and population dynamics, whether the models are well-specified or misspecified, and whether the models are deterministic or stochastic. However, it is noteworthy that while the learned parameter values can be quite different from the true ones, the learned population dynamics are still functionally similar to the true population dynamics. In addition, the learned policies are still competitive with the optimal policies despite relatively larger model learning errors as compared to the results on the deterministic data. All learned policies achieve at least 85% of the values of the optimal policies. This is possible because for the purpose of obtaining a good policy, we do not need to learn the population dynamics exactly, but we just need to learn the relevant region of the population dynamics well. These results also suggest that model learning is a harder problem than policy learning.

While it is hard to learn the carrying capacity K and the initial biomass B_0 in presence of stochasticity, the ratio K/B_0 is generally correctly identified, which is also true in the deterministic case as K and B_0. One possible explanation for the difficulty of learning exact values of K, B_0 and



(a) $\rho = 1.3$ (b) $\rho = 2.0$ (c) $\rho = 3.0$

FIGURE 7 Model and policy learning results on stochastic incomplete datasets. Top: $B_{t+1} = f(B_t; K; \cdot)$ against B_t for groundtruth and learned models. Middle: true catches and predicted catches using learned learned models. Bottom: values of learned policies. GT, BH, and SP refer to GT-POMDP, learned BH-POMDP and learned SP-POMDP models respectively. sto indicates to a stochastic model, and det a deterministic one.

q is that when q increases/decrease, we can decrease/increase K and B_0 to obtain very similar fishery dynamics. In particular, the catch equation $c_t = qe_tB_t$ suggests that very roughly, when the products of q with K and B_0 are constant, the fishery dynamics is likely to be similar.

5.2.2 | Effects of proliferation rates

It seems more difficult to learn the true parameter values when $\rho = 1.3$ as compared to when $\rho = 2.0$ and $\rho = 3.0$, whether the data is deterministic (Figure 5) or stochastic (Figure 6). This is more so in the case of stochastic data, where the learned parameters and population dynamics are quite different from the true ones, where the model is well-specified or misspecified model.

The difficulty of learning a good model for a fishery with low proliferation rate is likely due to the fishery's vulnerability to collapse. To elaborate, note that in Figure 5, we observe that from around $t = 30$ the red catch curve flattens out to zero, thus the fishery has collapsed. When the fishery collapses, the data provides little information on the true fishery dynamics. However, the effect of this is mainly that we do not learn much about the true fishery dynamics in the large biomass region, but the data still provides information on the low biomass region. This is why we only observe large differences between the population dynamics in the large biomass region.

Interestingly, the learned parameters can be more accurate if we learn a deterministic model. This suggests that sacrificing the need to model uncertainty can possibly make it easier to learn about the key parameters for the fishery system.

5.2.3 | Effects of missing values

While learning from stochastic incomplete data is expected to be the most challenging, we observe from Figure 7 that the learned policies are still very competitive with the optimal ones in general, with similar performance to the cases of deterministic data and stochastic complete data.

Unsurprisingly, the errors in the learned parameters are somewhat larger than the case of stochastic complete data. In addition, while the ratio K/B_0 can be recovered for complete data and stochastic data, this is not the case for stochastic incomplete data. Instead, the learned models generally have lower growth rates, but larger B_0 and K, and B_0 is closer to K.

Nevertheless, the population dynamics of the learned models are still reasonably close to the true ones, whether the learned model is well-specified or misspecified, deterministic or stochastic.

5.2.4 | Effects of model misspecification

We provide some further discussion on the effect of model specification in this section. While the set of parameters of a misspecified model are different from the set of parameters in the ground truth model, we observe that the learned population dynamics of a misspecified model appears to be functionally similar to the ground truth in the low biomass region, which is probably the region the models operate in. In addition, a misspecified model can still fit the dataset well and produce high-quality policies—Figure 8 provides policy learning results on additional stochastic datasets to support this. This is important as we typically work with misspecified models in practice. However, this observation is likely only valid when the misspecified model is capable of approximating the true model well in the functionally relevant region.

While the intrinsic growth rate r in the Schaefer model and the proliferation ρ in the Beverton–Holt model measure the growth behaviour of a species differently, we derive an approximate relationship between them and use it to better understand the performance of misspecified models. Specifically, given a Beverton–Holt model with parameters (ρ, K, B_0, q), we want to find a Schaefer model with parameters (r, K, B_0, q) such that they are roughly equivalent in the operating region. We focus on the region where the Beverton–Holt model is operating in an equilibrium state with maximum catch. This happens when the population increase $\frac{\rho BK}{(\rho-1)B+K} - B$ is maximized and the catch is equal to the population increase, or equivalently, when

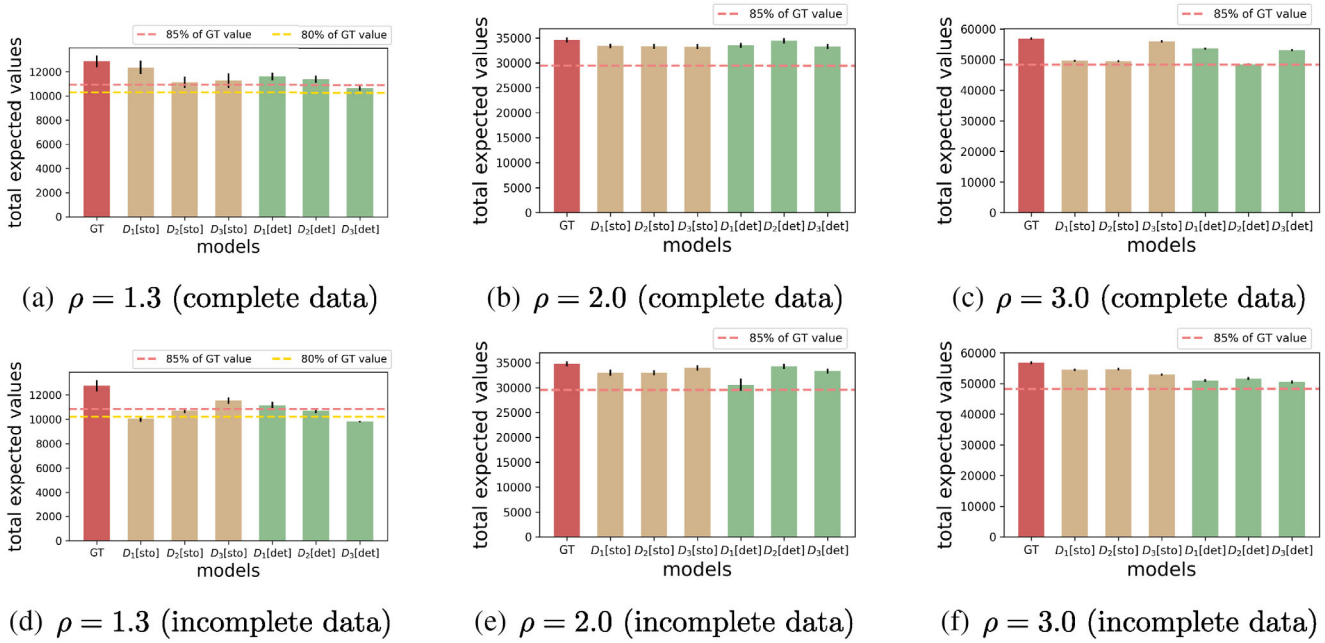


FIGURE 8 Policy learning results of misspecified models. Top: stochastic complete data (Figure 6 is for D_1). Bottom: stochastic incomplete data (Figure 7 is for D_1). D_i refers to ith simulated dataset in each situation. sto indicates to a stochastic model, and det a deterministic one.

$$\left(\frac{\rho BK}{(\rho-1)B+K}-B\right)=0,$$
$$\frac{\rho BK(\rho-1)-\rho K\left[(\rho-1)B+\rho K^2\right]}{[(\rho-1)B+K]^2}-1=0.$$

Solving the equation gives

$$B=\frac{K}{\sqrt{\rho}+1}.$$

The population increase is $\frac{\sqrt{\rho}-1}{\sqrt{\rho}+1}K$ in this case. Now for the Schaefer model to achieve this amount of increase when $B=\frac{K}{\sqrt{\rho}+1}$, we need to choose r such that

$$r\frac{K}{\sqrt{\rho}+1}\left(1-\frac{K}{\sqrt{\rho}+1}\right)=\frac{\sqrt{\rho}-1}{\sqrt{\rho}+1}K.$$

Solving the equation gives

$$r=\frac{\rho-1}{\sqrt{\rho}}. \tag{13}$$

In short, a Beverton–Holt model (ρ,K,B_0,q) operating under its maximum sustainable yield biomass is equivalent to a Schaefer model $\left(\frac{\rho-1}{\sqrt{\rho}},K,B_0,q\right)$ operating under the same biomass. In fact, the population dynamics of these two models often appear to be very similar. We can express ρ in terms of r as

$$\rho=\left(\frac{r+\sqrt{r^2+4}}{2}\right)^2. \tag{14}$$

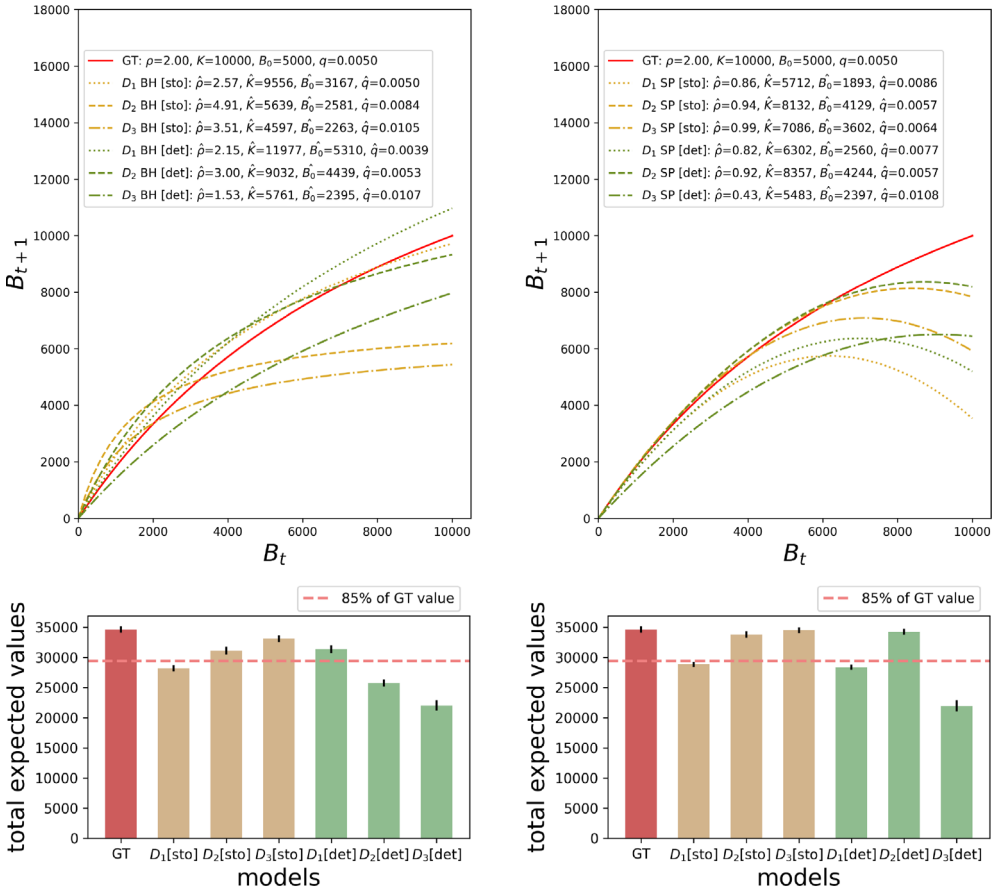
In Figure 5, when $\rho=1.3$, the learned r value is 0.28, and $2\left(\frac{r+\sqrt{r^2+4}}{2}\right)^2=1.32$, which gives a very accurate estimate for the true proliferation rate. When $\rho=2.0$, the learned r value is 1.47, and $\left(\frac{r+\sqrt{r^2+4}}{2}\right)^2=3.90$, which is quite different from the true value. This is because the learned K value in SP-POMDP is only about half of the true value, and this is consistent with the observed large differences between the SP-POMDP population dynamics and the ground truth population dynamics. Overall, whether the learned value of r in the missspecified SP-POMDP estimates the growth behaviour of the species well depends on whether K is correctly learned as well.

5.2.5 | Effects of effort variability

An important factor in a good policy performance is that there is a significant amount of effort variability in our datasets: the effort values roughly range from 5% harvest rate to 90% harvest rate, as described in Section 5.1. With such a wide range of effort values, the data provides sufficient information on how the population dynamics works under different conditions. This allows us to learn the population dynamics reasonably well at least in the operating region encountered in the training set, as seen in the discussions above.

While we obtained good models and policies using nearly constant effort values in our previous work (Ju et al., 2021), this is because the ground truth model is deterministic. When the ground truth model is stochastic, effort values fluctuating around a constant are no longer sufficiently probe the fishery system's behaviour. This prevents identifying the ground truth model from the data, even when using a well-specified model. In other words, without sufficient variability in the effort values, we are likely to suffer from non-identifiability.

To confirm that insufficient variability can cause identifiability issues, we learn models on stochastic complete datasets generated with efforts sampled from $N(10,3)$. We focus on the model and policy learning results in one environment of $\rho=2.0$ here (Figure 9), but the findings are qualitatively the same for $\rho=1.3$ and $\rho=3.0$, and these results are presented in Appendix B. From the population dynamics plots in Figure 9, the models learned on different datasets vary significantly even for well-specified models. In fact, even when learning on the same dataset, we observe that models with very similar training SSE can have very different population dynamics, leading to very different policies. The bottom figures of Figure 9 also illustrate that poor model learning can lead to poor policies in this case.



(a) BH-POMDPs (b) SP-POMDPs

FIGURE 9 Model and policy learning results on stochastic datasets of insufficiently variable effort data. Top: The left and the right are $B_{t+1} = f(B_t; K; \cdot)$ against B_t for learned well-specified models (BH-POMDPs) and misspecified models (SP-POMDPs) respectively. Bottom: values of learned policies. D_i refers to ith simulated dataset. GT, BH, and SP refer to GT-POMDP, learned BH-POMDP and learned SP-POMDP models respectively. sto indicates to a stochastic model, and det a deterministic one.

We perform further experiments to confirm that insufficient variability in the effort values is indeed causing a non-identifiability problem and policy degradation. We plot catches versus time steps under various constant-effort policies for $\rho = 1.3$ and $\rho = 2.0$ in Figure 10. When using effort values close to 10, the behaviour of the learned models and the ground truth model are similar (i.e., differences between equilibrium catches are smaller). The differences are larger for effort values that are far away from 10. In addition, the learned models and the ground truth model have different optimal effort levels. For example, when $\rho = 1.3$, an effort of 80 gives the largest equilibrium catch for the Beverton–Holt model, but this causes fishery collapse for the ground truth model.

5.2.6 | Discussion

We provide further discussion on the learning of SP-POMDP models in this section, and we demonstrate that SP-POMDP models are more prone to overfitting the training data, and thus more likely to learn dynamics that differ significantly from the ground truth Beverton–Holt model as compared to a BH-POMDP model.

From the results in Section 5.2, particularly those in Figures 5 and 6, the dynamics of the learned SP-POMDP models deviate significantly from the true ones. However, a Beverton–Holt model can be approximated well by a Schaefer model, as pointed out in the theoretical analysis in Section 5.2.4. Indeed, we plot the ground truth model and learned SP-POMDP model for $\widehat{r} = 1.54$ (blue dashed-dot curve for $\rho = 2.0$) in Figure 6, and also plot the Schaefer model that is predicted to be equivalent to the ground truth model, as shown in Figure 11. We can see that the Schaefer model obtained from our theoretical analysis does agree with the ground truth Beverton–Holt model very well.

We suspect that this is because the quadratic form of the Schaefer model allows it to fit the irregularities in the dataset, while the Beverton–Holt model's dynamics is in between a quadratic model and a straight line, which makes it harder to fit the irregularities. If this is the case, then

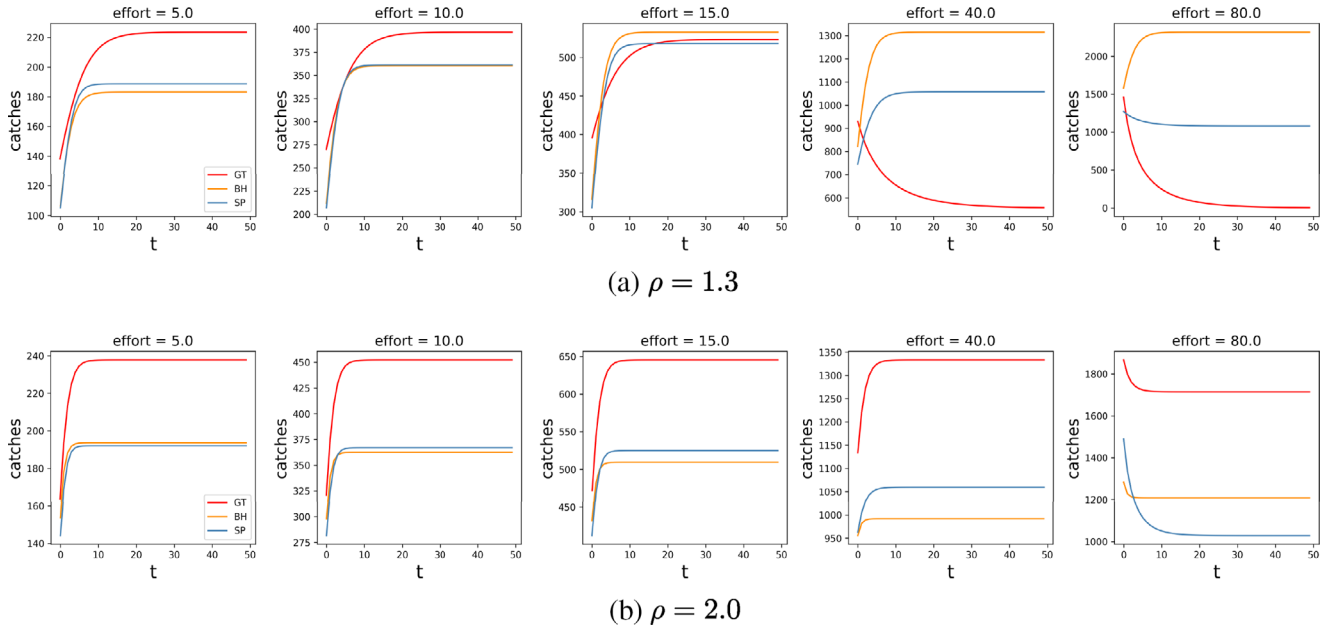


FIGURE 10 Catch series under different fixed actions. BH-POMDP and SP-POMDP are stochastic models learned on the datasets of constant level of efforts. The fixed actions are chosen around and beyond training effort values. Noise in the population dynamics is ignored when generating the plots.

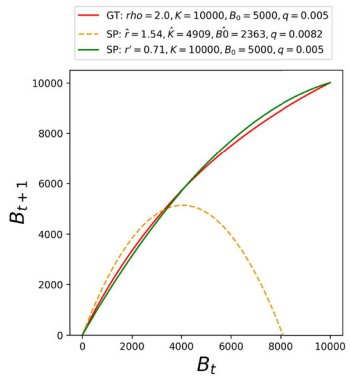


FIGURE 11 $f(B_t;K,B_0,\cdot)$ against B_t for learned SP-POMDP model $\hat{r} = 1.54$ (blue dashed-dot curve for $\rho = 2.0$ in Figure 6), corresponding ground truth model and predicted Schaefer model.

the Schaefer model should use its downward leg to fit the data in cases a poor SP-POMDP model is learned. To confirm this, we consider the models in Figure 6, and we simulate the GT-POMDPs, learned deterministic BH-POMDPs, and learned deterministic SP-POMDPs to generate their biomass time series B_t values. Note that B_{t+1} is the biomass after B_t grows and catch happens, thus we use $B_{t+1} + c_t$ as the biomass that is expected under natural growth. We plot the $B_{t+1} + c_t$ values against the B_t values when $\rho = 2.0$ and $\rho = 3.0$, as shown in Figure 12. We can see that indeed the learned Schaefer models operate in their downward legs when fitting the datasets.

5.3 | Comparison of model learning methods

We demonstrate the effectiveness of our model learning algorithm by using our model learning algorithm and the Bayesian method to learn a Beverton–Holt model and a Schaefer model on the stochastic dataset in Figure 6.

For the Bayesian method, we use the parameters $\boldsymbol{\theta} = [\rho, K, \psi, q]$ for the Beverton–Holt model and the parameters $\boldsymbol{\theta} = [\rho, K, \psi, q]$ for the Schaefer model, where $\psi = \frac{B_0}{K}$ is known as initial biomass depletion. We use the following weak prior: $\rho \sim U[1,6]$, $K \sim LogNormal\left(\ln(8c_{\max}), 5^2\right)$, $\psi \sim U[0,1]$, $q \sim U[0,0.1]$, and $r \sim U[0,6]$. The likelihood model is given by a modified version of Equation (8) where the term $f\text{-}g$ is replaced by $\max(f\text{-}g, 10^{-9}K)$ to keep the biomass positive, and a modified version of Equation (2) where the right hand side is multiplied with a log-normal

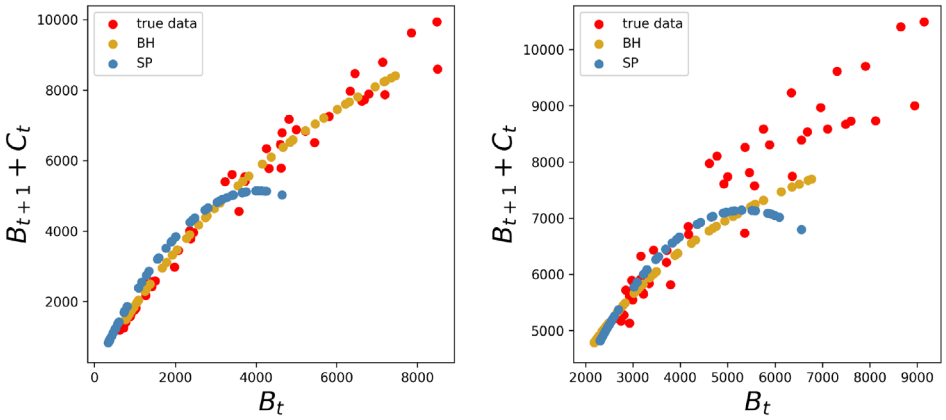


FIGURE 12 $(B_{t+1}+c_t)$ against B_t for stochastic complete data when $\rho = 2.0$ and $\rho = 3.0$. Predicted data are from BH-POMDP and SP-POMDP using deterministic approximation.

TABLE 2 Parameter values and SSEs for the models learned using our algorithm and the Bayesian method on the stochastic dataset in Figure 6.

		Well-specified POMDP (BH-POMDP)					Misspecified POMDP (SP-POMDP)				
ρ	Methods	$\hat{\rho}$	$\hat{K}$	$\hat{B}_0$	$\hat{q}$	SSE	$\hat{r}$	$\hat{K}$	$\hat{B}_0$	$\hat{q}$	SSE
1.3	Ours	1.823	5142	2222	0.00911	8687.56	0.903	3740	1992	0.0105	8642.20
	Bayesian	1.333	5545	5046	0.00436	143,689	0.440	4805	4101	0.00584	28,630
2.0	Ours	2.105	9511	4911	0.00534	56,070.80	1.541	4909	2363	0.00824	53,573.52
	Bayesian	1.981	10,281	5044	0.00496	53,512	1.527	15,952	9853	0.00261	8,642,418
3.0	Ours	3.933	8071	3134	0.00661	106,357.38	1.664	6702	3295	0.00609	122,927.36
	Bayesian	3.458	8913	4276	0.00584	107,930	1.111	10,180	5786	0.00436	143,689

noise e^{ε}. Here $\varepsilon \sim N(0,\sigma_n^2)$ and σ_n^{-2} has a Gamma prior with shape 4 and rate 0.01. We use JAGS (Plummer, 2003) to perform Bayesian inference (4 chains, 640,000 samples for each chain with half of the samples for burnin). The model modifications are necessary in JAGS.

We report the learned parameter values and the expected SSE as defined in Equation (9) in Table 2. We use posterior means as the learned values for the Bayesian method. The Bayesian method has significantly larger SSE than our method, particularly for smaller ρvalues. Note that when $\rho = 2.0$ and the model is misspecified, the model estimated by the Bayesian method often leads to fishery collapse. The Bayesian method yields parameters that are closer to the true parameters when $\rho = 3$.

6 | CONCLUSION

This paper proposes a model-based offline reinforcement learning approach for sustainable fishery management. Our approach leverages recent advances in POMDP solution algorithms, neural network learning, and model-based offline reinforcement learning. We perform systematic simulation study to quantify the effects of data quality, model misspecification, and proliferation rates. Our results suggest that when the effort levels are sufficiently variable, our method can produce competitive policies, even for the hardest case of noisy incomplete data and a misspecified model. Interestingly, the learned policies seem to be robust in the presence of model learning errors. Our results and analysis suggest that this is because the data has allowed learning the part of the population dynamics that is relevant for computing the optimal policy, even though the learned model parameters are very different and thus the learned dynamics differ significantly from the true ones in its entirety. However, non-identifiability kicks in if there is insufficient variability in the effort levels and the fishery system is stochastic. This often results in poor policies, highlighting the need for sufficiently informative data.

Our work provides a promising approach for adaptive management with a number of possible extensions. First, it can be extended to handle alternative types of data, such as catch and CPUE data. Second, it is also feasible to learn alternative fishery POMDP models, such as those in which catch is not linearly related to the biomass. Third, our imputation scheme imputes effort data only to minimize the amount of imputation,

but imputing catch data by regression and calculating the same objective function as complete data may still help when missing values are concentrated in a certain period. Lastly, it is helpful to develop an interpretable policy to make it simpler for fishery management experts to validate the proposed strategy.

ACKNOWLEDGEMENTS

We thank Jerzy Filar for many helpful discussions. Jun Ju thanks Yeming Lei for helpful discussions on Bayesian stock assessment and JAGS. This work is partially supported by the Australian Research Council (ARC) Discovery Project 200101049 and the ARC Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS, grant number CE140100049). Open access publishing facilitated by The University of Queensland, as part of the Wiley - The University of Queensland agreement via the Council of Australian University Librarians.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflicts of interest.

DATA AVAILABILITY STATEMENT

This paper used synthetic data only. The data and the code are available at https://github.com/jujun622/moorfisherys.

ORCID

Jun Ju https://orcid.org/0000-0002-2961-5979

REFERENCES

Andersen, P.-A., Goodwin, M., & Granmo, O.-C. (2021). Increasing sample efficiency in deep reinforcement learning using generative environment modelling. *Expert Systems*, *38*(7), e12537.

Beverton, R. J., & Holt, S. J. (1957). *On the dynamics of exploited fish populations*. Springer Science & Business Media.

Blamey, L. K., Plagányi, É. E., Hutton, T., Deng, R. A., Upston, J., & Jarrett, A. (2022). Redesigning harvest strategies for sustainable fishery management in the face of extreme environmental variability. *Conservation Biology*, *36*(3), e13864.

Bunnefeld, N., Hoshino, E., & Milner-Gulland, E. J. (2011). Management strategy evaluation: A powerful tool for conservation? *Trends in Ecology & Evolution*, *26*(9), 441–447.

Charles, A. T. (1998). Living with uncertainty in fisheries: Analytical methods, management priorities and the Canadian groundfishery experience. *Fisheries Research*, *37*(1–3), 37–50.

Damasio, L. d. M. A., Lopes, P. F., Guariento, R. D., & Carvalho, A. R. (2015). Matching fishers' knowledge and landing data to overcome data missing in small-scale fisheries. *PLoS One*, *10*(7), e0133122.

De Lara, M., & Martinet, V. (2009). Multi-criteria dynamic decision under uncertainty: A stochastic viability analysis and an application to sustainable fishery management. *Mathematical Biosciences*, *217*(2), 118–124.

Donovan, G. P. (1989). *The comprehensive assessment of whale stocks: The early years*. International Whaling Commission.

Ellison, A. M. (2004). Bayesian inference in ecology. *Ecology Letters*, *7*(6), 509–520.

Filar, J. A., Qiao, Z., & Ye, N. (2019). Pomdps for sustainable fishery management. In S. Elsawah (Ed.), *MODSIM2019, 23rd international congress on modelling and simulation* (pp. 645–651). Modelling and Simulation Society of Australia and New Zealand.

Fournier, D. A., Skaug, H. J., Ancheta, J., Ianelli, J., Magnusson, A., Maunder, M. N., Nielsen, A., & Sibert, J. (2012). Ad model builder: Using automatic differentiation for statistical inference of highly parameterized complex nonlinear models. *Optimization Methods and Software*, *27*(2), 233–249.

Hilborn, R., & Sibert, J. (1988). Adaptive management of developing fisheries. *Marine Policy*, *12*(2), 112–121.

Hilborn, R., & Walters, C. J. (1992). *Quantitative fisheries stock assessment: Choice, dynamics and uncertainty*. Springer Science & Business Media.

Ju, J., Kurniawati, H., Kroese, D., & Ye, N. (2021). Moor: Model-based offline reinforcement learning for sustainable fishery management. In R. W. Vervoort, A. A. Voinov, J. P. Evans, & L. Marshall (Eds.), *MODSIM2021, 24th international congress on modelling and simulation* (pp. 771–777). Modelling and Simulation Society of Australia and New Zealand.

Kidambi, R., Rajeswaran, A., Netrapalli, P., & Joachims, T. (2020). Morel: Model-based offline reinforcement learning. *Advances in Neural Information Processing Systems*, *33*, 21810–21823.

Kristensen, K., Nielsen, A., Berg, C. W., Skaug, H., & Bell, B. (2015). Tmb: Automatic differentiation and Laplace approximation. *arXiv Preprint arXiv: 1509.00660*.

Kurniawati, H., Hsu, D., & Lee, W. S. (2008). Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces. In *Robotics: Science and systems* (Vol. 2008). Citeseer.

Lane, D. E. (1989). A partially observable model of decision making by fishermen. *Operations Research*, *37*(2), 240–254.

Levine, S., Kumar, A., Tucker, G., & Fu, J. (2020). Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv: 2005.01643*.

Liu, D. C., & Nocedal, J. (1989). On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, *45*(1), 503–528.

Magnusson, K. G., & Stefánsson, G. (1989). A feedback strategy to regulate catches from a whale stock. *by GP Donovan. Reports of the International Whaling Commission (Special Issue 11)* (pp. 171–189).

Matsuzaki, S.-i. S., & Kadoya, T. (2015). Trends and stability of inland fishery resources in Japanese lakes: Introduction of exotic piscivores as a driver. *Ecological Applications*, *25*(5), 1420–1432.

Memarzadeh, M., Britten, G. L., Worm, B., & Boettiger, C. (2019). Rebuilding global fisheries under uncertainty. *Proceedings of the National Academy of Sciences of the United States of America*, *116*(32), 15985–15990.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533.

Moerland, T. M., Broekens, J., & Jonker, C. M. (2020). Model-based reinforcement learning: A survey. *arXiv Preprint arXiv:2006.16712*.

Nocedal, J., & Wright, S. (2006). *Numerical optimization*. Springer Series in Operations Research and Financial Engineering.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., & Lerer, A. (2017). Automatic differentiation in pytorch. In *NIPS 2017 workshop autodiff*.

Pella, J. J., & Tomlinson, P. K. (1969). A generalized stock production model. *Inter-American Tropical Tuna Commission*, *13*(3), 421496.

Plummer, M.(2003). Jags: A program for analysis of bayesian graphical models using gibbs sampling. In *Proceedings of the 3rd international workshop on distributed statistical computing* (Vol. 124, pp. 1–10). Vienna, Austria.

Punt, A. E., Butterworth, D. S., de Moor, C. L., De Oliveira, J. A., & Haddon, M. (2016). Management strategy evaluation: Best practices. *Fish and Fisheries*, *17*(2), 303–334.

Punt, A. E., & Hilborn, R. (1997). Fisheries stock assessment and decision analysis: The bayesian approach. *Reviews in Fish Biology and Fisheries*, *7*, 35–63.

Ricard, D., Minto, C., Jensen, O. P., & Baum, J. K. (2012). Examining the knowledge base and status of commercially exploited marine species with the ram legacy stock assessment database. *Fish and Fisheries*, *13*(4), 380–398.

Rudd, M. B., & Branch, T. A. (2017). Does unreported catch lead to overfishing? *Fish and Fisheries*, *18*(2), 313–323.

Sainsbury, K. J., Punt, A. E., & Smith, A. D. (2000). Design of operational management strategies for achieving fishery ecosystem objectives. *ICES Journal of Marine Science*, *57*(3), 731–741.

Schaefer, M. B. (1954). Some aspects of the dynamics of populations important to the management of the commercial marine fisheries. *Inter-American Tropical Tuna Commission*, *1*(2), 25–56.

Schaefer, M. B. (1957). A study of the dynamics of the fishery for yellow fin tuna in the eastern tropical pacific ocean. *Bullettin of the Inter-American Tropical Tuna Commission*, *2*, 247–285.

Sethi, G., Costello, C., Fisher, A., Hanemann, M., & Karp, L. (2005). Fishery management under multiple uncertainty. *Journal of Environmental Economics and Management*, *50*(2), 300–318.

Shortreed, S. M., Laber, E., Lizotte, D. J., Stroup, T. S., Pineau, J., & Murphy, S. A. (2011). Informing sequential clinical decision-making through reinforcement learning: An empirical study. *Machine Learning*, *84*(1), 109–136. https://onlinelibrary.wiley.com/doi/full/10.1111/exsy.13076

Sierra-Garcia, J. E., & Santos, M. (2022). Combining reinforcement learning and conventional control to improve automatic guided vehicles tracking of complex trajectories. *Expert Systems*, e13076.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., & Hassabis, D. (2017). Mastering the game of go without human knowledge. *Nature*, *550*(7676), 354–359.

Silver, D., & Veness, J. (2010). Monte-carlo planning in large pomdps. *Advances in Neural Information Processing Systems*, *23*, 2164–2172.

Smallwood, R. D., & Sondik, E. J. (1973). The optimal control of partially observable markov processes over a finite horizon. *Operations Research*, *21*(5), 1071–1088.

Sutton, R. S. (1990). Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Machine learning proceedings 1990* (pp. 216–224). Elsevier.

Walters, C. J. (2007). Is adaptive management helping to solve fisheries problems? *AMBIO: A Journal of the Human Environment*, *36*(4), 304–307.

Walters, C. J., & Hilborn, R. (1976). Adaptive control of fishing systems. *Journal of the Fisheries Board of Canada*, *33*(1), 145–159.

Ye, N., Somani, A., Hsu, D., & Lee, W. S. (2017). Despot: Online pomdp planning with regularization. *Journal of Artificial Intelligence Research*, *58*, 231–266.

Zhang, Z. (2016). Missing data imputation: Focusing on single imputation. *Annals of Translational Medicine*, *4*(1), 9. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4716933/

How to cite this article: Ju, J., Kurniawati, H., Kroese, D., & Ye, N. (2023). Model-based offline reinforcement learning for sustainable fishery management. *Expert Systems*, e13324. https://doi.org/10.1111/exsy.13324

APPENDIX A: ADDITIONAL RESULTS ON DIFFERENT RANDOM DATASETS

We provide results on additional stochastic complete datasets and stochastic incomplete datasets in this section. These are qualitatively similar to the results provided in the main text (Figures A1–A4).

A.1. | Stochastic complete data

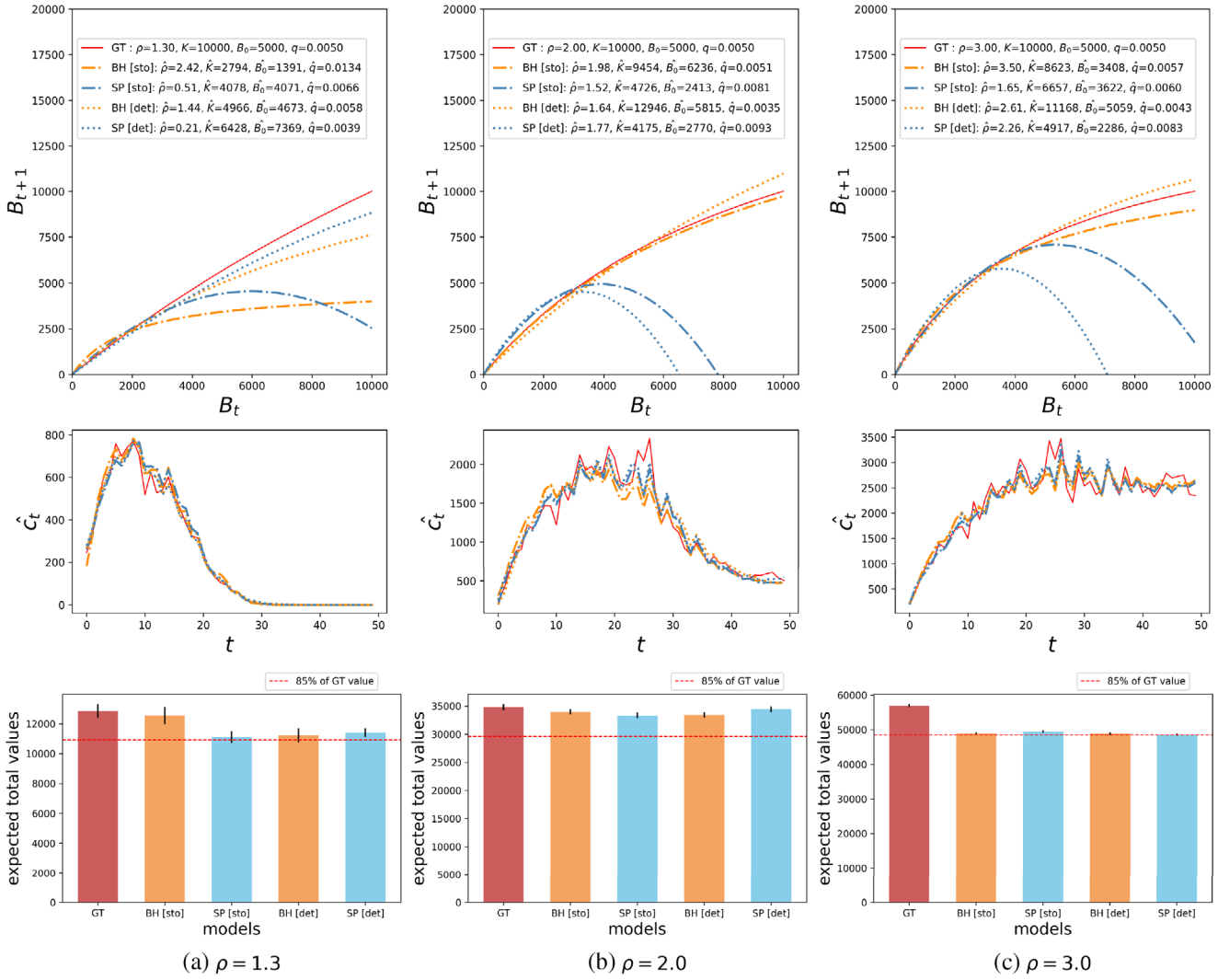
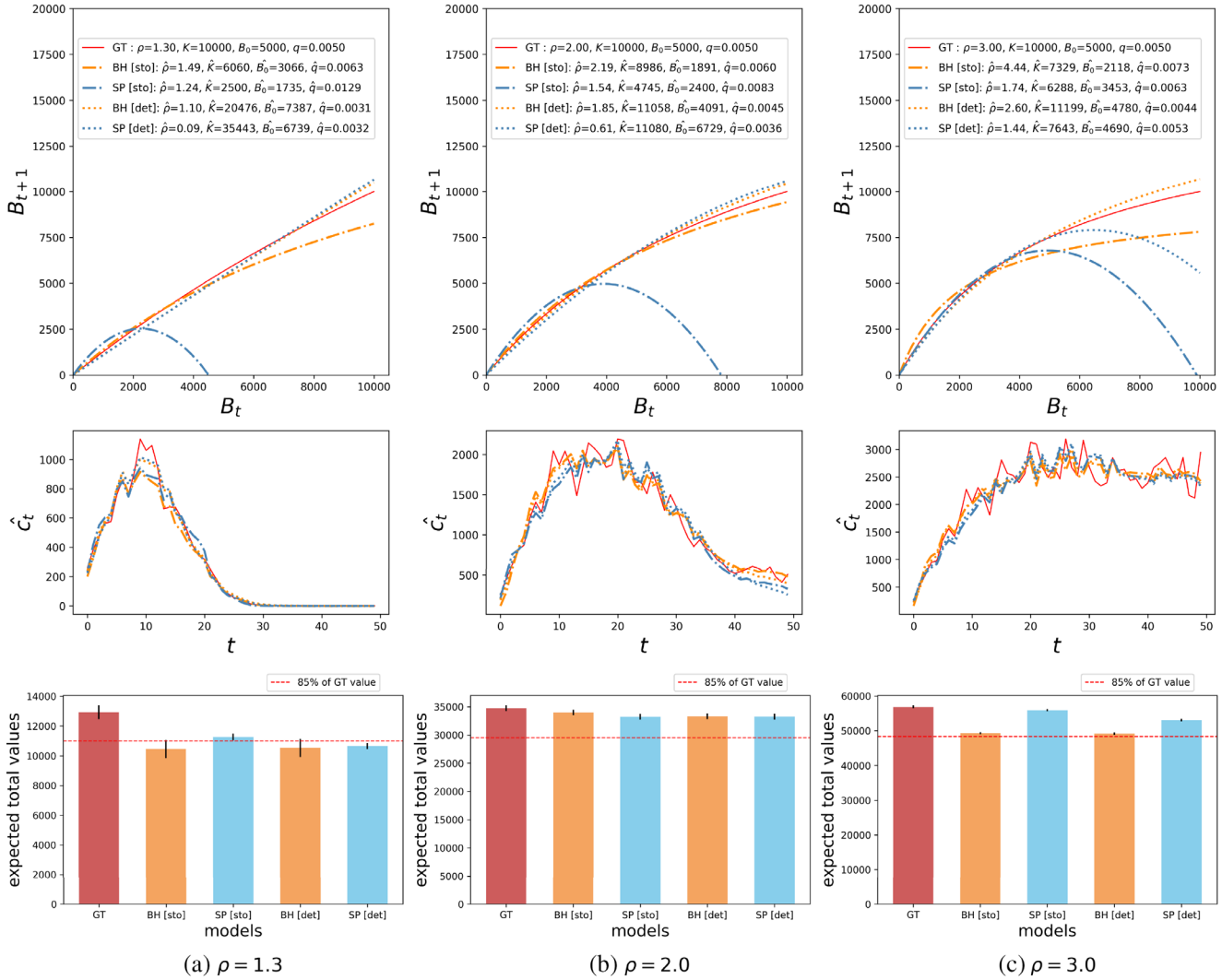


FIGURE A1 Model and policy learning results on stochastic complete dataset D_2. Top: $B_{t+1} = f(B_t; K; \cdot)$ against B_t for ground truth and learned models. Middle: true catches and predicted catches using learned learned models. Bottom: values of learned policies. GT, BH, and SP refer to GT-POMDP, learned BH-POMDP and learned SP-POMDP models respectively. sto indicates a stochastic model, and det a deterministic one.



(a) $\rho = 1.3$ (b) $\rho = 2.0$ (c) $\rho = 3.0$

FIGURE A2 Model and policy learning results on stochastic complete dataset D_3. Top: $B_{t+1} = f(B_t; K; \cdot)$ against B_t for ground truth and learned models. Middle: true catches and predicted catches using learned learned models. Bottom: values of learned policies. GT, BH, and SP refer to GT-POMDP, learned BH-POMDP and learned SP-POMDP models respectively. sto indicates a stochastic model, and det a deterministic one.

A.2. | Stochastic incomplete data

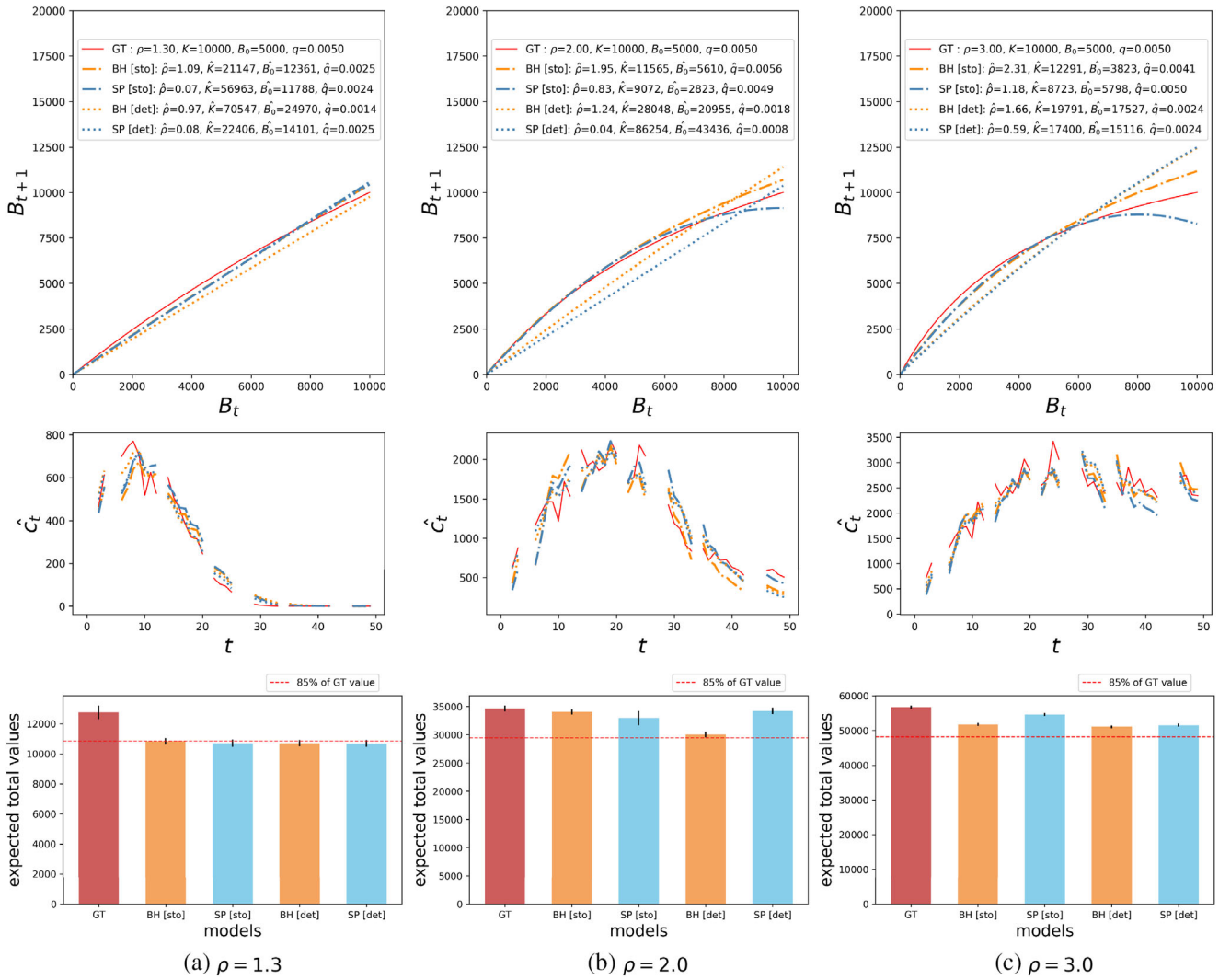
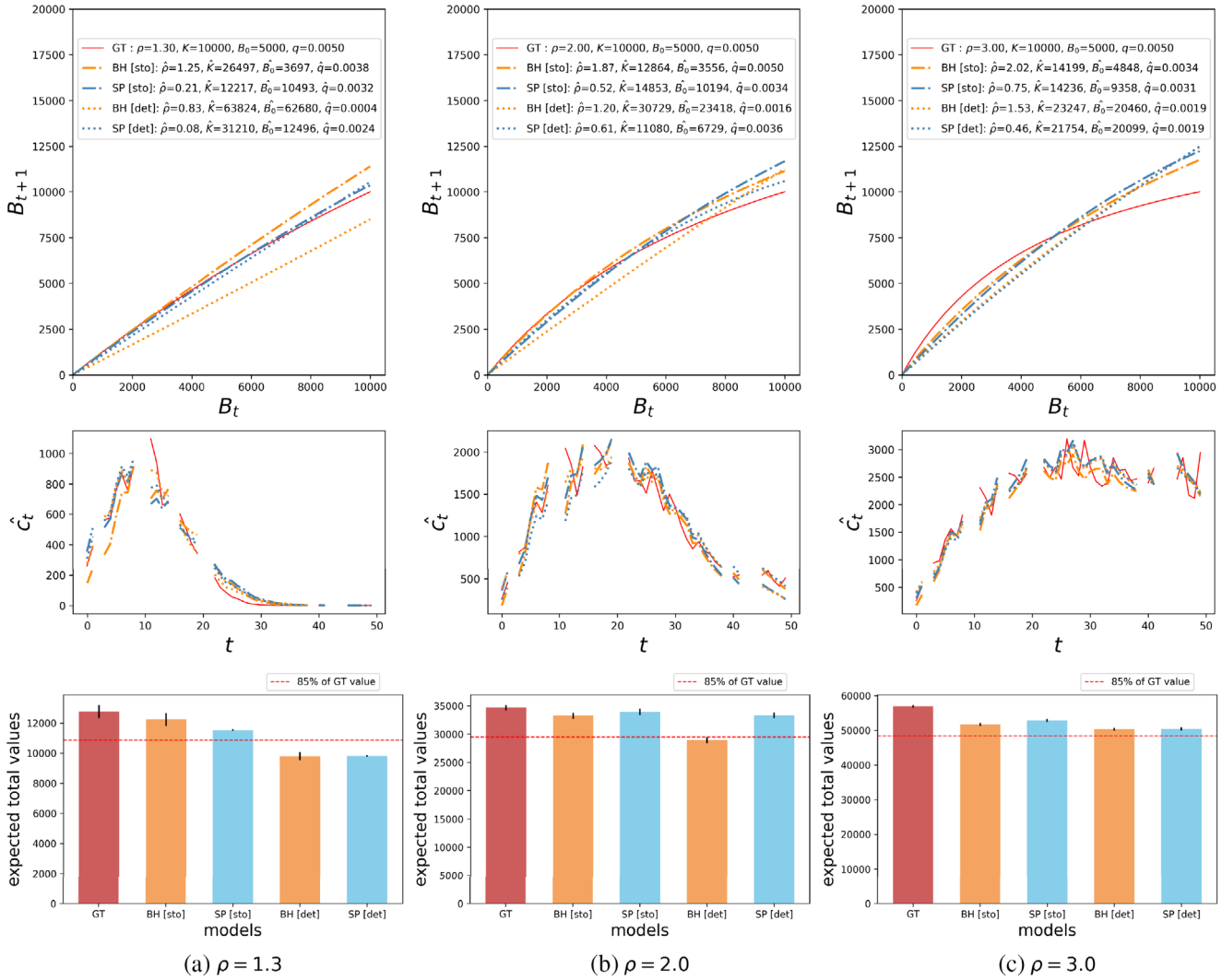


FIGURE A3 Model and policy learning results on stochastic incomplete dataset D_2. Top: $B_{t+1} = f(B_t; K; \cdot)$ against B_t for ground truth and learned models. Middle: true catches and predicted catches using learned learned models. Bottom: values of learned policies. GT, BH, and SP refer to GT-POMDP, learned BH-POMDP and learned SP-POMDP models respectively. sto indicates a stochastic model, and det a deterministic one.

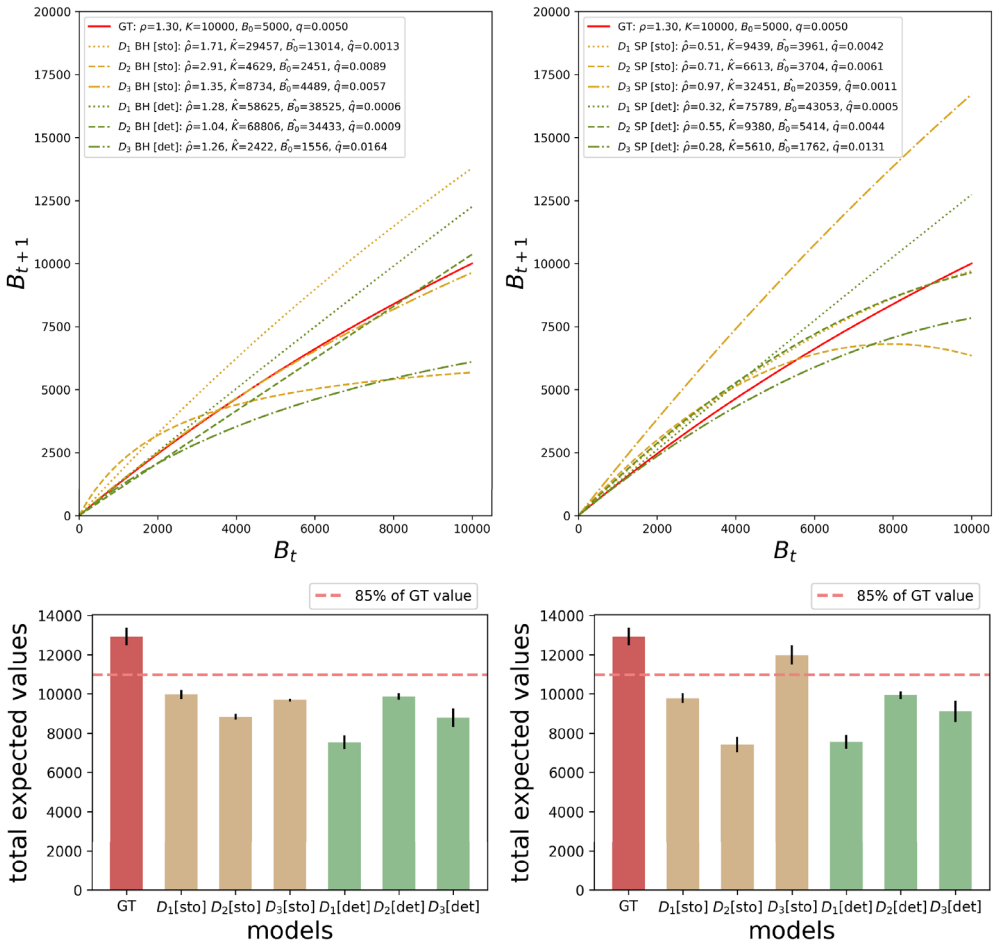


(a) $\rho = 1.3$ (b) $\rho = 2.0$ (c) $\rho = 3.0$

FIGURE A4 Model and policy learning results on stochastic incomplete dataset D_3. Top: $B_{t+1} = f(B_t; K; \cdot)$ against B_t for ground truth and learned models. Middle: true catches and predicted catches using learned learned models. Bottom: values of learned policies. GT, BH, and SP refer to GT-POMDP, learned BH-POMDP and learned SP-POMDP models respectively. sto indicates a stochastic model, and det a deterministic one.

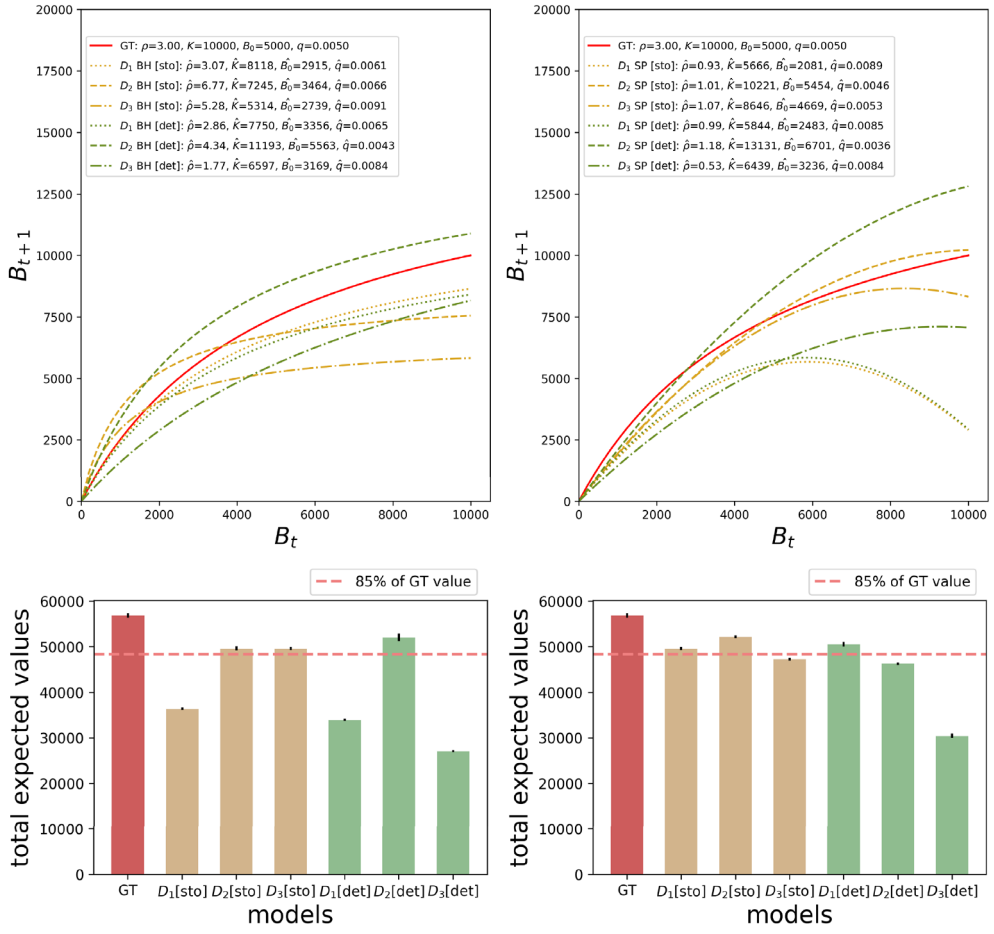
APPENDIX B: ADDITIONAL RESULTS OF INSUFFICIENTLY VARIABLE EFFORT DATA

This appendix displays model learning and policy learning results for datasets in which effort data is generated from $N(10,3)$ when $\rho = 1.3$ and $\rho = 3.0$ (Figures B1 and B2).



(a) BH-POMDPs (b) SP-POMDPs

FIGURE B1 Model and policy learning results on stochastic datasets of insufficiently variable effort data when $\rho = 1.3$. Top: $B_{t+1} = f(B_t; K; \cdot)$ against B_t for ground truth and learned models. Bottom: values of learned policies. GT, BH, and SP refer to GT-POMDP, learned BH-POMDP and learned SP-POMDP models respectively. det is used to highlight that the models are deterministic.



(a) BH-POMDPs (b) SP-POMDPs

FIGURE B2 Model and policy learning results on stochastic datasets of insufficiently variable effort data when $\rho = 3.0$. Top: $B_{t+1} = f(B_t; K; \cdot)$ against B_t for ground truth and learned models. Bottom: values of learned policies. GT, BH, and SP refer to GT-POMDP, learned BH-POMDP and learned SP-POMDP models respectively. det is used to highlight that the models are deterministic.

AUTHOR BIOGRAPHIES

Jun Ju is a Ph.D. student in the School of Mathematics and Physics of the University of Queensland. She has a Master of Data Science degree in the University of Queensland and a Bachelor of Natural Science degree in Shanghai University of International Business and Economics. Her research interest includes reinforcement learning, deep learning, decision making under uncertainty and sustainable fishery management.

Hanna Kurniawati is a professor at the School of Computing, Australian National University (ANU) and holds the SmartSat CRC Chair for System Autonomy, Intelligence, and Decision Making. Her research focuses on algorithms to enable robust decision theory to become practical software tools, with applications in robotics and the assurance of autonomous systems. Together with collaborators and students, her work has received multiple recognitions, including a best paper award at the International Conference on Automated Planning and Scheduling (ICAPS) 2015, a gold award for ICT researcher of the year 2015 from the Australian Computer Society, a keynote talk at IEEE/RSJ Int. Conf. on Intelligent Robots (IROS) 2018, and the Robotics: Science and Systems 2021 Test of Time Award.

Dirk Kroese is a professor of Mathematics and Statistics at the School of Mathematics and Physics of the University of Queensland. He has held teaching and research positions at The University of Texas at Austin, Princeton University, the University of Twente, the University of Melbourne, and the University of Adelaide. He has over 120 peer-reviewed publications, including 6 highly cited monographs. His research interests include Monte Carlo methods, adaptive importance sampling, randomized optimization, and rare-event simulation, the cross-entropy method and applied probability.

Nan Ye is a lecturer of Statistics and Data Science at the School of Mathematics and Physics of the University of Queensland. His research spans machine learning algorithms, theory, and applications. He has published on topics including sequential decision making under uncertainty, weakly supervised learning, probabilistic graphical models, statistical learning theory, in venues such as NeurIPS, ICML, ICLR, UAI, JAIR, JMLR. He received an IJCAI-JAIR Best Paper Prize in 2022, and a UAI Best Student Paper Award in 2014.