Planning with Prediction in the Natural Environment

Johanna Hansen
Mobile Robotics Lab
Centre for Intelligent Machines
McGill University
Advised by Prof. Gregory Dudek
johanna.hansen@mail.mcgill.ca

Contents

Preprint. Work in progress.

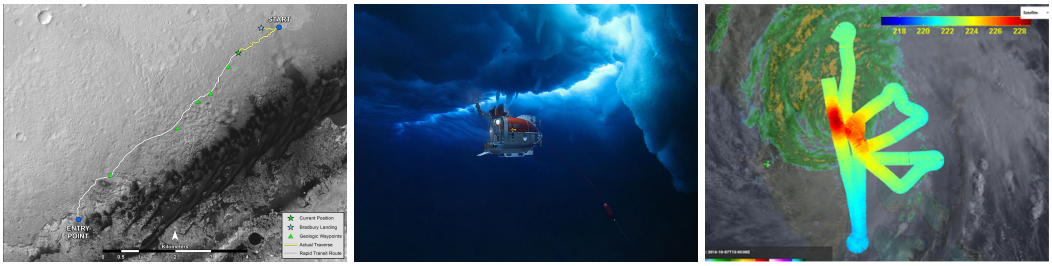
1 Introduction

The natural world is a complex dynamic system which humans have only begun to understand in a quantitative way. How does flooding in urban areas impact the ecosystem of a nearby coastal reef? Does the texture of the underside of sea ice dictate microbial life? How is phytoplankton in Arctic lakes effected by climate change? These are just a few samples of the types of questions that scientists are attempting to answer. In this thesis proposal, I introduce my plans to develop tools for improving our understanding of the world. They broadly break down into two parts: 1) learn models based on existing data that explain variation over space and time and 2) create robot behaviours based on these models which efficiently collect meaningful new data points in a cost-effective way.

1.1 Motivation

Mobile scientific sampling robots, such as those depicted in Figure 1, are regularly deployed to harsh environments to gather data for scientific discovery and monitoring. These robots are equipped with specialized sensors travel which collect data as the robot travels to various parts of a survey region collecting *in-situ* (local) observations of a phenomenon which is changing over a spatial and/or temporal scale.

Despite their accomplishments and sophistication, the "autonomous" robots in Figure 1 each require an entire team of professional engineers and technicians to help plan and prioritize their activity (Silver, 2010). Though *truly* autonomous robots have the opportunity to autonomously fill important holes in scientific exploration and data acquisition, at this time most autonomous scientific mobile vehicles only perform tasks under the watchful eye of experts. Today's robots are hindered by their lack of understanding of global factors which impact both the phenomenon that they are



(a) Extraterrestrial Sampling (b) Sampling under Sea Ice (c) Sampling a Hurricane

Figure 1: This panel depicts various scientific observation robots. The images are roughly ordered by the degree to which they rely on human operators for decision making. In Fig 1a, we see an example of Curiosity's semi-autonomous exploration. Scientists back on earth choose specified waypoints of interest (marked in green in the figure) and Curiosity chooses a safe path through the waypoints (Webster, 2013). In Fig 1b, the Nereid Under-Ice (NUI) (Jakuba, 2014) vehicle is depicted performing a teleoperated mapping mission under a shifting pack of ice. Robots working in under-ice environments require advanced and reliable reasoning and localization capabilities so that they are able to perform tasks far from the base station ship and still return safely if/when the fiber optic tethered connection to the ship is lost. In Figure 1c, an unmanned aircraft is teleopererated by human experts in an adaptive manner to collect data during Hurricane Michael (JPL, 2016). This image depicts temperature of the atmosphere as collected by the drone on top of data from ground-based radar.

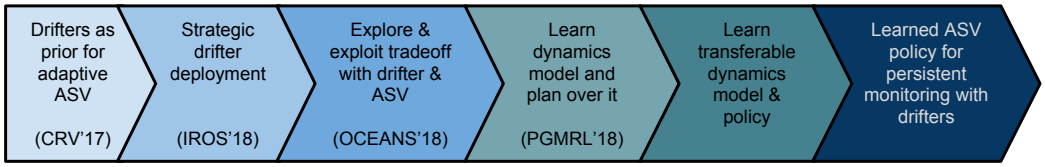


Figure 2: Major contributions of this thesis and publication where applicable. Section 4.2 discusses results in shades of blue and Section 4.1 highlights progress in turquoise.

trying to observe and the dynamic forces in the world which inhibit their ability to collect data efficiently (such as ocean currents or tides). As a result of the expense associated with modern robot supervision, exploration robots tend to only take on roles which are much too costly, tedious, and/or dangerous for humans.

Less sophisticated robots, such as those that handle boxes in warehouses or control row-driving tractors, can interact with domain experts rather than specially trained engineers, but these robots are usually confined to relatively controlled (structured) environments and specific tasks. Machines that do interact with non-expert people in human spaces have historically taken on tasks which have a low sensitivity to failure such as vacuum cleaning or lawn mowing.

1.2 Contribution

My research is concerned with developing intelligent, but low-cost sampling systems which allow long-term and ubiquitous data collection of the (primarily natural) world. One of the chief barriers preventing even expensive modern robots from being fielded outside of controlled environments without a team of experts is their inability to reason in uncertain or unstructured environments. Structured environments are those in which the space or environment is clearly defined, usually with simple shapes and controlled lighting. However, most of the real world is unstructured, with an infinite number of unknown and dynamic variables.

My research seeks to empower artificial systems and scientists alike with strong models of the natural world, so that their environment appears more structured, thanks to greater context. I propose several techniques for improving autonomy in robots by building and using models which enable agents to make predictions about the unobserved environment, allowing our systems effectively work in teams, explore an unknown region, and exploit predictable spaces. A substantial part of my work is motivated by the need to inexpensively collect samples with high spatial and temporal resolution from Canadian lakes. This mostly applied work in distributed scientific sampling is discussed in detail in Section 4.2. In Section 4.1, I present early results from my ongoing research in model-based approaches for solving high-level tasks. Finally, in Section 5, an outline for future work is presented.

In the following section, I provide a brief overview of how robots make decisions. This is followed by a discussion of environment models in Section 2.3 and challenges in Section 3. A base background knowledge in decision making (Thrun et al., 2005; Sutton and Barto, 1998), robotic systems (Dudek and Jenkin, 2010), and machine learning (Bishop, 2006; Goodfellow et al., 2016) is assumed and can be found in the included citations.

I realize that the terms *model* and *environment* are overloaded in the context of this document. For my purposes, a *model* is a framework that an agent can use to predict how the the environment around it will change, taking into account the

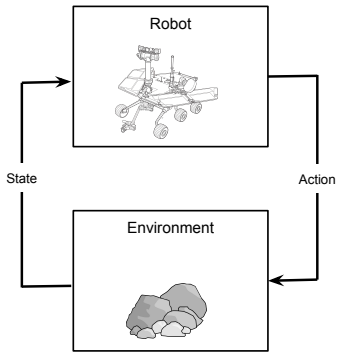


Figure 3: A Robot interacts with its environment through a series of state observations and actions.

agent's own actions. Although we seek to build robots which collect data in the natural environment (phytoplankton, wind, etc), for the purpose of this document, *environment* will refer to the world apart from the robot itself.

2 Robot Decision-Making

Robots interact with the world around them by making observations of the environment through sensors, and by taking control *actions* which change the current state of the environment. Robots must combine observations and measurements from multiple sources over time and space in a process called *state estimation*, where a robot's *state* is defined as the collection of all parameters about the robot and its environment that may influence its future, including the robot's *pose* (location and orientation). Importantly, an autonomous mobile robot must be able to interpret its own sensors to orient itself in the world and to make navigation or interaction decisions.

Most modern robotic agents operate in a sequential decision making environment in which they execute an action based on the best information available at the time to achieve some goal (see Figure 3). A Markov Decision Process (MDP) (Bellman, 1957) provides us with a convenient mathematical model for understanding decision making in environments in which the agent's actions only partly influence its future. An MDP is a discrete process (s, a, r, s') in which at each time step, t, an agent observes a state, s, chooses some action, a, and incurs a reward, r, thus progressing to the next state, s'.

Agents select actions according to their *policy*, π, defines a robots way of behaving at a particular point in time. Policies are, broadly, one of two categories. The first utilizes engineered reactions to sensory observations and does not change its policy based on experience. The second attempts to learn appropriate reactions based on experience. Furthermore, decision making approaches can be viewed as either reactive (model-free) or deliberative (model-based). Reactive agents choose actions only in response to observed states, whereas model-based or deliberative approaches reference some model of the world when selecting an action.

2.1 Reactive Methods

Reactive agents make decisions based on a state observation by referring to either a pre-programmed state-action sequence or by referring to a learned controller. Simple

engineered reactive agents may convert sensor observations or their location into motion vectors for avoiding obstacles or moving to a specified goal location (Khatib, 1985; Rivera et al., 1986).

Reactive agents that learn fall under a class of algorithms known as *model-free reinforcement learning*. In this paradigm, an agent will sample actions according to an internal policy, and then adjust the policy to optimize for reward (Bellman, 1957).

Reinforcement learning methods are theoretically advantageous because they make minimal assumptions about the world and are usually fast at decision time. Large advances in performance have been demonstrated by combining deep learning methods with the value-based Q-learning approach, such as human-level performance on a suite of Atari (Bellemare et al., 2012) games in (Mnih et al.). Similarly, policy-gradient based reinforcement learning approaches have had remarkable success in distributed learning settings and continuous control (Mnih et al., 2016).

2.2 Deliberative Methods

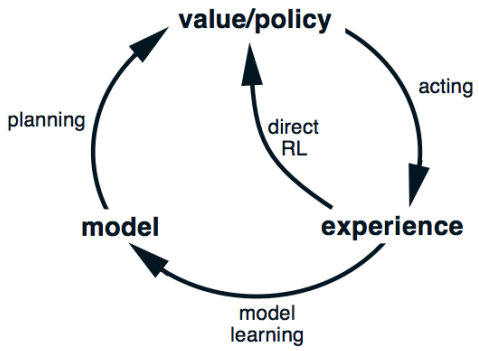


Figure 4: Relationship between learning, planning, decision making from (Sutton and Barto, 1998)

Deliberative agents perform forward search from the current state of the MDP to find actions (LaValle, 2006). This forward search is typically performed on-line, referencing the agent's current model of its future. This model may be the true future if available as in the case of many games. However, the model is often estimated when no exact future is available as in the case of most robotic applications.

Historically, robot motion planning has been dominated by graph based search algorithms. These approaches usually require a simplified *costmap* model of the robot's environment so that the robot can search for a path towards a goal. A costmap is constructed by quantizing the environment into cells, each classified by its content. Roadmap planning algorithms such as a *Visibility Graph* or *Voronoi* work by discretizing a given map into an undirectected graph where nodes are free space, and then searching that graph to find an optimal path.

In many real systems, a reliable model of the entire environment to discretize is often not available. Even if it is available, it may be inaccurate because of sensing errors or dynamic features. To address this, practitioners often utilize on-line algorithms which allow robots to update their plan as they improve their model of the world through observations. These online algorithms may be graph-based as in the case of D* (Stentz, 1997) or sampling based like Probabilistic Roadmap (PRM) (Kavraki et al., 1996) and Rapidly-exploring Random Trees (RRT) (Lavalle, 1998). One

important limitation of search methods is that they may be slow to run in real time in complex search spaces.

Monte-Carlo Tree Search (MCTS) (Browne and Powley, 2012) with Upper Confidence Bounds for Trees (UCT) (Kocsis and Szepesvári, 2006) is a popular ((Silver et al., 2016; Guo et al., 2014; Bellemare et al., 2012; Lipovetzky et al., 2015)) search method which incorporates Upper Confidence Bounds (UCB) from the bandit literature with search to handle exploration/exploitation tradeoff. Given an accurate representation of the future and sufficient time to compute, MCTS performs well (Pepels et al., 2014), even when faced with large state or action spaces. MCTS works by *rolling out* many sequences of actions on possible future scenarios to acquire an approximate (Monte Carlo) estimate of the value of taking a specific action from a particular state.

In planning, goals and hazards are typically specified ahead of deployment time, however one could see how it might be difficult (and expensive) to enumerate every possible failure case. We might want agents which learn, improving performance over time. We can utilize similar techniques from model-free reinforcement learning to learn both a model of the world and a policy which allows an agent to learn how to act. In model-based reinforcement learning, agents are generally able to improve sample efficiency over model-free methods by performing at least some of the agent's training within its learned model (Deisenroth and Rasmussen, 2011; Sutton, 1991). One of the earlier architectures which integrated learning, planning, and reacting into an agent is *Dyna*, which was introduced in 1991 (Sutton, 1991) (see Figure 4). In the Dyna architecture, the agent builds a one-step action model of its world and uses this *world model* to train a reactive controller. At decision time, it uses the reactive controller to make decisions, but incorporates the experience into its model.

While model-free decision-making methods can fail in ways that are hard to interpret, model-based approaches move part of the problem to an interpretable space where failures in modeling the environment itself can be analyzed more directly. However, modeling high-dimensional observational data with complex temporal dynamics is a challenging task, and incorrect models can result in significant errors.

2.3 Models

In my research, I have mostly utilized existing deliberative decision-making methods paired with strong models of the environment for model-based planning or reinforcement learning. As the model of the future improves (Oh et al., 2015), agent performance also improves.

A model of the environment is anything that an agent can use to predict how the environment will respond from a given state and action. We focus on models which predict the s', given the s, and a (minimizing Equation 1), though models may also predict reward or value of a state.

$$\sum_{t} ||f(s_t, a_t) - s'_t||^2 \tag{1}$$

If a robot is interacting in an environment in which dynamics are well known, we can give the agent access to known models for predicting the future. This is the case in many games (such as chess), where rules of the game form a model which can be used play out possible futures. We can also employ a similar technique when the laws of physics describe an agent's world. Physics models have been used with robots utilizing wind for transport (Douglas Luders et al., 2016) and trying to find

phenomena such as radiation sources (Rolf et al., 2018). In Section 4.2, I'll discuss our project which utilizes physics equations to predict robot sensor trajectories in a marine flowfield.

However, in many robotics applications, a full physics or rule-based model is not available. In this case, we can attempt to learn a model of the environment which can be utilized by the agent to imagine future scenarios for use in decision making. There has never been more data available from which to develop models which describe how the world works. Although electronic sensors have existed for decades, recent developments in lightweight computers, sensors, and batteries coupled with reliable wireless communication and localization schemes allow us to gather information from environments that were previously seen as too costly or risky to instrument. Consequently, we have an incredible amount of data which describes human (Lane et al., 2010), built (Zanella et al., 2014), and natural (Hart and Martinez, 2006; Villarini et al.) dynamics and distributions over space and time. In addition, artificial worlds developed from real observations of human environments (Xia et al., 2018), simulation environments based on the natural world (Manderson and Dudek, 2018), and computer games (Bellemare et al., 2012) allow artificial agents to gain unlimited, risk-free experience that may be transferable to real robots.

Agents which learn a model by interacting with the environment usually follow the following iterative steps to acquire and use a model for planning (Silver):

1. Run base policy to collect trajectory, $s, a, {s'}_t$
2. Learn dynamics model $f(s, a)$ to minimize model error (Equation 1)
3. Utilize deliberative or reactive policy to select action
4. Execute single action, observe s'
5. Add (s, a, s') to dynamics model dataset

The model may be learned independent of the agent as in (Ha and Schmidhuber, 2018) and (Hansen et al., 2018a) or the agent may have to take steps to properly explore then environment in order to build an environment model. We'll discuss both approaches Section 4.1.

2.3.1 Learned Models

Density estimation is core problem in machine learning and is a key component of learning models for decision-making agents. In the following few paragraphs, I introduce several models that I've used and that will be referenced later in the document.

The type of model chosen to learn a model varies based on the task. I have focused much of my work on learning latent-variable generative models of environments. A key insight of latent variable models is that there is some underlying representation which can explain features or dynamics in a larger observed state. Latent environment models can be learned in an unsupervised manner to provide agents with a so-called *imagination* that can be queried when preparing to act in the true environment without predicting full future observations.

In autoencoders, a bottleneck in the neural network forces the system to learn a concise representation from which to reconstruct the input. Ideally, the latent variables produced in the bottleneck will represent a compressed spatial and temporal representation of the environment which may be useful for our agents.

The Variational Autoencoder (VAE) (Kingma and Welling, 2013; Rezende et al., 2014) is an interpretation of an autoencoder as a graphical model where the archi-

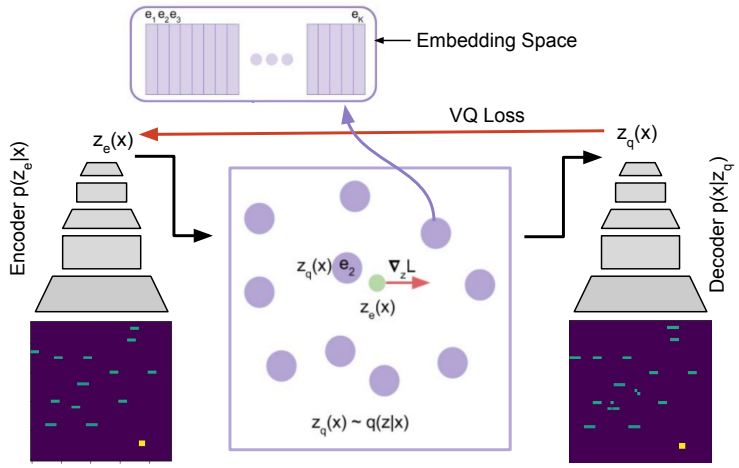


Figure 5: VQ-VAE optimizes a discrete embedding space by moving embedding centers, e_i with respect to the output from the encoder, $z_e(x)$. This figure is a modified version of Figure 1 in van den Oord et al. (2017b).

tecture introduces the concept of a prior. The network trains by gradient descent by utilizing the *reparameterization trick* to maximize the lower bound on the log-likelihood from samples from the posterior. VAEs are popular in part because the ease of working with the Gaussian prior has enabled a number of follow-up applications. One common failure of VAEs is *mode collapse* which occurs when the latent representation fails to capture useful information in latent space. This is especially a concern when VAEs are paired with expressive autoregressive decoders such as (van den Oord et al., 2016b). There have been several research threads which attempt to solve mode collapse including (Alemi et al., 2017; Gulrajani et al., 2016; van den Oord et al., 2017b; Graves et al., 2018) and I'll detail of these models that I have used in the following paragraphs.

The Vector Quantised Variational Autoencoder (VQ-VAE) model is one of the extensions to the VAE which helps avoid mode collapse. VQ-VAE transforms input into a discrete latent representation utilizing Vector Quantization (VQ) (Gray, 1984), rather than assuming an explicit density as in VAEs. The VQ-VAE employs a three-part loss described in Equation 3. The first component of the loss is the reconstruction loss, which utilizes a straight-through gradient estimate. The second and third terms in the loss function seeks to learn better positions of the embedding vectors e_i with respect to the encoder output, $z_e(x)$ (as depicted in Figure 5). The VQ-VAE does not have an explicit prior which can be sampled, however a prior may be learned after training. This model architecture has proven data-efficient, robust to mode-collapse, and useful for learning models for decision-making agents. The discrete latent space is nice because it allows us to utilize search-based agents with rollouts of the future as in (Hansen et al., 2018a).

Associative Compression Networks (Graves et al., 2018) are a form of variational autoencoders in which the prior distribution used to model each code in the latent space is conditioned on a similar code from the dataset, forming a non-parametric prior. This coding scheme allows the prior to account for local variations in the latent space in a flexible way, which helps create information-dense codes and prevents mode-collapse while preserving neighborhoods. The loss is described by Equation 4.

$$L^{VAE}(x) = KL(q(z|x)||p(z) - \mathbb{E}[\log r(x|z)] \tag{2}$$

$$L^{VQVAE}(x) = \log p(x|z_q(x)) + ||sg[z_e(x)] - e||_2^2 + \beta||z_e(x) - sg[e]||_2^2 \tag{3}$$

$$L^{ACN}(x) = \mathbb{E}[KL(q(z|x)||p(z|\hat{c})] - \mathbb{E}[\log r(x|z)] \tag{4}$$

Autoregressive models like (Larochelle and Murray, 2011; Germain et al., 2015; van den Oord et al., 2016) and (van den Oord et al., 2016b) improved the state of the art having have proven that they are capable of estimating high-dimensional data such as raw images (van den Oord et al., 2016; van den Oord et al., 2016b; Salimans et al., 2017), audio (van den Oord et al., 2016a, 2017a), and video (Kalchbrenner et al., 2016). The autoregressive technique utilizes the chain rule to reinterpret the joint distribution as an exact product of conditional distributions (see Equation 5). Combined with clever masking, these conditional distributions can be found efficiently at train time, though the chain must be sampled sequentially at evaluation, resulting in slow generation. One key drawback, is that there is no latent variable in these models, however, they can be combined with encoders which produce latent variables such as VQ-VAE and ACN.

$$\begin{aligned} p(x) &= p(x_0, x_1, x_2, ...x_n) \\ p(x) &= p(x_0) \cdot (x_0|x_1) \cdot (x_2|x_0, x_1) \cdot ...p(x_n|x_{n-1}, ..., x_2, x_1, x_0) \\ p(x) &= \prod_{i=1}^{n^2} p(x_i|x_1, x_{i-1}) \end{aligned} \tag{5}$$

A recurrent neural network (RNN) (Elman, 1990) is a model that is particularly useful for modeling sequential data. The network has recurrent connections which allow it to retain a "memory" across multiple inputs. This style of recurrent model is also known as a *state space model* or *dynamical system*. Long short-term memory (Hochreiter and Schmidhuber, 1997) are RNN-style models which improve gradient propagation by adding additional *gates* to the RNN structure which effectively allow the network to manage information flow.

The non-parametric Gaussian Process (GP) (Rasmussen, 2006) is often employed for modeling spatiotemporal processes (Singh et al., 2010; Zhao et al., 2016; Kim et al., 2011; Reece et al., 2011) and in our own work in (Hansen and Dudek, 2018; Hansen et al., 2018b). GPs represent training data as a distribution over a family of functions. At evaluation time, the GP measures the similarity between all data points using a kernel function. Gaussian Processes are data efficient and provide a measure of uncertainty, but the standard implementation has difficulty handling large datasets ($n > 10^4$), as it requires $O(n^3)$ computations and $O(n^2)$ storage for n training points (Rasmussen, 2006), though there has been work to scale the technique exactly to larger datasets ($n = 10e6$) (Wang et al., 2019). Recent extensions combine GPs with neural methods (Vinyals et al., 2016; Garnelo et al., 2018), attempting to combine the positives of GPs such as rapid adaptation to new data and uncertainty with the computational efficiency at evaluation of neural networks.

Modeling is a particularly active area of research and thus there are many new model architectures worth considering in model-based decision making. Some which may be particularly applicable include (Li et al., 2018) which enables high-performance super resolution as well as transformations (night to day) which may enable agent environment transfer. Transformer models (Vaswani et al., 2017) which utilize attention mechanisms rather than RNN-style recurrence have shown good performance in sequential data. My workflow is well-situated to adapt to new models as performance improves.

3 Challenges

This section provides a general background on a subset of the challenges faced in developing decision making agents.

3.1 Imperfect Observations

Real-world systems are plagued by imperfect or underactuated mechanics and perception systems that provide limited insight into their environment. Determining a robot's location in their environment and relationship to other objects is often a difficult problem. The absolute position of robots operating outdoors can be found with sensors which interpret messages from known remote locations such as Global Positioning System (GPS) receivers or Ultra-Short Baseline (USBL) transceivers. Robots may find their relative position by systematically keeping track of their own movement using a combination of observational sensors such as cameras, inertial measurement units (IMUs), or tachometers and algorithms like dead-reckoning (Dudek and Jenkin, 2010), simultaneous localization and mapping (SLAM) (Davison et al., 2007), visual odometry (VO) (Nister et al., 2004).

In addition to localization and observation sensors, scientific mapping robots are equipped with a combination of sensors that allow them to observe phenomena in the environment, such as fluorometers, sonars, cameras, or thermometers. Robots may need to interact with their surroundings by collecting samples, probing the earth, or deploying additional independent measurement devices. A major challenge to measurement is the presence of noise or error in sensor observations.

These systems must make decisions in a limited decision time due to timing constraints in physical systems, usually with only the computation that they carry and power with batteries.

The paradigm introduced in the MDP of "the future is independent of the past given the present" does not hold in the presence in real-life sensor limitations. This means robots often operates in a Partially Observed MDP (POMDP) as the full state is not observed.

3.2 Sample Efficiency

Despite high-profile success, the use of model-free approaches for physical robots has been relatively limited due to their sample inefficiency. Training robots in their target environments requires physical interactions that are often slow, resource intensive, and often dangerous for the robots as well as other inhabitants of the environment.

Two popular approaches to reducing the number of samples needed to train an agent are *transfer learning* and *imitation learning*. Typically, in *transfer learning* regimes, the agent learns how to interact well in a world in which it is safe and easy to gather experience (usually a simulator), and then finds a way to leverage this knowledge in a target environment where it is usually much more expensive to gather experience (Xia et al., 2018) (Taylor and Stone, 2009). In *imitation learning* frameworks (Schaal, 1999; Abbeel and Ng, 2005; Ratliff et al., 2009; Ross et al., 2011), the agent leverages a few examples from an expert to begin learning from a fairly good policy.

3.3 Exploration

A key problem in decision-making is that it can be difficult to observe the entire state space needed to make good decisions. An agent needs to exploit what it already knows in order to succeed in maximizing its objective, but must explore to discover actions which may lead to an even greater reward. Historically, many successful reinforcement learning approaches have used an ϵ-greedy approach to exploration, taking random actions some percentage of the time during training rather than following the current policy. Though unbiased, this approach does not induce temporally-extended exploration and is data inefficient.

Recent work has worked to incentivize agent exploration defining intrinsic rewards with *pseudo-counts* that rewards state-action pairs which have been experienced infrequently (Bellemare et al., 2016). Osband et. al utilizes bootstrapped agents to achieve *deep exploration* in (Osband et al., 2016). This work is extended in (Osband et al., 2018) to include random prior functions for each bootstrapped agent, achieving high uncertainty in unfamiliar states which can be used to drive exploration.

4 Research Direction

My research efforts seek to improve robot decision making with the ultimate goal of improving autonomy in scientific sampling tasks. This work builds on techniques for model-based decision making using modern approaches for learning environment models. In Section 4.1, I present related and motivating research as well as my own work on building models for agents in simulated environments. In Section 4.2, I introduce a scientific sampling problem that is a major motivating problem for my thesis. This project relies on learning an environment model to facilitate sensor transport for low-cost persistent sampling in marine environments.

4.1 Model-Based Decision Making Agents

In this line of work, we consider learning unsupervised models of the future in high-dimensional dynamic environments for use by a decision-making agent.

4.1.1 Related and Motivating Work

Much of the progress in decision-making agents in recent years has come from model-free reinforcement learning approaches, which have learned effective policies in complex tasks, even in cases where the observational space is large. This progress has largely been attributed to employing deep neural network architectures similar to those used in machine vision tasks such as classification (Krizhevsky et al., 2017), detection (Ren et al., 2015), segmentation (Ronneberger et al., 2015). However, modern model-free methods require many interactions between the agent and its environment, making this approach infeasible for many robotics applications. Model-based approaches, hold the promise of improving efficiency of learning agents, as they may learn actions in a simulated model of their environment. However, until recently, most model-based approaches have been held back by poorly-modeled environments. As learned environment models improve, the p0erformance of model-based decision making agents will also progress.

Planning is a powerful approach to sequential decision making problems where the environment dynamics are known. The success of the model-based AlphaGo (Silver et al., 2016) and AlphaGoZero (Silver et al., 2017) in the large state-space, sparse-reward game of Go has inspired a bevy of work (H. S. Segler et al., 2017; Anthony et al., 2017; Guez et al., 2018; **?**) (including our own (Hansen et al., 2018a)) in

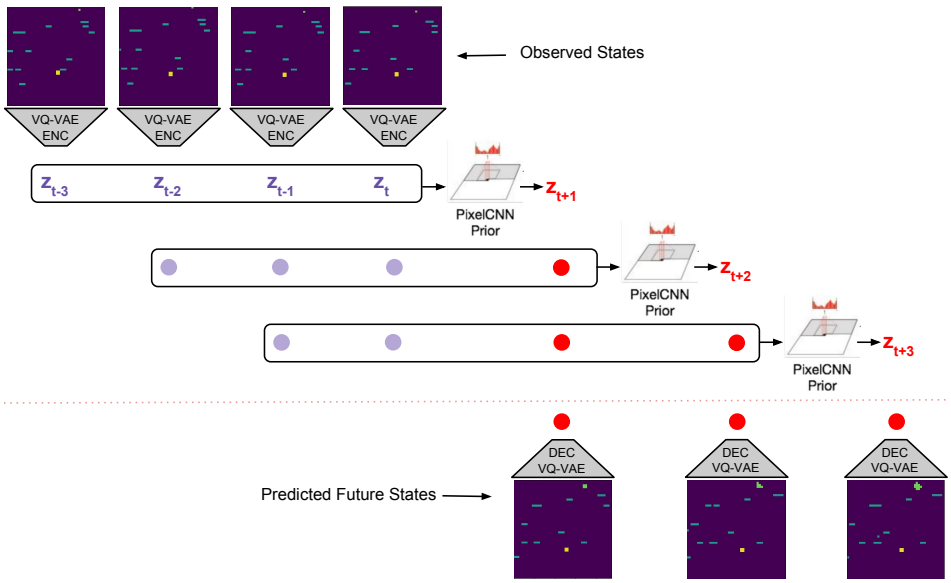


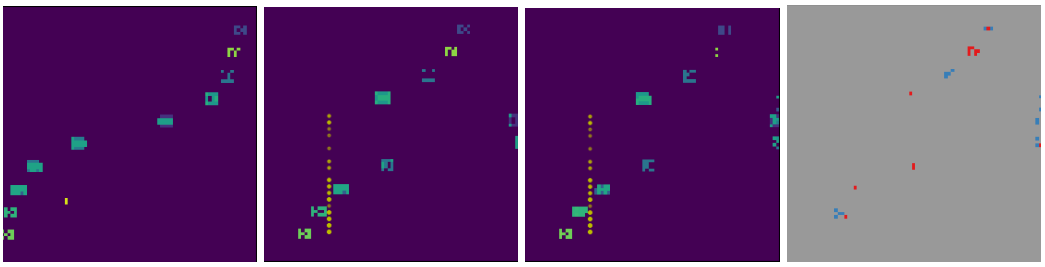
Figure 6: VQ-VAE+PCNN model diagram shown predicting 3 states into the future (shown in red), conditioned on 4 past z states (purple) on Freeway.

model-based decision making where neural networks are used in coordination with traditional search for decision making. In AlphaZero, the rules of the game provide a perfect model by which to train a two-headed network which takes the current game state as input and chooses actions based on an approximation of Monte-Carlo Tree Search.

The 2014 paper (Guo et al., 2014) showed offline MCTS over RAM representations of game state in Atari (Bellemare et al., 2012) out-performs learned policies without access to the true future. However, outside of games, most environment dynamics are not given. Recent work has paired learned models of the environment with planning over a learned latent space (**?**) or learning agents (Kaiser et al., 2019). When making decisions in unknown environments, the agent needs to experience the environment in order to build a model. In (Kaiser et al., 2019), a a learning agent with a learned model of observed Atari frames, achieves human-level performance in with <100K interactions with the environment (<2 hours of game play). This result is in stark contrast to the many millions of steps needed by the best model-free approaches to achieve similar performance.

Key problems on planning on learned environments are those related to model inaccuracy. These include accumulating errors as the imagined rollout is farther from the observed state. May fail to capture many possible futures and an agent can learn confident policies in regions which are actually outside the training distribution of the model.

Much work has shown that learning a dynamics model from a compressed latent space can improve efficiency in reinforcement learning agents (Oh et al., 2015; Ha and Schmidhuber, 2018; Finn et al., 2016; Buesing et al., 2018). In (Finn and Levine, 2017), the authors develop a method for combining deep action-conditioned video prediction models with model-predictive control that uses entirely unlabeled training data to enable a real robot to perform manipulation. In (Ha and Schmidhuber, 2018), learn a lightweight controller which interprets the latent representation of the current state from a VAE, z_t and the action-conditioned estimate of the future, z_{t+1} from an RNN. The models int this case are trained from random rollouts of the agent. In



(a) Observed State t_{14} (b) True t_{14+13} (c) Estimate t_{14+13} (d) Error t_{14+13}

Figure 7: This panel depicts an example of our model rolled out 13 time steps in the future in the Atari Freeway game (Bellemare et al., 2012). The yellow pixels depict the agent position. This yellow pixel is the true position in Figure 7a and is the imagined states in Figures 7b and 7c. Error in our imagined state from Figure 7c is shown in Figure 7d. This error is represented as false negatives in red and false positives in blue.

I2A, (Racanière et al., 2017), the agent's policy is informed by the outputs of both a model-free value estimate and Monte-Carlo estimate from a rollout on a learned model. This dual approach allows the agent learn to interpret its imperfect internal model to construct implicit plans in new ways.

4.1.2 Progress to Date

In preliminary results (Hansen et al., 2018a), we demonstrate the utility of using a two-stage pipeline to train a forward model over latent representations of an action-independent environment. We utilize a VQ-VAE (van den Oord et al., 2017b) to learn discrete representations and pair this with a a conditional gated PixelCNN (van den Oord et al., 2016b) to predict one-step ahead z representations of sequential frames when conditioned on previous z representations. Our VQ-VAE configuration compresses an input size from our state observation of $48 \times 48 \times 1$ down to a Z space of $6 \times 6 \times 1$. This concise representation reduces the time needed for sequential generation by the autoregressive model, bringing our agent closer to real-time decision making.

To introduce Markovian conditions, our conditional PixelCNN is fed a spatial conditioning map of 4 past z encodings causing it to learn a model corresponding to $p(z_{t_{i,j}}|z_{t_{<i,<j}}, z_{t-1}, z_{t-2}, z_{t-3}, z_{t-4})$. Each dimension (i, j) of Z_t is conditioned on all valid dimensions relative to the current position via autoregressive masking and also on the previous 4 frames which are fed to the model by a spatial conditioning map. Combined with the previously trained VQ-VAE decoder this results in a model which generates 1 frame ahead, given 4 previous frames. It is possible to generate an arbitrary number of frames forward given an initial 4 frames, by chaining 1 step generations though we expect results to degrade as forward trajectory lengths increase. We illustrate this structure in Figure 6. We tested this forward-model with an agent utilizing MCTS for action selection and demonstrated that it performed well compared to MCTS agents which had access to the true fully-observed future in a dynamic environment.

A key component which made the previous approach computationally feasible is that the environment we tested on was *not* action conditional, meaning dynamics in the world continued regardless of what actions are chosen by the agent in its rollouts. This means that generated future frames were be shared across all rollouts in MCTS, greatly reducing the overall cost of running the autoregressive model.

Combined with the speed improvements from generating in a compressed space given by VQ-VAE, forward generation was accomplished in reasonable time. This action-independence is the case in many scientific sampling environments which we care about. For instance an autonomous underwater vehicle will not have much effect on the ocean current or coral reef distributions and can use future predictions about the environment to make reasonable sampling decisions.

4.1.3 Future Work

Current and future work is centered around extending initial results to action-conditional environments. Complex interactive environments require exploration to fully model the state-space. As discussed previously, our current approach is likely too slow to realistically be realized in interactive environments. In addition, we need a model which can also estimate rewards and end-of-life scenarios. The bullet points below enumerate the tasks required for building a full action-conditional learning agent.

- I have incorporated ideas from (Osband et al., 2018) for building models in environments which require exploration to observe the state space. Exploration didn't really effect our agent in the simple environment shown in Figure 6 as the state space was fully observed and we used simple rules to indicate reward to the planning agent. An agent trained by reinforcement learning will need to experience different factors in the environment in order for its model to depict them accurately.
- Although we saw very good performance from VQ-VAE reconstructions, the latent space in this model does not have implicit ordering, effectively serving as a learned lookup-table in latent space. Knowing neighbors in the latent space such as in VAEs might be useful for imagining future scenarios. My ongoing work seeks to incorporate ideas from ACN into a VQ-VAE style model for encouraging latents to have meaningful neighbors.
- Ultimately, I would like to examine how well learned environment models can be transferred to related but different environments. Future work will involve transferring to robotic tasks, such as navigation in different buildings in (Xia et al., 2018).

Ideas and methodologies from this section are extended to a scientific sampling environments in the next section.

4.2 Flowfield Modeling for Low-Cost Persistent Autonomous Sampling

A significant motivating problem in my work has been that of efficiently collecting samples necessary to gain understanding of spatiotemporal data phenomena in bodies of water, such as pollutants, coral, or algae. The process of gathering in-situ data from a spatiotemporal marine field is an expensive undertaking for the scientists or regional managers trying to understand near-shore activity.

Without monetary constraints, most scientists would opt to deploy a dense array of powered static sensors throughout their survey space such as in (Brainard et al., 2009). This deployment method allows for high spatial and time resolution and is capbable of producing powerful datasets which can be used for understanding the world. However, most of the time, this instrumentation is too costly to deploy over large regions with high spatial resolution. In addition, because these array are not easily transporable, they cannot be resued at new locations.

Other surveying techniques employ sophisticated robots, such as aerial drones or autonomous surface vehicles (ASVs). These robotic surveying systems usually employ an exhaustive sampling strategy. The exhaustive approach, though guaranteed to produce a correct model of the data given Nyquist sampling over an apparently static environment and calibrated sensors, can be tedious if the survey space is large and the data is predictably distributed. Efficiency can often be achieved with the same class of observational sensors by either better anticipating where important information is located and adaptively sampling the spatial field (Bourgault et al., 2002; Yog, 2016; Das et al., 2015; Low et al., 2008; Rahimi et al., 2005; Singh et al., 2007; Fiorelli et al., 2006; Chadwick et al., 2016) and/or by reducing the cost of traveling to each sample point (gli, 2004; Busquets et al., 2012).

4.2.1 Heterogeneous Sensing Teams

My approach to low-cost autonomous surveying system utilizes learned and known models to enable long-term (multi-day) surveys with a heterogeneous robot team in marine regions. In addition to faster coverage with multiple agents, we know that robot teams are often more resilient than single robot systems, since task completion can still be achieved despite individual failures. Moreover, when teams are appropriately heterogeneous, task distribution can be utilized to allocate individual classes of robots for specific tasks (Dudek et al., 1996).

Our sampling team consists of an autonomous surface vehicle (ASV) which is able to navigate around a survey region and may deploy a limited number, n of passive floating sensors called *drifters*. Drifters are low-cost, passive sensors require very little battery power to collect samples because their movement upon deployment is governed by the ambient flowfield. The major focus of this project is built upon the premise of accurately modeling the flowfield in order to exploit it for sensor transport. Others have utilized flowfields for tracking features of interest (Kularatne and Hsieh, 2015) and for generating informed paths (Kularatne et al., 2018; Inanc et al., 2005; Kwok and Martínez, 2010), however, we believe we are the first to consider autonomous deployment and cooperative sampling with drifters. Similar

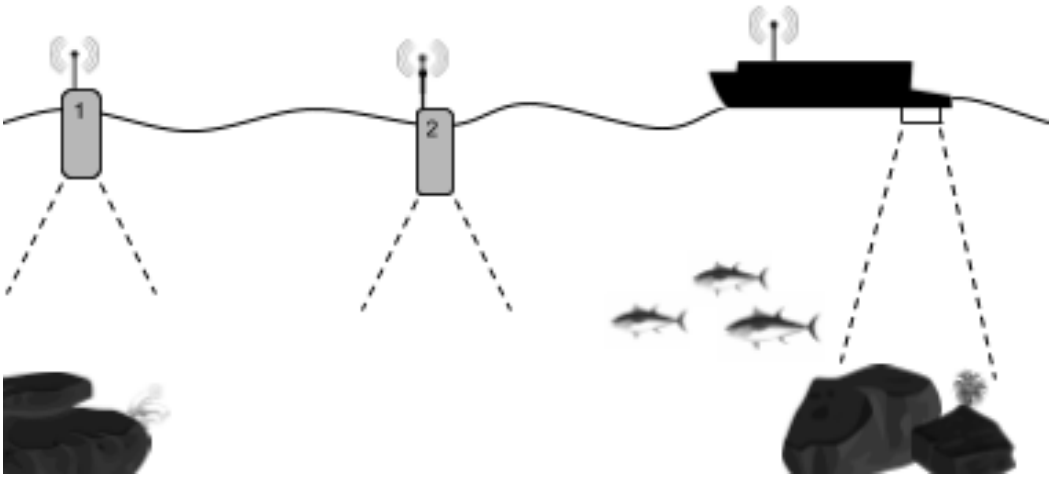


Figure 8: Our heterogeneous marine sensing team described in Section 4.2. An autonomous boat deploys a team of inexpensive passive drifters equipped with observational sensors which collect georeferenced samples of environmental phenomena, including the local flowfield.

techniques have been applied to mediums with observable dynamics that can be used to transport an agent at low cost (such as animal tracking (Hart and Martinez, 2006) or high-altitude parafoils for air-quality monitoring (Douglas Luders et al., 2016)).

Drifters can be equipped with a variety of sensors for collecting georeferenced data and have been used extensively in oceanography (Wilson et al., 1996; Soreide et al., 2001; Lumpkin et al., 2007; Meghjani et al., 2016; Shkurti et al., 2012; Alam et al., 2018; Aoyagi et al., 2004). Priced at under $\$150$CAD, these devices fill an important niche in ocean observations with their long battery life and low-cost, however, because these devices are not controllable once deployed, their deployment location must be considered carefully in order to capture data from informative regions of the survey area.

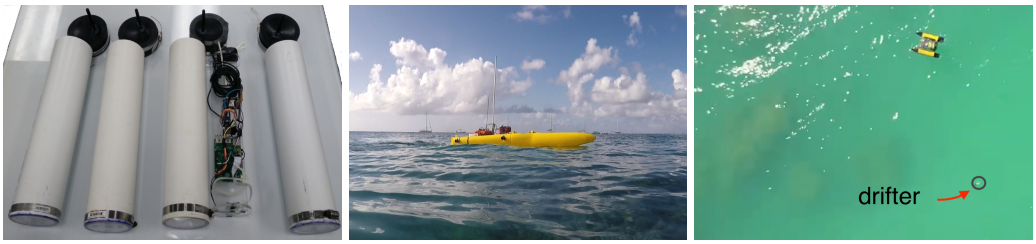
4.2.2 Challenges

There are many practical and learning problems involving the deployment of heterogeneous sensor systems. Within all distributed robot tasks there are important considerations of power management, communication disruptions, and errors in sensing and localization. For a given task it may be important to optimize for different parameters, such as performance, speed, cost, or safety of the system. In this work, we primarily focus on problems related to environment modeling and action selection and largely assume low sensor noise, good wireless infrastructure and a predictable battery-recharge cycle.

$$\vec{V} = u(x, y, z, t)\vec{i} + v(x, y, z, t)\vec{j} + w(x, y, z, t)\vec{k}$$

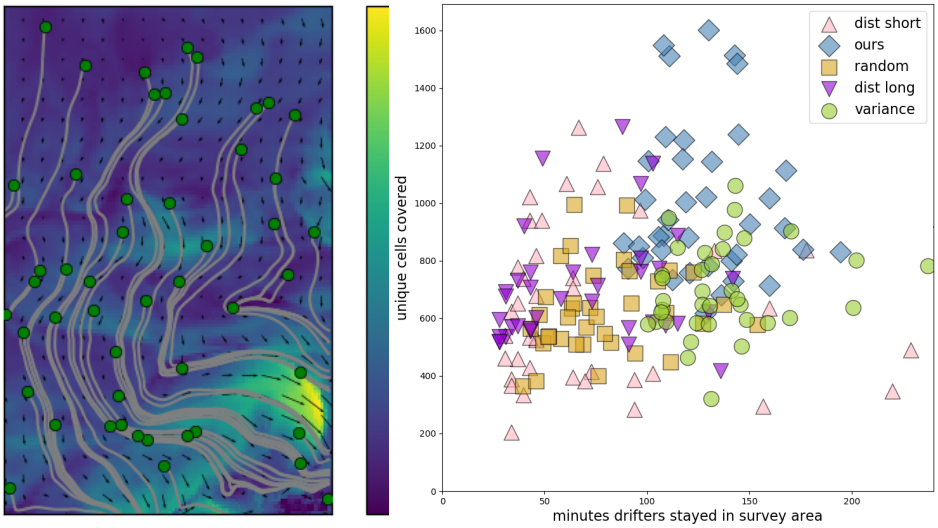
Strategic deployment is necessary to achieve any form of efficient coverage with drifters. This was clear in our early work, (Quattrini Li et al., 2016; Manjanna et al., 2016), in which we utilize data from randomly deployed drifters for informing adaptive sampling schemes in controllable vehicles. In these early experiments, the drifters tended to clump together or rapidly exit the survey area, reducing their utility. However, previous research (Alam et al., 2018; Lumpkin and Elipot, 2010; lum, 2005; Salman et al., 2008; Slivinski) has shown that, given a known flowfield, (Equation 4.2.2), an initial deployment location of an object, and a perfect physical description, a trajectory of the object through a flowfield, $\vec{V}$ (described by Equation 4.2.2), can be calculated using the advection equation as a function of time.

However, in the scale of survey regions that we consider ($< 20km^2$ with sample resolution of $< 10m$), the flowfield for most bodies of water is largely unknown before the survey begins and is only precisely observable in-situ. We utilize the real-time flow observations from our deployed sensors to estimate the true flowfield, however, this configuration is the classic exploration vs. exploitation scenario . We must explore in order to determine the flowfield which will dictate a drifter's trajectory that in turn dictates information gain.



(a) ocean drifter (b) autonomous surface vehicle (c) overhead of deployment

Figure 9: Hardware used in drifter field experiments.



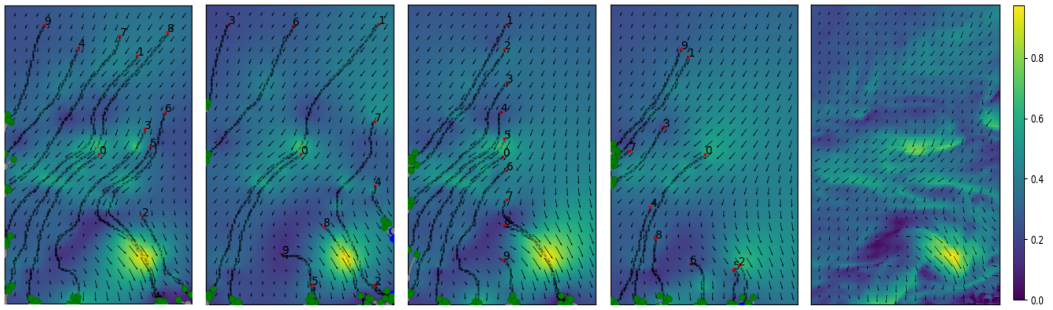
(a) Drifters (b) Baseline Performance

Figure 10: Figure 10a demonstrates the use of a physics simulator, OpenDrift (Dagestad et al., 2017), to determine drifter trajectories in a flowfield that was used in our experiments. The drifters depicted in this figure were randomly released at the green points and traveled along the gray tracks over the period of 5 simulated hours. The flowfield direction is described in each grid cell by an arrow and current speed is represented by the colormap in $\mathrm{m/s}$ according to the color bar. Figure 10b shows a performance comparison of several baselines over 40 flowfield maps from (Hansen and Dudek, 2018). Good surveys contain a large number of unique points and have drifters which stay in the survey region for as long as possible (i.e. the top right corner is best). Our deployment scheme performed best in 37 of the 40 flowfields we examined.

4.2.3 Progress to Date

As part of a larger collaboration, I have assisted in building software and hardware for the drifters and ASV seen in Figure 9. Though we have conducted several experiments in open bodies of water, most of my work focuses on simulated experiments on archival data. I built a comprehensive environment simulator for deploying drifters under a variety of flowfield and environmental models utilizing Regional Ocean Modeling Systems (ROMS) (Shchepetkin and McWilliams, 2005) geophysical data. This simulated world allow for quantitative comparisons against known baselines.

- In our first experiments with incorporating drifters into the survey task, we randomly deployed drifters into a survey region (Quattrini Li et al., 2016). The drifters sampled a region in tandem with several autonomous sampling vehicles, leading to increased coverage of the region. In (Manjanna et al., 2016), we also deployed drifters randomly over a survey area, but this time, we used the data relayed from these sensors to inform an ASV for efficient adaptive sampling (Manjanna et al., 2016).
- In (Hansen and Dudek, 2018), we introduced an approach for finding drifter deployment points which optimize for survey coverage, using the ASV only for drifter deployment (Figure 11). In this paradigm, deployed drifters collect observations of the flowfield use this data to model the flowfield using a Gaussian Process (GP). Use the uncertainty output of the Gaussian Process to sample a computationally feasible number of potential deployment points



(a) ours (b) variance (c) distributed long (d) random (e) ground truth (f)

Figure 11: This panel shows the deployments (numbered red points) chosen for 10 drifters under 4 different deployment schemes. Our approach in (Hansen and Dudek, 2018), which optimized drifter deployment for coverage, is compared to 4 baselines (only 3 best performing are shown). The true flowfield for these experiments is depicted in Figure 11e. The background ($\mathrm{m/s}$ refers in colorbar in Figure 11f) and arrows in the experimental figures depict each deployment scheme's estimate of the flowfield at the terminal state of the survey.

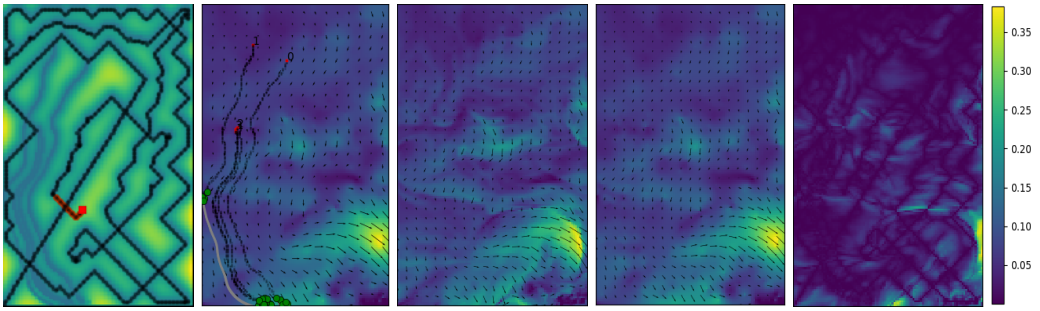
and then find the trajectories from these points using the flowfield estimate and a particle simulator. Score the hypothetical trajectories based on expected information gain and deploy a drifter to the highest value deployment point. Repeat until all drifters are deployed.

- In (Hansen et al., 2018b), we introduce a comprehensive technique which considers a surveying task in which both the ASV and the drifters are capable of observing the flowfield and phenomena of interest. We perform task distribution between drifters and the ASV, optimizing for model error in the flowfield of interest. The resulting policy results in the ASV deploying drifters to points which result in long, informative trajectories as seen in Figure 12. The ASV samples around the drifters and their predicted trajectories, resulting in efficient surveys.

4.2.4 Future Work

I aim to tackle recovery and redeployment of drifters for persistent sampling in marine flowfields. In previous work, we have limited our surveys to regions in which the flowfield remains constant over time. Future work will consider temporal changes in the flowfield.

The physics simulator used to estimate drifter trajectories in (Hansen and Dudek, 2018) and (Hansen et al., 2018b) is relatively slow at the scale at which we are estimating. In future work, I will investigate the incorporation of neural models such as those presented in (Tompson et al., 2016) for accelerating eurlerian calculations. My previous flowfield modeling efforts have started training a model from scratch at every survey. However, there are many attributes of flowfields that may be informed from previously observed data. I'll look at approaches for transfering understanding to new environments and for using available priors such as bathymetry maps and satellite imagery. Rather than utilizing a rules-based approach as in previous work for ASV actions, I will learn an policy for drifter deployment and collection.



(a) ASV Path (b) Drifter Path (c) True Flowfield (d) Estimate (e) Final RMSE (f)

Figure 12: Figure 12a shows the full boat path (black points) at the end of an experiment in which our algorithm distributes observation tasks between an ASV and several drifters as described in (Hansen et al., 2018b). In this figure, the last position of the ASV is shown in red and all sampled positions (drifters and ASV) are shown in violet. The background is the normalized reward map used for driving ASV sampling and follows the colorbar in Figure 10a. Figure 12c is the true flowfield and Figure 12d is what the experiment estimated the flowfield to be at the end of the experiment. The background for Figures 12b,12c and 12d refer to speed in $\mathrm{m/s}$ in the colorbar seen in Figure 10a. In Figure 12d, we also depict actual trajectory of the drifters in black as well as their starting positions in red. Estimates of the future trajectories are included in gray points with a green marker indicating the final position, however, at this point in time all of the drifters have actually exited the survey area. The rightmost image, Figure 12e, shows the Root Mean Square Error (RMSE) in $\mathrm{m/s}$ of Figure 12d with respect to the ground truth. The colorbar in Figure 12f provides the metric for Figure 12e.

5 Timeline

In the remaining time during my PhD, I aim to accomplish the tasks presented below in Table 1 and discussed in Sections 4.2.4 and 4.1.3.

Status	Months	Project	Problem Description
C	2	D	Random drifter distribution for building prior for ASV using real data collected from Bellairs Reef at the McGill Field Station in Barbados.
C	6	D	Cooperative development of hardware/software platforms for ASV and drifters.
C	4	D	Build infrastructure pipeline for testing drifter deployment in simulation using (Shchepetkin and McWilliams, 2005) for data and the particle physics simulator, (Dagestad et al., 2017).
C	3	D	Develop deployment strategy for coverage with drifters in unknown flow field.
C	2	D	Explore/exploit with ASV and drifter sensors in unknown flow field.
C	5	D	Learn fully observable environment dynamics and combine with search.
W	4	G	Incorporate exploration to learn dynamics model in environments which are not fully observable through random actions.
W	5	G	Sample-efficient model-based RL agent on popular RL baselines.
W	8	G	Sample-efficient transfer of model-based RL agent to unseen, but related environments such as those in (Plappert et al., 2018), (Cobbe et al., 2018), or (Xia et al., 2018).
F	4	S	Mars sample discovery and retrieval for the 2020 Sample Return Project at Jet Propulsion Lab (JPL) summer internship.
F	3	D	Learn transferable environment models and policy on flow fields with passive sensors in temporally changing flow field.
F	3	D	Recovery and redeployment of drifter for long-term (multiple week) studies in temporally changing flow field in simulation.
S	6	D	Extend to simulated multi-agent, active-drifter setting.
S	4	S	Extend to additional domains like hurricane (JPL, 2016) or air pollution (NOAA), or various earth-science datasets (Google).
S	6	D	Develop strategy for drifter recovery using vision on real ASV and demonstrate in multi-day field deployment (requires significant hardware and field time).

Table 1: This table describes the tasks that I believe are necessary for completing my PhD in model-based surveying for scientific sampling robots. In the **Status** column, "C" indicates that a task has been completed, "W" indicates that I am currently working on this task, "F" indicates that this is future work, and "S" indicates a stretch goal. Integers in the **Months** column depict my estimate for the number of months necessary to achieve the task (note that tasks may overlap). A "D" in the **Project** column means that the task is intended to be applied to the "Drifters" project. An "S" in the **Projects** columns indicates that this is a broader scientific-sampling focused problem, and "G" suggests that the research is oriented towards general decision-making agents.

References

Underwater gliders for ocean research. *Marine Technology Society Journal*, 38(2), 2004.

Near-surface circulation in the tropical atlantic ocean. *Deep Sea Research Part I: Oceanographic Research Papers*, 52(3):495 – 518, 2005. ISSN 0967-0637.

Modeling curiosity in a mobile robot for long-term autonomous exploration and monitoring. 40:1267–1278, 2016.

P. Abbeel and A. Y. Ng. Exploration and apprenticeship learning in reinforcement learning. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 1–8, New York, NY, USA, 2005. ACM. ISBN 1-59593-180-5. doi: 10.1145/1102351.1102352. URL `http://doi.acm.org/10.1145/1102351.1102352`.

T. Alam, G. M. Reis, L. Bobadilla, and R. N. Smith. A data-driven deployment approach for persistent monitoring in aquatic environments. In *IEEE International Conference on Robotic Computing (IRC)*, pages 147–154, Jan 2018. doi: 10.1109/IRC.2018.00030.

A. A. Alemi, B. Poole, I. Fischer, J. V. Dillon, R. A. Saurous, and K. Murphy. An information-theoretic analysis of deep latent-variable models. *CoRR*, abs/1711.00464, 2017. URL `http://arxiv.org/abs/1711.00464`.

T. Anthony, Z. Tian, and D. Barber. Thinking fast and slow with deep learning and tree search. *CoRR*, abs/1705.08439, 2017.

H. Aoyagi, Y. Michida, M. Inada, H. Otobe, and R. Takimoto. Experiment of particle dispersion on the sea surface with gps tracked drifters. In *OCEANS '04. MTTS/IEEE TECHNO-OCEAN '04*, volume 1, pages 139–145 Vol.1, Nov 2004. doi: 10.1109/OCEANS.2004.1402908.

M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The arcade learning environment: An evaluation platform for general agents. *CoRR*, abs/1207.4708, 2012.

M. G. Bellemare, S. Srinivasan, G. Ostrovski, T. Schaul, D. Saxton, and R. Munos. Unifying count-based exploration and intrinsic motivation. *CoRR*, abs/1606.01868, 2016. URL `http://arxiv.org/abs/1606.01868`.

R. Bellman. Dynamic programming princeton university press. *Princeton, NJ*, 1957.

C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738.

F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte. Information based adaptive robotic exploration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 540–545 vol.1, Sept 2002. doi: 10.1109/IRDS.2002.1041446.

R. Brainard, R. Moffitt, M. Timmers, G. Paulay, L. Plaisance, N. Knowlton, J. Caley, F. Fohrer, A. Charette, C. Meyer, et al. Autonomous reef monitoring structures (arms): A tool for monitoring indices of biodiversity in the pacific islands. In *11th Pacific Science Inter-Congress, Papeete, Tahiti*, 2009.

C. Browne and E. Powley. A survey of monte carlo tree search methods. *Intelligence and AI*, 4(1):1–49, 2012. ISSN 1943-068X. doi: 10.1109/TCIAIG.2012.2186810.

L. Buesing, T. Weber, S. Racanière, S. M. A. Eslami, D. J. Rezende, D. P. Reichert, F. Viola, F. Besse, K. Gregor, D. Hassabis, and D. Wierstra. Learning and querying fast generative models for reinforcement learning. *CoRR*, abs/1802.03006, 2018.

J. Busquets, J. V. Busquets, D. Tudela, F. Pérez, J. Busquets-Carbonell, A. Barberá, C. Rodríguez, A. J. García, and J. Gilabert. Low-cost auv based on arduino open source microcontroller board for oceanographic research applications in a collaborative long term deployment missions and suitable for combining with an usv as autonomous automatic recharging platform. In *2012 IEEE/OES Autonomous Underwater Vehicles (AUV)*, pages 1–10, Sept 2012. doi: 10.1109/AUV.2012.6380720.

B. Chadwick, C. Katz, J. Ayers, J. Oiler, M. Grover, A. Sybrandy, J. Radford, T. Wilson, and P. Salamon. Gps drifter technologies for tracking and sampling stormwater plumes. In *OCEANS 2016 MTS/IEEE Monterey*, pages 1–10, Sept 2016. doi: 10.1109/OCEANS.2016.7761010.

K. Cobbe, O. Klimov, C. Hesse, T. Kim, and J. Schulman. Quantifying generalization in reinforcement learning. *CoRR*, abs/1812.02341, 2018. URL `http://arxiv.org/abs/1812.02341`.

K.-F. Dagestad, J. Röhrs, Ø. Breivik, and B. Ådlandsvik. Opendrift v1.0: a generic framework for trajectory modeling. *Geoscientific Model Development Discussions*, 2017:1–28, 2017. doi: 10.5194/gmd-2017-205. URL `https://www.geosci-model-dev-discuss.net/gmd-2017-205/`.

J. Das, F. Py, J. B. Harvey, J. P. Ryan, A. Gellene, R. Graham, D. A. Caron, K. Rajan, and G. S. Sukhatme. Data-driven robotic sampling for marine ecosystem monitoring. *The International Journal of Robotics Research*, 34(12):1435–1452, 2015. doi: 10.1177/0278364915587723. URL `https://doi.org/10.1177/0278364915587723`.

A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(6):1052–1067, June 2007. ISSN 0162-8828.

M. Deisenroth and C. Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, pages 465–472. Omnipress, 2011.

B. Douglas Luders, A. Cole Ellertson, J. How, and I. Sugel. Wind uncertainty modeling and robust trajectory planning for autonomous parafoils. 39:1–17, 05 2016.

G. Dudek and M. Jenkin. *Computational Principles of Mobile Robotics*. Cambridge University Press, New York, NY, USA, 2nd edition, 2010. ISBN 0521692121, 9780521692120.

G. Dudek, M. R. M. Jenkin, E. Milios, and D. Wilkes. A taxonomy for multi-agent robotics. *Autonomous Robots*, 3(4):375–397, Dec 1996. ISSN 1573-7527. doi: 10.1007/BF00240651.

J. L. Elman. Finding structure in time. *COGNITIVE SCIENCE*, 14(2):179–211, 1990.

C. Finn and S. Levine. Deep visual foresight for planning robot motion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2786–2793, May 2017. doi: 10.1109/ICRA.2017.7989324.

C. Finn, X. Y. Tan, Y. Duan, T. Darrell, S. Levine, and P. Abbeel. Deep spatial autoencoders for visuomotor learning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 512–519, May 2016. doi: 10.1109/ICRA.2016.7487173.

E. Fiorelli, N. E. Leonard, P. Bhatta, D. A. Paley, R. Bachmayer, and D. M. Fratantoni. Multi-AUV control and adaptive sampling in monterey bay. *IEEE Journal of Oceanic Engineering*, 31(4):935–948, 2006.

M. Garnelo, J. Schwarz, D. Rosenbaum, F. Viola, D. J. Rezende, S. M. A. Eslami, and Y. W. Teh. Neural processes. *CoRR*, abs/1807.01622, 2018. URL http://arxiv.org/abs/1807.01622.

M. Germain, K. Gregor, I. Murray, and H. Larochelle. Made: Masked autoencoder for distribution estimation. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 881–889, Lille, France, 07–09 Jul 2015. PMLR. URL http://proceedings.mlr.press/v37/germain15.html.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Google. Google earth: A planetary-scale platform for earth science data analysis. https://developers.google.com/earth-engine/datasets/. Accessed: 2018-11-10.

A. Graves, J. Menick, and A. van den Oord. Associative compression networks for representation learning. *CoRR*, abs/1804.02476, 2018. URL http://arxiv.org/abs/1804.02476.

R. Gray. Vector quantization. *IEEE ASSP Magazine*, 1(2):4–29, April 1984. ISSN 0740-7467. doi: 10.1109/MASSP.1984.1162229.

A. Guez, T. Weber, I. Antonoglou, K. Simonyan, O. Vinyals, D. Wierstra, R. Munos, and D. Silver. Learning to search with mctsnets. *CoRR*, abs/1802.04697, 2018.

I. Gulrajani, K. Kumar, F. Ahmed, A. A. Taïga, F. Visin, D. Vázquez, and A. C. Courville. Pixelvae: A latent variable model for natural images. *CoRR*, abs/1611.05013, 2016. URL http://arxiv.org/abs/1611.05013.

X. Guo, S. Singh, H. Lee, R. L. Lewis, and X. Wang. Deep learning for real-time atari game play using offline monte-carlo tree search planning. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3338–3346. Curran Associates, Inc., 2014.

M. H. S. Segler, M. Preuss, and M. P. Waller. Learning to plan chemical syntheses. 08 2017.

D. Ha and J. Schmidhuber. World models. *CoRR*, abs/1803.10122, 2018.

J. Hansen and G. Dudek. Coverage optimization with non-actuated, floating mobile sensors using iterative trajectory planning in marine flow fields. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2018. URL http://johannah.github.io/publications/iros2018driftercoverage.pdf.

J. Hansen, K. Kastner, A. Courville, =, and G. Dudek. Planning in dynamic environments with conditional autoregressive models. In *Prediction and Generative Modeling in Reinforcement Learning Workshop at International Conference on Machine Learning (ICML)*, pages 1–6, July 2018a. URL http://reinforcement-learning.ml/papers/pgmrl2018_hansen.pdf.

J. Hansen, S. Manjanna, A. Q. Li, I. Rekleitis, and G. Dudek. Autonomous marine sampling enhanced by strategically deployed drifters in marine flow fields. In *OCEANS'18 MTS/IEEE Charleston*, pages 1–7, October 2018b. URL http://johannah.github.io/publications/iros2018driftercoverage.pdf.

J. K. Hart and K. Martinez. Environmental sensor networks: A revolution in the earth system science? *Earth-Science Reviews*, 78(3):177 – 191, 2006. ISSN 0012-8252.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8): 1735–1780, Nov. 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL http://dx.doi.org/10.1162/neco.1997.9.8.1735.

T. Inanc, S. C. Shadden, and J. E. Marsden. Optimal trajectory generation in ocean flows. In *Proceedings of the 2005, American Control Conference, 2005.*, pages 674–679, June 2005. doi: 10.1109/ACC.2005.1470035.

M. Jakuba. Nui technical overview. http://www.whoi.edu/main/nereid-under-ice/technical-overview, 2014. Accessed: 2018-09-30.

JPL. Nasa jpl satellites dissect powerful hurricane matthew. https://www.jpl.nasa.gov/news/news.php?feature=6643, 2016. Accessed: 2010-09-30.

D. Kaiser, M. Babaeizadeh, P. Milos, B. Osinski, R. H. Campbell, K. Czechowski, D. Erhan, C. Finn, P. Kozakowski, S. Levine, R. Sepassi, G. Tucker, and H. Michalewski. Model-based reinforcement learning for atari. *CoRR*, abs/1903.00374, 2019.

N. Kalchbrenner, A. van den Oord, K. Simonyan, I. Danihelka, O. Vinyals, A. Graves, and K. Kavukcuoglu. Video pixel networks. *CoRR*, abs/1610.00527, 2016.

L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, Aug 1996. ISSN 1042-296X. doi: 10.1109/70.508439.

O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, volume 2, pages 500–505, March 1985. doi: 10.1109/ROBOT.1985.1087247.

K. Kim, D. Lee, and I. Essa. Gaussian process regression flow for analysis of motion trajectories. In *2011 International Conference on Computer Vision*, pages 1164–1171, Nov 2011. doi: 10.1109/ICCV.2011.6126365.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

L. Kocsis and C. Szepesvári. Bandit based monte-carlo planning. In J. Fürnkranz, T. Scheffer, and M. Spiliopoulou, editors, *Proceedings of the Seventeenth European Conference on Machine Learning (ECML 2006)*, volume 4212 of *Lecture Notes in Computer Science*, pages 282–293, Berlin/Heidelberg, Germany, 2006. Springer. ISBN 3-540-45375-X. URL http://www.sztaki.hu/~szcsaba/papers/ecml06.pdf.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, May 2017. ISSN 0001-0782. doi: 10.1145/3065386. URL http://doi.acm.org/10.1145/3065386.

D. Kularatne and A. Hsieh. Tracking attracting Lagrangian coherent structures in flows. In *Robotics: Science and Systems (RSS)*, 2015.

D. Kularatne, S. Bhattacharya, and M. A. Hsieh. Optimal path planning in time-varying flows using adaptive discretization. *IEEE Robotics and Automation Letters*, 3(1):458–465, Jan 2018. doi: 10.1109/LRA.2017.2761939.

A. Kwok and S. Martínez. A coverage algorithm for drifters in a river environment. In *Proceedings of the 2010 American Control Conference*, pages 6436–6441, June 2010. doi: 10.1109/ACC.2010.5531467.

N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell. A survey of mobile phone sensing. *IEEE Communications Magazine*, 48(9):140–150, Sept 2010. ISSN 0163-6804. doi: 10.1109/MCOM.2010.5560598.

H. Larochelle and I. Murray. The neural autoregressive distribution estimator. In G. Gordon, D. Dunson, and M. Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 29–37, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. URL http://proceedings.mlr.press/v15/larochelle11a.html.

S. M. Lavalle. Rapidly-exploring random trees: A new tool for path planning. Technical report, 1998.

S. M. LaValle. *Planning Algorithms*. Cambridge University Press, New York, NY, USA, 2006. ISBN 0521862051.

K. Li, T. Zhang, and J. Malik. Diverse image synthesis from semantic layouts via conditional IMLE. *CoRR*, abs/1811.12373, 2018. URL http://arxiv.org/abs/1811.12373.

N. Lipovetzky, M. Ramirez, and H. Geffner. Classical planning with simulators: Results on the atari video games. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 1610–1616. AAAI Press, 2015. ISBN 978-1-57735-738-4.

K. H. Low, J. M. Dolan, and P. Khosla. Adaptive multi-robot wide-area exploration and mapping. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1*, pages 23–30. International Foundation for Autonomous Agents and Multiagent Systems, 2008.

R. Lumpkin and S. Elipot. Surface drifter pair spreading in the north atlantic. *Journal of Geophysical Research: Oceans*, 115(C12):n/a–n/a, 2010. ISSN 2156-2202. C12017.

R. Lumpkin, M. Pazos, N. Oceanographic, and A. Administration. Measuring surface currents with surface velocity program drifters: the instrument, its data, and some recent results||. chapter two of lagrangian analysis. In *and Prediction of Coastal and Ocean Dynamics*. University Press, 2007.

T. Manderson and G. Dudek. Gpu-assisted learning on an autonomous marine robot for vision based navigation and image understanding. In *OCEANS'18 MTS/IEEE Charleston*, pages 1–7, October 2018.

S. Manjanna, N. Kakodkar, M. Meghjani, and G. Dudek. Efficient terrain driven coral coverage using gaussian processes for mosaic synthesis. In *13th Conference on Computer and Robot Vision (CRV), 2016*, pages 448–455. IEEE, 2016.

M. Meghjani, S. Manjanna, and G. Dudek. Multi-target rendezvous search. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016*, pages 2596–2603. IEEE, 2016.

V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, (7540): 529–533, 02 .

V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In M. F. Balcan and K. Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1928–1937, New York, New York, USA, 20–22 Jun 2016. PMLR. URL http://proceedings.mlr.press/v48/mniha16.html.

D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition,*

2004. CVPR 2004., volume 1, pages I–I, June 2004. doi: 10.1109/CVPR.2004.1315094.

NOAA. National oceanic and atmospheric administration. https://www.ncdc.noaa.gov/data-access. Accessed: 2018-11-10.

J. Oh, X. Guo, H. Lee, R. Lewis, and S. Singh. Action-conditional video prediction using deep networks in atari games. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, pages 2863–2871, Cambridge, MA, USA, 2015. MIT Press.

I. Osband, C. Blundell, A. Pritzel, and B. Van Roy. Deep exploration via bootstrapped dqn. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4026–4034. Curran Associates, Inc., 2016.

I. Osband, J. Aslanides, and A. Cassirer. Randomized prior functions for deep reinforcement learning. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 8617–8629. Curran Associates, Inc., 2018. URL `http://papers.nips.cc/paper/8080-randomized-prior-functions-for-deep-reinforcement-learning.pdf`.

T. Pepels, M. H. M. Winands, and M. Lanctot. Real-time monte carlo tree search in ms pac-man. *IEEE Transactions on Computational Intelligence and AI in Games*, 6(3):245–257, Sept 2014. ISSN 1943-068X. doi: 10.1109/TCIAIG.2013.2291577.

M. Plappert, M. Andrychowicz, A. Ray, B. McGrew, B. Baker, G. Powell, J. Schneider, J. Tobin, M. Chociej, P. Welinder, V. Kumar, and W. Zaremba. Multi-goal reinforcement learning: Challenging robotics environments and request for research, 2018.

A. Quattrini Li, I. Rekleitis, S. Manjanna, N. Kakodkar, J. Hansen, G. Dudek, L. Bobadilla, J. Anderson, and R. N. Smith. Data correlation and comparison from multiple sensors over a coral reef with a team of heterogeneous aquatic robots. In *International Symposium of Experimental Robotics (ISER)*, Tokyo, Japan, Mar. 2016.

S. Racanière, T. Weber, D. Reichert, L. Buesing, A. Guez, D. Jimenez Rezende, A. Puigdomènech Badia, O. Vinyals, N. Heess, Y. Li, R. Pascanu, P. Battaglia, D. Hassabis, D. Silver, and D. Wierstra. Imagination-augmented agents for deep reinforcement learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5690–5701. Curran Associates, Inc., 2017.

M. Rahimi, M. Hansen, W. J. Kaiser, G. S. Sukhatme, and D. Estrin. Adaptive sampling for environmental field estimation using robotic sensors. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3692–3698, 2005.

C. E. Rasmussen. Gaussian processes for machine learning. 2006.

N. D. Ratliff, D. Silver, and J. A. Bagnell. Learning to search: Functional gradient techniques for imitation learning. *Auton. Robots*, 27:25–53, 2009.

S. Reece, R. Mann, I. Rezek, and S. Roberts. Gaussian Process Segmentation of Co-Moving Animals. In A. Mohammad-Djafari, J.-F. Bercher, and P. Bessiére, editors, *American Institute of Physics Conference Series*, volume 1305 of *American Institute of Physics Conference Series*, pages 430–437, Mar. 2011. doi: 10.1063/1.3573650.

S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc., 2015. URL http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networ pdf.

D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of The 31st International Conference on Machine Learning (ICML)*, pages 1278–1286, 2014.

D. E. Rivera, M. Morari, and S. Skogestad. Internal model control: Pid controller design. *Industrial & Engineering Chemistry Process Design and Development*, 25(1):252–265, 1986. doi: 10.1021/i200032a041. URL https://doi.org/10.1021/i200032a041.

E. Rolf, D. Fridovich-Keil, M. Simchowitz, B. Recht, and C. Tomlin. A Successive-Elimination Approach to Adaptive Robotic Sensing. *ArXiv e-prints*, Sept. 2018.

O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4.

S. Ross, G. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In G. Gordon, D. Dunson, and M. Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 627–635, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. URL http://proceedings.mlr.press/v15/ross11a.html.

T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma. Pixelcnn++: A pixelcnn implementation with discretized logistic mixture likelihood and other modifications. In *ICLR*, 2017.

H. Salman, K. IDE, and C. K. R. T. JONES. Using flow geometry for drifter deployment in lagrangian data assimilation. *Tellus A*, 60(2):321–335, 2008. ISSN 1600-0870. doi: 10.1111/j.1600-0870.2007.00292.x.

S. Schaal. Is imitation learning the route to humanoid robots?, 1999.

A. F. Shchepetkin and J. C. McWilliams. The regional oceanic modeling system (ROMS): a split-explicit, free-surface, topography-following-coordinate oceanic model. *Ocean Modelling*, 9:347–404, 2005.

F. Shkurti, A. Xu, M. Meghjani, J. C. G. Higuera, Y. Girdhar, P. Giguere, B. B. Dey, J. Li, A. Kalmbach, C. Prahacs, K. Turgeon, I. Rekleitis, and G. Dudek. Multi-domain monitoring of marine environments using a heterogeneous robot team. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1447–1753, Portugal, Oct. 2012.

D. Silver. Lecture 8: Integrating learning and planning. http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching_files/dyna.pdf. Accessed: 2018-10-15.

D. Silver. *Learning Preference Models for Autonomous Mobile Robots in Complex Domains*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, December 2010.

D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman,

D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, Jan. 2016. doi: 10.1038/nature16961.

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

A. Singh, A. Krause, C. Guestrin, W. J. Kaiser, and M. A. Batalin. Efficient planning of informative paths for multiple robots. In *International Joint Conferences on Artificial Intelligence (IJCAI)*, volume 7, pages 2204–2211, 2007.

A. Singh, F. Ramos, H. D. Whyte, and W. J. Kaiser. Modeling and decision making in spatio-temporal processes for environmental surveillance. In *2010 IEEE International Conference on Robotics and Automation*, pages 5490–5497, May 2010. doi: 10.1109/ROBOT.2010.5509934.

L. . R. I. . O. M. . R. B. . M. J. . E. S. Slivinski, Laura Pratt.

N. N. Soreide, C. E. Woody, and S. M. Holt. Overview of ocean based buoys and drifters: present applications and future needs. In *MTS/IEEE Oceans 2001. An Ocean Odyssey. Conference Proceedings (IEEE Cat. No.01CH37295)*, volume 4, pages 2470–2472 vol.4, 2001. doi: 10.1109/OCEANS.2001.968388.

A. Stentz. *Optimal and Efficient Path Planning for Partially Known Environments*, pages 203–220. Springer US, Boston, MA, 1997. ISBN 978-1-4615-6325-9. doi: 10.1007/978-1-4615-6325-9_11.

R. S. Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *SIGART Bull.*, 2(4):160–163, July 1991. ISSN 0163-5719. doi: 10.1145/122344.122377. URL `http://doi.acm.org/10.1145/122344.122377`.

R. S. Sutton and A. G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 1998. ISBN 0262193981.

M. E. Taylor and P. Stone. Transfer learning for reinforcement learning domains: A survey. *J. Mach. Learn. Res.*, 10:1633–1685, Dec. 2009. ISSN 1532-4435. URL `http://dl.acm.org/citation.cfm?id=1577069.1755839`.

S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005. ISBN 0262201623.

J. Tompson, K. Schlachter, P. Sprechmann, and K. Perlin. Accelerating Eulerian Fluid Simulation With Convolutional Networks. *ArXiv e-prints*, July 2016.

A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. In *Arxiv*, 2016a.

A. van den Oord, N. Kalchbrenner, L. Espeholt, k. kavukcuoglu, O. Vinyals, and A. Graves. Conditional image generation with pixelcnn decoders. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4790–4798. Curran Associates, Inc., 2016b.

A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel Recurrent Neural Networks. *ArXiv e-prints*, Jan. 2016.

A. van den Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. van den Driessche, E. Lockhart, L. C. Cobo, F. Stimberg, N. Casagrande, D. Grewe, S. Noury, S. Dieleman, E. Elsen, N. Kalchbrenner, H. Zen, A. Graves, H. King, T. Walters, D. Belov, and D. Hassabis. Parallel wavenet: Fast high-fidelity speech synthesis. *CoRR*, abs/1711.10433, 2017a.

A. van den Oord, O. Vinyals, and k. kavukcuoglu. Neural discrete representation learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6306–6315. Curran Associates, Inc., 2017b.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017. URL `http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf`.

G. Villarini, P. V. Mandapaka, W. F. Krajewski, and R. J. Moore. Rainfall and sampling uncertainties: A rain gauge perspective. *Journal of Geophysical Research: Atmospheres*, 113(D11).

O. Vinyals, C. Blundell, T. P. Lillicrap, K. Kavukcuoglu, and D. Wierstra. Matching networks for one shot learning. *CoRR*, abs/1606.04080, 2016. URL `http://arxiv.org/abs/1606.04080`.

K. A. Wang, G. Pleiss, J. R. Gardner, S. Tyree, K. Q. Weinberger, and A. G. Wilson. Exact Gaussian Processes on a Million Data Points. *arXiv e-prints*, art. arXiv:1903.08114, Mar 2019.

G. Webster. Nasa's mars curiosity debuts autonomous navigation. `https://mars.nasa.gov/news/nasas-mars-curiosity-debuts-autonomous-navigation/`, 2013. Accessed: 2018-09-30.

T. C. Wilson, J. A. Barth, S. D. Pierce, P. M. Kosro, and B. W. Waldorf. A lagrangian drifter with inexpensive wide area differential gps positioning. In *OCEANS 96 MTS/IEEE Conference Proceedings. The Coastal Ocean - Prospects for the 21st Century*, volume 2, pages 851–856 vol.2, Sep 1996. doi: 10.1109/OCEANS.1996.568340.

F. Xia, A. R. Zamir, Z.-Y. He, A. Sax, J. Malik, and S. Savarese. Gibson Env: real-world perception for embodied agents. In *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*. IEEE, 2018.

A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi. Internet of things for smart cities. *IEEE Internet of Things Journal*, 1(1):22–32, Feb 2014. ISSN 2327-4662. doi: 10.1109/JIOT.2014.2306328.

Y. Zhao, F. Yin, F. Gunnarsson, F. Hultkratz, and J. Fagerlind. Gaussian processes for flow modeling and prediction of positioned trajectories evaluated with sports data. In *2016 19th International Conference on Information Fusion (FUSION)*, pages 1461–1468, July 2016.