Planning Informative Benthic AUV Surveys

Jackson Shields BE (Hons 1)

A thesis submitted in fulfillment
of the requirements of the degree of
Doctor of Philosophy

Australian Centre for Field Robotics
School of Aerospace, Mechanical and Mechatronic Engineering
The University of Sydney

March 2023

Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the University or other institute of higher learning, except where due acknowledgement has been made in the text.

Jackson Shields

23 March 2023

Attribution

The following first author publications have resulted from this thesis:

1. The conference paper 'Towards Adaptive Benthic Habitat Mapping' by Jackson Shields, Oscar Pizarro and Stefan Williams, appearing in *ICRA2020*. This paper has been expanded into Chapter 2. I designed the study, conducted the experiments, analysed the results and wrote the manuscript.

2. The journal paper 'Feature Space Exploration For Planning Initial Benthic AUV Surveys' by Jackson Shields, Oscar Pizarro and Stefan Williams, published in *Field Robotics* Vol. 3, pp. 652-686. This paper has been adapted into Chapter 3. I designed the study, conducted the experiments, analysed the results and wrote the manuscript.

In addition to the statements above, in cases where I am not the corresponding author of a published item, permission to include the published material has been granted by the corresponding author.

Jackson Shields, 28/04/2023

As supervisor for the candidature upon which this thesis is based, I can confirm that the authorship attribution statements above are correct.

Stefan Williams

Abstract

There is an ongoing global effort to explore and characterise marine environments, including the benthos or seafloor habitat, that is driven by research in ecology, geology and archaeology. Remotely-sensed data, such as bathymetry (a depth map of the seafloor), can be collected at large scales, however it does not have the required resolution to observe benthic habitats. To collect in-situ observations, special purpose Autonomous Underwater Vehicles (AUVs) are deployed to collect imagery of the seafloor communities. However, due to the small sensor footprint of the imagery, the sampling capacity of AUVs is limited relative to the vast areas that need to be surveyed. This creates the need to plan AUV trajectories that sample sparsely yet effectively. This thesis proposes methods for AUV sampling to meet this goal.

The seafloor structure can be observed through bathymetry which provides insights into the potential benthic habitats. When combined with in-situ samples, the relationship between the bathymetry and benthic habitats can be learned, allowing the prediction of the distribution of benthic habitats beyond where they can be feasibly sampled. The learned models can be used to create habitat maps that provide continuous predictions of the benthic habitat over the survey area. These maps are vital for monitoring, observation and communication to stakeholders, including government, industry and community. The creation of habitat maps is one of the key outputs of benthic AUV surveys. This thesis examines the creation of habitat models that can be used for adaptive sampling, where the quality of the model estimates should be capable of directing further sampling to improve the model fidelity. Bayesian Neural Networks (BNNs) are proposed for this task, which allow for the estimation of the uncertainty associated with a prediction. This in turn can be used to identify areas where further sampling will most improve the model. The validity of these uncertainty estimates is analysed to provide guidelines for use in directing further sampling.

Before a habitat model can be created, in-situ samples need to be collected. This thesis investigates the planning of AUV surveys that can efficiently characterise a new survey area, providing a seed dataset that can be used to train habitat models. As the remotely-sensed data provides insights into the environments observed, this thesis proposes sampling to comprehensively cover a feature-space representation of

the remotely-sensed data contained in the survey area. This will enable the survey to cover all the different classes of bathymetric terrain in its initial survey, while also observing all the benthic habitat types. Informative path planners are developed that plan freeform trajectories to explore the feature space while traversing the physical space. The resulting survey paths are shown to efficiently characterise a survey area with a limited budget, becoming an important tool for AUV survey planning.

The objective of many AUV surveys is the creation of habitat maps. Therefore, AUV sampling should be directed to maximise the accuracy of these habitat maps. AUV deployments are often conducted within a campaign, where AUVs are deployed multiple times in a given survey area. These campaigns are conceptualised as an active sampling cycle, where after each AUV deployment the data is downloaded and annotated. This data is then used to update the habitat model, which is in turn used to plan the next sampling trajectory. This thesis links informative path planning and active learning to prioritise sampling that will most improve the habitat model. Information objectives are designed that utilise the habitat model and the previous sampling locations to design an informative sampling trajectory for the AUV. Model-based, feature-based and hybrid active learning strategies are evaluated for the task of active AUV sampling. A hybrid strategy was shown to be the most effective, as it combines the model-based and feature-based approaches, which is beneficial as the model component directs it to areas that will most improve the model, while the feature component ensures there is diversity amongst the samples collected, and also allowing it to compensate for overconfidence in the habitat models.

The strategies developed in this thesis provide an avenue to efficiently conduct in-situ sampling with a limited deployment budget, which can be used to maximise output from AUV surveys. While this thesis focuses on planning informative AUV trajectories, it can be generalised to a variety of applications, where in-situ sampling is used to ground-truth remotely-sensed data. Alongside marine research, there are analogues to this task in agriculture and space exploration.

Acknowledgements

First to my supervisors, Oscar and Stefan. Thank you for inspiring me to get into marine research. It is a fascinating field and I now can't imagine doing anything else. There were many unique experiences throughout my PhD that wouldn't have been possible without your encouragement and support. Thank you for helping me think outside the box and develop our ideas into a thesis.

To my friends from ACFR, I never thought I'd get such a good crew doing our PhDs together. The brainstorming sessions, coffee breaks and post work drinks at the Rose all helped us get through it together.

To my other friends, thanks for always being there for me and always having a good time. It was always a much needed break. I look forward to hanging out a lot more now this chapter is coming to an end.

To my family, your unwavering love and support was essential in helping me get through this. Thank you for always believing in me. A special thanks to Mum who spent long hours editing my papers and this thesis with me.

To my loving partner, Molly. Thank you for always being there for me. Being with you makes it all worth it.

For my family and friends.

Contents

List of Figures

List of Tables

List of Algorithms

Nomenclature

List of Acronyms

ACFR	Australian Centre for Field Robotics
UAV	Unmanned Aerial Vehicle
AUV	Autonomous Underwater Vehicle
ROV	Remotely Operated Vehicle
GPS	Global Positioning System
SLAM	Simultaneous Localisation And Mapping
lidar	light detection and ranging *(also commonly known as 'ladar' or a 'laser range scanner')*
IMU	Inertial Measurement Unit
GP	Gaussian Process
RRT	Rapidly-exploring Random Tree
BNN	Bayesian Neural Network
MCTS	Monte Carlo Tree Search
RIG	Randomly-expanding Information Gathering
TSP	Travelling Salesman Problem
CNN	Convolutional Neural Network
DVL	Doppler Velocity Log
IMU	Internal Measurement Unit
USBL	Ultra-short Base Line
TRI	Terrain Ruggedness Index
VAE	Variational AutoEncoder
GMM	Gaussian Mixture Model
SVM	Support Vector Machine
VI	Variational Inference
SVI	Stochastic Variational Inference
VGP	Variational Gaussian Process
BALD	Bayesian Active Learning by Disagreement
OOD	Out Of Distribution
sonar	Sound Navigation And Ranging

UTM	Universal Transverse Mercator
MSE	Mean-Squared Error

Glossary

Bathymetric map A depth map of the seafloor.

Bathymetry The underwater topography of the seafloor.

Benthic Refers to anything that occurs near, or on, the seafloor.

Benthos The communities and habitats that occupy the region near, or on, the seafloor.

In-situ data Data collected in the vicinity of the target.

Remotely-sensed data Data collected without being in the vicinty of the target. Examples include observing the earth from airborne or satellite sensors, and the observation of the seafloor from ships.

Chapter 1

Introduction

1.1 Motivation

The ocean covers 71% of the earth's surface (Mayer et al., 2018) and yet it is still largely unexplored. While we are able to extensively map terrestrial environments with airborne and satellite-based optical sensors, the high attenuation of light in water prevents similar large-scale observations of marine environments. To map marine environments, including the seafloor, acoustic sensors are typically utilised, albeit with significantly lower resolution and discriminatory power when compared to optical sensors. To survey marine environments, acoustic sensors such as multibeam sonars are mounted on ships or underwater vehicles and collect depth observations of the seafloor. When combined, these depth observations create an underwater elevation map, commonly known as bathymetry (Brown et al., 2011a). Due to the smaller sensor footprint of these sensors and the relatively slow speed at which they collect data, the bathymetry coverage of the seafloor is limited. Collaborative programs such as the GEBCO Seabed 2030 project (Mayer et al., 2018) are driving further acoustic mapping of the seafloor. The collected bathymetric data is vital for uses in ecology, geology and archaeology (Brown et al., 2011a). One of the key objectives for ocean exploration is to observe the species that inhabit the seafloor and to investigate the habitats in which they exist. A habitat can refer to the communities of organisms

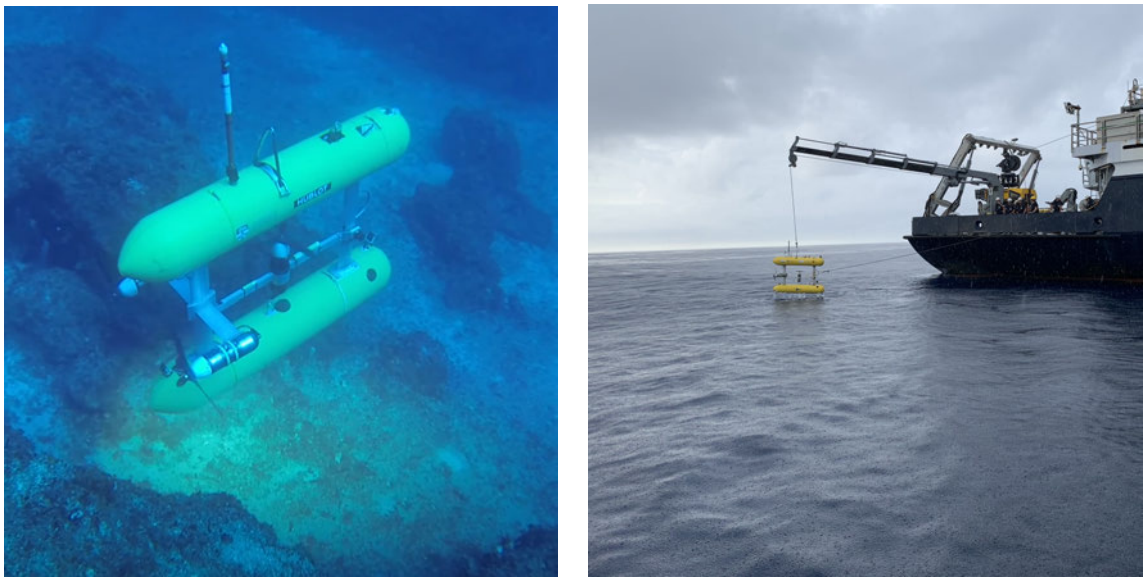
that exist together or the set of physical conditions that may allow these communities to exist (Greene et al., 2007). Spatially mapping these habitats is essential for ongoing monitoring programs (Baker and Harris, 2020). However bathymetric data alone does not contain enough information to directly deduce the habitats of the seafloor. Therefore, once broad-scale bathymetric data has been collected, the question becomes: what is down there? More precisely, what species and communities inhabit the seafloor structures observed in the bathymetric data.

Scientists deploy an array of instruments to observe and analyse the seafloor. Optical imagery is highly informative at perceiving the seafloor when collected at close range. Benthic-imaging AUVs (as shown in Figure 1.1) are underwater robots designed to collect imagery of the seafloor. They are vital for the collection of benthic imagery, due to their enhanced endurance, operational depth capability and navigational sensors which enables accurate localisation and the subsequent georeferencing of imagery. Due to the high attenuation of light in water, the robot needs to be positioned within several metres of the seafloor, which results in a small sensor footprint ($1 \rightarrow 10m^2$). This small sensor footprint, combined with the slow speed of the AUV, limits the area that can be visually surveyed. With this limited sampling capacity, where should the robot go to collect visual samples?

Broad-scale remotely-sensed data, such as bathymetry, can provide insights into the benthic habitat. Using bathymetry, the structure of the seafloor can be used to predict the benthic habitat (Brown et al., 2011a). The acoustic reflectance (backscatter) of the seafloor is also a useful indicator. Utilising the remotely-sensed data is critical to the planning of effective AUV surveys as it can guide the AUV to interesting areas, or ensure the full range of habitat types are explored. AUV surveys are typically planned manually by experts who analyse the remotely-sensed data to visit points of interest or to survey a given area. There is also the opportunity to plan surveys automatically.

One approach to planning informative sampling is to match the sampling strategy to the mission objectives. A key output of many benthic AUV surveys is the development of habitat maps, which relate the remotely-sensed data to in-situ observations

in order to predict the benthic habitat of an area, beyond where can be feasibly sampled. Therefore sampling should be conducted to enable the creation of accurate habitat maps. The objective of this thesis is to examine the coupling of AUV survey planning with the habitat modelling process, in order to *plan efficient AUV surveys* that *generate accurate habitat maps*.



(a) **(b)**

Figure 1.1 – (a) ACFR AUV Sirius collecting benthic imagery with its downward-facing stereo cameras. (b) ACFR AUV Sirius being deployed from R/V Falkor.

1.2 Problem Statement

There is a multifaceted need to survey the ocean floor, however due to the large areas that need to be surveyed, full coverage can only be achieved with remote sensing. Although remotely-sensed data can be collected at scale, it is not discriminative enough and does not provide enough 'resolution' to observe the species and habitats of the benthos. Direct observation of the seafloor is needed to support scientific studies. However the sampling capacity of these sensors is limited compared to the areas to be surveyed, creating the need for sparse yet effective sampling strategies and the ability to extrapolate these samples beyond where can be feasibly sampled. Under

the essential assumption that there is a better than random relationship between remote sensing observations and in-situ visual observations, this thesis investigates the related problems of:

1. Modelling the relationship between the low-resolution remotely-sensed data to the sparse in-situ observations to allow for inference beyond where can be feasibly sampled.
2. Planning AUV sampling trajectories to characterise an environment given prior remotely-sensed data.
3. Leveraging remotely-sensed data and prior observations to efficiently improve an existing model through additional sample collection.

1.3 Contributions

The objective of this thesis is to develop informative path planning capabilities for AUV surveys, to allow efficient characterisation of a survey area as well as the creation of accurate habitat maps. There is limited existing research in this area, which leads to the novel, overarching contribution of this thesis:

- **Active sampling for benthic AUV surveys**: Leveraging remotely-sensed data to plan informative AUV trajectories that aim to maximise the accuracy of habitat maps. An innovative active learning approach is used for sample prioritisation that enables better utilisation of the limited sampling capacity of the AUV.

Supporting this contribution are the following core contributions of this thesis:

- **Bathymetric Feature Extraction** - use of autoencoders to compress bathymetric patches into a compressed but expressive latent space that represents the structure of the seafloor. This feature extraction method can be leveraged

to train more accurate habitat models than geometric features. This also enables unsupervised clustering to generate spatial clusters that have a higher correlation to the benthic habitat than using geometric features.

- **Bayesian Neural Networks for Habitat Modelling** - development of BNNs for habitat modelling. These networks allow for the accurate prediction of the habitat and an estimation of the corresponding uncertainty. This uncertainty estimate is crucial for the planning of future AUV surveys. This method is compared against other probabilistic habitat models, based on the accuracy and validity of the uncertainty estimates.

- **Informative path planning for feature space exploration** - conceptualisation of the problem of exploring a feature space representation of remotely-sensed data by traversing the physical space. This thesis focuses on the case of exploring the bathymetric feature space to plan informative initial benthic AUV surveys. The seafloor structure is indicative of the benthic habitat and ensuring the AUV samples from the complete range of bathymetric terrain is a useful way to ensure the AUV observes every benthic habitat present in the survey area.

- **Informative Path Planners** - development of informative path planners that leverage information objectives to plan informative AUV samples. Planners based on RRTs and MCTS are developed that utilise information objectives to plan trajectories of the AUV to collect the highest information reward given the budget constraint. These planners are capable of collecting samples that *together* optimise the information reward.

- **AUV Campaign as an Active Learning Cycle** - interpreting an AUV campaign as a sequence of dives with the aim to improve the underlying habitat model. By labelling the collected imagery and training a habitat model in between deployments of the AUV, further AUV trajectories can be optimised to visit areas of that will most improve the model.

- **Feature and model-based active learning for informative path planning** - links active learning and informative path planning by creating objective functions that reward visiting areas that will most improve the model. Feature-based active learning techniques reward the model for exploring the feature space, while model-based techniques use the habitat model's uncertainty to direct further sampling. Combining these approaches allows the AUV to both explore the feature space and exploit the model uncertainty, while also ensuring the samples collected are *together* informative.

While this thesis focuses on the problem of visiting diverse bathymetric terrain with an AUV, the contributions are applicable to any application where in-situ sampling is planned from remotely sensed data. Analogues to this problem exist in space exploration (Li et al., 2005) and agriculture monitoring (Ramin Shamshiri et al., 2018).

1.4 Outline

Chapter 1 introduces this thesis, provides an overarching motivation for this work. It then defines the problem statement and provides an outline of the contributions proposed in this thesis.

Chapter 2 investigates habitat modelling which establishes the basis for this thesis. It provides an analysis of feature extraction for bathymetry and proposes probabilistic classifiers that can be effective for habitat modelling. It analyses the uncertainty estimates which provides the basis for adaptive sampling.

Chapter 3 explores the problem of collecting an informative initial survey. It proposes using feature-based active learning techniques to comprehensively cover the feature space. It links these techniques with informative planners that can be used to plan freeform trajectories that jointly explore the feature and physical spaces.

Chapter 4 investigates planning informative AUVs surveys when there are prior surveys available. It conceptualises an AUVs campaign as an active learning cycle, where

the goal of the deployments is to maximise the accuracy of the habitat model. Both model and feature-based active learning techniques are evaluated, with this research finding significant improvement when combining these approaches. Figure 1.2 shows this active learning cycle and how it brings together the methods of each chapter.

Chapter 5 concludes this work, providing a summary of the contributions contained in this thesis and identifying avenues of future research.

Each chapter is written to be standalone and each contains a literature review for content in the corresponding chapter.

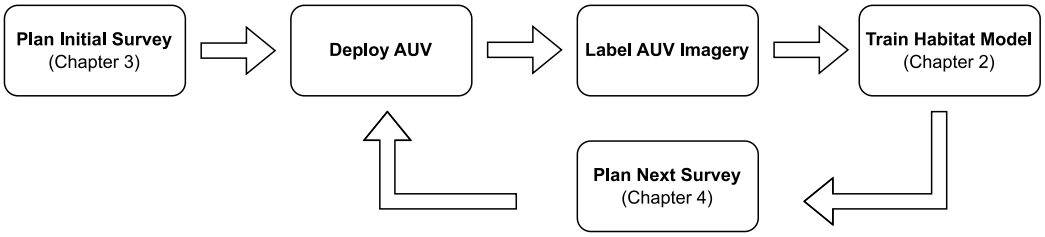


Figure 1.2 – Overview the active learning cycle for benthic AUV surveys, as proposed in this thesis. Chapter 3 is used to plan an informative initial survey that explores the bathymetric feature space and seeds the active sampling cycle. An AUV is then deployed to collect georeferenced seafloor imagery. Chapter 2 relates the remotely-sensed data to the in-situ observations using probabilistic habitat models, that can identify where should be sampled next. Chapter 4 plans informative AUV trajectories that aim to most improve the habitat model.

Chapter 2

Habitat Modelling

2.1 Introduction

Marine scientific surveys are conducted in support of a range of marine science research including ecology, geology and archaeology (Valentine et al., 2010; Williams, 2016; Williams et al., 2012). One such application is the creation of habitat maps, that predict the benthic habitat for an entire survey area. These habitat maps are useful tools for communicating with stakeholders in the marine environment and they play a crucial role in the creation and management of marine parks (Baker and Harris, 2020). Due to the vast areas that need to be surveyed, it is infeasible to collect full coverage in-situ observations of the benthic habitat. Therefore, in-situ observations need to be paired with remotely-sensed data that can be collected for large areas. Habitat mapping is the process of relating this relatively low-resolution remotely-sensed data to the sparse in-situ observations, in order to predict, beyond where can be feasibly sampled, the benthic habitat of the survey area.

Autonomous Underwater Vehicles (AUVs) are crucial for conducting benthic surveys, as they can collect data in areas that are inaccessible to humans, they have greater endurance and they can collect a wealth of data using a range of onboard sensors (Singh et al., 2004). However, due to the strong attenuation of electromagnetic radiation in water, visual data has to be collected close to the target, resulting in a

relatively small sensor footprint. Even given the enhanced endurance of current generation AUVs, the area they can visually sample from is limited by this small sensor footprint, and often results in limited or sparse coverage of the survey area.

Remotely-sensed data is captured by sensors with large sensor footprints, allowing coverage over vast areas. Acoustic sensors are primarily used underwater, as sound signals have comparatively low attenuation in water. Bathymetry, which is an elevation map of the seafloor, is collected by multibeam sonars, mounted on ships or high-flying AUVs. For shallow water depths, bathymetry can also be collected efficiently from aeroplanes with light detection and ranging (lidar) or from satellites (Morgan, 2021). The depth and structure of the seafloor can provide insights into the benthic habitat (Brown et al., 2011a). A habitat map can be created from relating this low-resolution bathymetric data to the high resolution in-situ observations captured from the AUV.

Machine learning algorithms are used to learn the relationship between the remotely-sensed data and the habitat observations. Neural networks have demonstrated to be effective at learning complex relationships, but do not produce a probabilistic estimate and therefore cannot effectively identify what they do not know. Utilising probabilistic models (Bender et al., 2012a) allows the user to identify areas that are underrepresented in the training data, and where further sampling could improve the model.

If a key output of a marine survey is the characterisation of the survey area and the creation of habitat maps, the limited sampling budget for the survey should be allocated to efficiently create accurate habitat models. This chapter explores the creation of habitat maps that can be used to guide further exploration. It focuses on bathymetry as the remotely-sensed data source and investigates an unsupervised approach for bathymetric feature extraction. Using the bathymetric features as input, this research proposes the use of a probabilistic neural network for benthic habitat mapping as it allows for accurate classification whilst also providing an uncertainty estimate that is crucial to direct further learning. The habitat mapping process is demonstrated in Figure 2.1. This chapter also analyses the validity of these uncer-

tainty estimates, in the context of marine habitat mapping. Furthermore, this chapter investigates clustering using the bathymetric features, which can be used to identify areas of similar terrain without the need for habitat observations. Finally, this chapter assesses the data needs of habitat models, that can be used to scope the sampling budget for benthic AUV surveys.

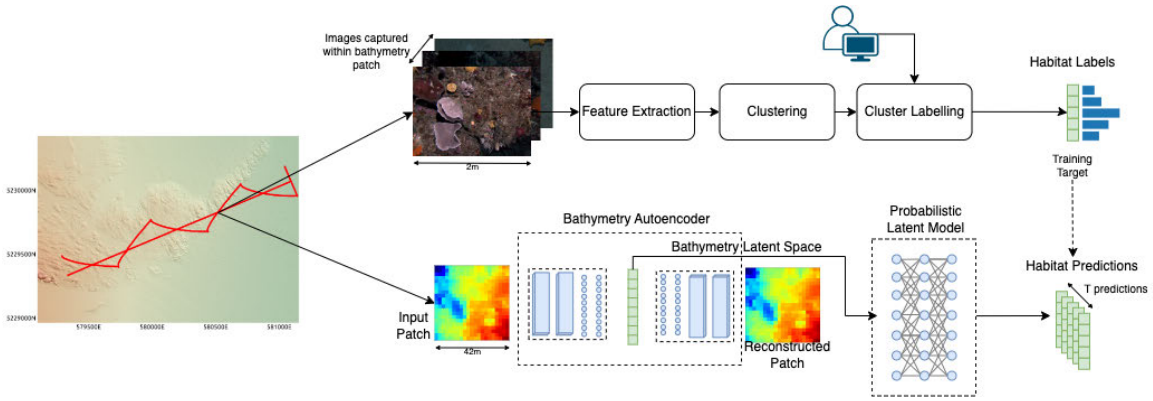


Figure 2.1 – Process Diagram. (Top row) Images are first clustered and then consolidated by an expert to yield a small set of ecologically relevant class labels. As the bathymetric patches are significantly larger than the footprint of an image, the distribution of habitat labels within each patch is calculated. (Bottom row) Bathymetric patches are fed through an autoencoder that yields a latent space used to train a Bayesian neural network. The network is trained to classify unseen patches using the training targets provided by the image-based habitat labels.

The contributions presented in this chapter are:

- Learning descriptive bathymetric features using an autoencoder, which is trained by randomly sampling patches from a bathymetric map.
- Performing unsupervised clustering with these learnt features to identify regions of similar bathymetric features. This is useful for examining a survey area before samples have been collected.
- Using Bayesian Neural Networks (BNNs) for habitat modelling, which allows for the estimation of the uncertainty associated with a prediction. This provides the basis for adaptive sampling tasks in Chapter 4.
- A comparison of several probabilistic approaches to habitat modelling, examining the classification accuracy and data requirements.

- An analysis of the uncertainty estimation for each of these models, for a unique dataset that has naturally high data noise.

The remainder of this chapter is as follows: Section 2.2 provides a review of the related literature, Section 2.3 presents an overview of bathymetry, Section 2.4 explains how benthic imagery is labelled, Section 2.5 looks at the data used for this chapter, Section 2.6 investigates bathymetric feature extraction, Section 2.7 explores clustering with bathymetric features, Section 2.8 presents the probabilistic habitat models developed in this work, Section 2.9 contains an overview of Gaussian Processes (GPs) which are used as a benchmark method, Section 2.10 evaluates the habitat models, Section 2.11 analyses the uncertainty estimates of the classifiers. Finally, Section 2.13 summarises this chapter.

2.2 Background

A benthic habitat is a seafloor area, that has a set of physical and environmental characteristics that support certain biological communities (Brown et al., 2011a). Although not explicitly linked to the occurrence of a particular species, a habitat can be thought of as a 'potential habitat' (Greene et al., 2007) for a given species, where given conditions exist that are likely to support the species.

Habitat maps can be either continuous or discontinuous (Brown et al., 2011a). A continuous habitat map predicts the likelihood of occurrence of a single habitat type, given the remotely-sensed data. For example, predicting the likelihood of a coral habitat being observed. These are known as biota predictors. A discontinuous habitat assigns a single habitat label to each location. With this type of habitat map, it is assumed that one habitat ends where another begins. This is often incorrect, with the boundary between habitats often consisting of a combination of the neighbouring habitats (Lucieer and Lucieer, 2009). The advantage of a discontinuous habitat map is the easier interpretation by end-users.

A habitat map allows the visualisation and interpretation of the benthic habitat, often from relatively sparse in-situ observations. Directly interpreting the remotely-sensed data is difficult and often there is a complex relationship between the remotely-sensed data and the observed habitat. Habitat maps are crucial for designing and monitoring marine resource monitoring programs (Baker and Harris, 2020). They form a key role in communicating information about the marine areas to stakeholders, including demonstrating the need for marine parks or visualising the changing environment. Habitat mapping can be used to identify critical habitats for at-risk species, representative habitats of greater management areas, areas of significant biodiversity and iconic features (Baker and Harris, 2020).

When using bathymetry for habitat mapping, both the depth and the topography of the seafloor can be used as indicators of the benthic habitat. Wilson et al. (2007) uses a set of geometric features to summarise the structure of the seafloor for habitat modelling. These features, including slope, aspect, curvature and ruggedness, are key drivers of benthic habitat formation and can be used to predict the benthic habitat. Extracting these features at different scales provides further information and can improve the classification process (Wright and Heyman, 2008). The bathymetry needs to be gridded at a relatively high resolution in order for it to be effective for habitat mapping (Calvert et al., 2015). The acoustic reflectance of the seafloor can also provide insights into the benthic habitat. Backscatter or side-scan sonar records the strength of the received sonar pulse. As different seafloor substrates reflect the sonar pulse differently, the relationship between the strength of the return and the seafloor substrate or benthic habitat can be learnt (Brown et al., 2011b). A key challenge with the use of sonar reflectance is the calibration and consistency of measurements Brown et al. (2019). Each individual sonar has different transmitter and receiver characteristics, meaning that the measurements from one unit cannot be easily applied to another. While data captured with a single unit can be cohesive, collating data from different surveys needs to be carefully calibrated. Brown et al. (2019) collected multispectral multibeam sonar across the same track from multiple years, with the same model of multibeam sonar but different units. They showed a significant

difference in the strength of the returns between the years. Despite the calibration difficulties, habitat modelling with backscatter is effective, with a clear relationship between the different benthic habitats and the multispectral backscatter data. The bathymetry and backscatter are complimentary and can both be simultaneously used for habitat mapping. Marsh and Brown (2009) use self-organising maps to train an unsupervised bathymetry and backscatter classifier. This classifier groups similar areas of bathymetry and backscatter, however it has no notion of the habitat classes present in these groups, and further classification is needed to predict the habitat class.

For shallower areas, satellite or airborne optical or hyperspectral sensors can be used to observe the marine habitats. This is effective for observing and mapping coral habitats, which predominantly form in shallow areas. The Allen Coral Atlas (Allen Coral Atlas, 2020) is a large-scale habitat observation and mapping project, that utilises satellite multispectral data to comprehensively map coral regions, providing an important snapshot of the current reef systems and a platform for ongoing observation and management. The multispectral data collected from the satellites is first processed to remove atmospheric and sea-surface effects. The bathymetry is estimated from the satellite observations, to estimate the attenuation of the observed frequencies in the water column. From this, the relationship between the corrected multispectral data and the benthic habitat is learnt from a random forest classifier (Lyons et al., 2020). The benthic habitat labels are sourced from diver surveys or existing high-confidence habitat maps.

Auxiliary oceanographic data can also provide useful context for habitat mapping, as this information is linked to the biological processes. Data sources that can be utilised include currents, temperature, waves and chlorophyll (Brown et al., 2011b; Lyons et al., 2020). This information can be interpolated from point-based samples or observed from satellite data.

The remotely-sensed data needs to be matched with ‘ground-truth’ habitat observations, as the habitat cannot be directly observed from these data sources. Diver or snorkel based surveys are the simplest way to collect in-situ habitat observations,

however the capability of surveys conducted by divers is limited. The diver-based surveys are limited to shallow waters where divers can reach the seafloor. The extent of these surveys is further limited by the limited endurance of humans when swimming. Furthermore diving is dangerous and efforts should be taken to minimise risk. Finally, there are navigation and localisation issues when conducting seafloor diver-based surveys that can impact the accuracy of the seafloor surveys. It is very difficult to navigate underwater, as the primary mode for human navigation is visual observations, which are limited underwater. Divers need to be equipped with acoustic positioning systems such as a Doppler Velocity Log (DVL) or Ultra-short Base Line (USBL) acoustic positioning system to localise to the required accuracy underwater. Due to the significant limitations to diver-based surveys, the habitat is typically observed through imagery taken of the seafloor (Brown et al., 2011a). To collect habitat observations at individual points, 'drop cameras' can be used (Kostylev et al., 2001), which are waterproof camera housings that are lowered to observe the seafloor. Although simple, this method is slow and has limited sampling capacity. To collect more data, towed video can be used, which involves towing a camera system behind a boat, that is controlled to hover above the seafloor (Rattray et al., 2009). When using this platform, long transects are collected enabling for significantly more data to be collected than a drop camera. However these platforms cannot control their altitude with the required response and therefore they cannot be used over areas where there are large changes in depth, such as rugose reefs or cliffs. Furthermore, they cannot go into the littoral (close to shore) as the towing vessel needs to be able to turn around.

Robotic underwater vehicles have revolutionised marine habitat surveys by offering greater endurance, improved sensor capabilities, access to new environments and improved safety for human operators (Wynn et al., 2014). Remotely Operated Vehicles (ROVs) are tethered to a support vessel, where they are typically controlled by human operators. Autonomous Underwater Vehicles (AUVs) are untethered and autonomously conduct the seafloor survey, typically by following a set of waypoints. These autonomous vehicles have a much greater exploration capability, with vehicles

reaching full ocean depth, exploring under ice (Jakuba et al., 2018) and the littoral (Schultz et al., 2017). The much greater endurance greatly expands the area that can be sampled, with AUVs such as the AutoSub Long Range having a range of 1800km (Roper et al., 2021). The sensor payloads of robotic underwater vehicles allow for enhanced localisation and navigation capabilities (Paull et al., 2014). Many underwater vehicles are equipped with an Internal Measurement Unit (IMU), DVL and a USBL which through sensor fusion can aid in accurate localisation. Furthermore, some vehicles use Simultaneous Localisation And Mapping (SLAM) from cameras or sonar to further refine the localisation estimate (Williams et al., 2009). Benthic imaging AUVs, such as the Seabed platform (Singh et al., 2004), use altimeters to hover a constant altitude from the seafloor. This ensures the imagery is in focus and has consistent lighting, which greatly improves the delivered data product. The ability to deliver accurate and reliable georeferenced imagery of the seafloor has made autonomous platforms a key component for underwater surveys.

Benthic habitat mapping is the process of relating the remotely-sensed and auxiliary data (input data) to the habitat observations (ground truth), which is an example of machine learning. Machine learning is the process of using computing algorithms to find patterns in the data (Lecun et al., 2015). Benthic habitat mapping is primarily supervised, where the algorithm learns to map the inputs to the ground truth labels. In the absence of ground truth labels, unsupervised learning can be utilised to identify patterns or groupings in the data. The development of machine learning algorithms has been driven by the search for 'artificial intelligence' and the need to quantify an increasing amount of data (Lecun et al., 2015). These developments have been applied to marine benthic habitat mapping. Decision trees are a logical formalisation of a decision process, which was formalised into the CART (Classification And Regression Tree) machine learning algorithm (Brieman, 2001). Decision trees are explainable as they show how each feature in the data contributed to the overall decision. Rattray et al. (2009) use a decision tree classifier to predict the benthic habitat from both bathymetry and backscatter. A primary limitation of decision trees is they are prone to overfitting. To overcome the overfitting issue of decision trees, Brieman (2001) pro-

posed random forests use an ensemble of decision trees all with different combinations of features, which avoids the decision being solely focused on a single feature. The decision is based on the mean value of all the individual trees. Random forests are effective for habitat mapping and have been deployed in large-scale habitat mapping projects (Lyons et al., 2020).

Another foundational classification algorithm is the Support Vector Machine (SVM) (Cortes and Vapnil, 1995). SVM is an iterative algorithm that learns the decision boundary between features. In its basic form, an SVM can only be used to solve linearly separable problems. However, a kernel function, can be used to project the data into a higher dimension and can subsequently solve linearly inseparable problems. SVMs are versatile and can be readily applied to habitat mapping problems. Hasan et al. (2012) compares several supervised machine learning algorithms and finds that SVM and Random Forests have the highest classification accuracy for benthic habitat mapping with backscatter data.

Recent progress in machine learning can be largely attributed to neural networks and deep learning methods. Neural networks were introduced in 1986 (Rumelhart et al., 1986) as the multilayer perceptron. The perceptron (Rosenblatt, 1958) is a biologically-inspired machine learning algorithm, that mimics the activation of a neuron. While a single neuron has little discrimination ability, a network of neurons can be used to learn abstract patterns in the data. A neural network is trained using backpropagation, where the error is fed backwards through the network to update the weights. The uptake of neural networks was originally limited, due to the problem of vanishing gradients and the high computation requirements. However, recent advances in neural network theory, new computing hardware (GPUs) and creation of large labelled datasets (Russel and Norvig, 2014) has driven major progress in the capabilities of neural networks. Convolutional Neural Networks (CNNs)s have been an exceptional driver of this progress. A convolutional network is formed from stacked convolutional layers. A convolutional layer is comprised of a set of filters, with each of these filters performing a kernel operation on the input. These filters are trained so that effective features are extracted from each layer. The first layers of a CNN

for computer vision are typically edge, corner or colour gradient detectors (Selvaraju et al., 2017). As the layers progress, the features become more abstract, forming complex relationships between the features. Interestingly the filters generated from supervised training are formed in such a way that the features progressively segregate the classes (Oyallon, 2017). The effectiveness of CNNs and deep learning methods are highlighted through the large-scale image classification challenge ImageNet(Deng et al., 2009), where the classification error rate improved from 26% to 2.3% with the introduction of CNNs. Now, machine learning for image classification is now well-established and is used for a wide variety of purposes including self-driving cars (Chen et al., 2016) and cancer detection (Sirinukunwattana et al., 2016). Machine learning methods are not limited to image classification, and can be utilised on a variety of abstract data sources, including satellite imagery (Audebert et al., 2018), hyperspectral data (Windrim et al., 2018) and point cloud data (Engelcke et al., 2017).

Data interpretation has typically been a two-stage process, with the first stage the extraction of useful information, and the second the translation of this information into a useful interpretation, for example a classification. The feature extraction process involves transforming the data to focus on interesting patterns or features. Historically this has been a manual process performed by specialists, who use prior knowledge of the data to select features that match or differentiate the data. In computer vision, examples of these features are edge, colour and corner detectors (Bewley et al., 2012). For bathymetry data, this involves extracting geometric features such as slope, ruggedness and aspect (Wilson et al., 2007). Although many of these methods are still state-of-the-art, they do not generalise well and require specialists to select appropriate feature extractors. Advances in machine learning, particularly neural networks, can be used to create and combine features, enabling greater generalisation and the ability to learn from complex data sources. These advances have also led to new approaches to habitat modelling, as exemplified by Rao et al. (2017) who utilise a convolutional autoencoder for feature extraction of both bathymetry and co-located imagery, enabling the co-learning of the relationship between these two modalities.

A machine learning algorithm should have a measure of uncertainty. If presented

with out-of-distribution data, data unlike any it has seen before, it should be able to recognise that it is uncertain about it. However, many machine learning algorithms do not have this capability. Classification neural networks are typically trained with a softmax output activation function, that predicts the likelihood the input data belongs to a specific class, but this is generally overconfident and has no mechanism for epistemic uncertainty (Gal and Ghahramani, 2016). This capability is essential for the deployment of machine learning algorithms in safety critical applications, such as self driving cars. Gaussian Processes (GPs) are sets of random variables, where each combination of these variables form a Gaussian distribution (Rasmussen and Williams, 2006). When optimised to fit the training data, a GP provides a posterior distribution, which can be used to predict the mean and uncertainty of an output variable given input data. GPs can be used for regression and classification. Bender et al. (2012b) utilises GPs for habitat mapping, using the geometric features of depth, slope, aspect and rugosity extracted from the bathymetry to predict the broad-scale habitat class. The GP can be used to estimate the uncertainty with a prediction, which for habitat mapping can provide an insight into what areas are underrepresented in the training data and where further samples should be allocated. Building on this, Bender (2013) uses the uncertainty estimate from this GP to plan a survey that will most improve the model. Historically, GPs scaled poorly with the number of training data points, however advances in GP approximation have largely solved this limitation (Hensman et al., 2015). Compared to neural networks, GPs have lesser discriminatory power and rely on an effective feature representation as inputs. There is interest in creating probabilistic neural networks, which can allow for increased accuracy, use with large datasets and an uncertainty estimate associated with the prediction. Neal (1996) proposes a Bayesian neural network, where each weight and bias is represented as a Gaussian distribution. Training this network is complex and does not scale well, as it relies on Bayesian inference of a large system. Blundell et al. (2015) utilise Stochastic Variational Inference (SVI) to train the network, providing a method 'Bayes by Backprop' that runs error propagation on the Bayesian network. During inference, the weights from the network are sampled from and the posterior is approximated from multiple draws from the network. Bayesian neural networks

trained with SVI are known to have poor scalability compared to standard neural networks, preventing their use in many applications (Ovadia et al., 2019). To adapt a neural network to be probabilistic, Gal and Ghahramani (2016) propose Monte Carlo Dropout (MC Dropout) where dropout is left on during inference. Dropout is a regulariser often used when training a network, which randomly sets a given percentage of the activations of each layer to zero, which has the effect of preventing overfitting by necessitating there be multiple pathways to explain the data. This is usually turned off during inference, but leaving it on has the effect of sampling a different network during every sample, which is akin to having an ensemble of networks. The posterior can be estimated through Monte Carlo sampling of this network. Monte Carlo dropout allows a probabilistic network to be created from networks that are far larger in scale than is possible using a Bayesian neural network, allowing probabilistic estimates from deep neural networks. Creating an ensembling of models can also be used to provide an approximation of a probabilistic neural network, with the posterior distribution estimated from the predictions of each of the models (Lakshminarayanan et al., 2017). The degree to which neural networks can be utilised to provide well-calibrated uncertainty estimates is contested and an active area of research. Ovadia et al. (2019) compares the accuracy and uncertainty estimates of probabilistic classifiers on out-of-distribution data. They observe that for small models, *SVI* is the most effective but cannot be use for larger models. and for larger models, *Ensembles* are the best approximation, followed by *MC Dropout*. These approaches outperform temperature scaling (Guo et al., 2017). They remark that there is still significant room for improvement for probabilistic neural networks. Probabilistic models are critical for habitat modelling, as they offer insights into areas that are under sampled and where further sampling could improve the model.

Supervised learning, is where a model is used to learn the relationship between inputs and some ground-truth data. However this ground-truth data can often be expensive to collect and label. This is true for habitat modelling, where the in-situ observations are more expensive to collect. Furthermore, labelling the in-situ observations is a time-consuming and expensive process and a major bottleneck in the data collection

pipeline (Steinberg, 2012). Unsupervised learning involves finding patterns in the data, without the use of ground-truth labels. This type of learning is attractive, as it is easy to collect vast volumes of data but difficult to find useful insights from it. A common form of unsupervised learning is clustering, where groupings of similar data points are identified, which is achieved through algorithms such as K-means and GMMs. The difficulty with clustering is that the clusters identified by the clustering algorithm may not correspond to human interpretations of the data, which is especially true for higher dimensional data. While many clustering algorithms (including K-means and GMMs), need the number of clusters specified, Steinberg (2012) proposes non-parametric clustering methods that automatically detect the number of clusters, which is applied to the annotation of AUV imagery. When the clusters generated are homogeneous and each contain examples of the different benthic habitat, the imagery can be efficiently annotated and the labelling bottleneck eliminated. Clustering can also be performed on remotely-sensed data to identify regions of similarity. Calvert et al. (2015) performs this on bathymetry using the geometric features of depth, slope, rugosity, curvature and aspect to find areas of similar terrain, which provides an insight into the potential habitats found in these regions.

Unsupervised learning can also be used to relate two data sources. Rao et al. (2017) use a Restricted Boltzmann Machine (RBM) to study the interrelationship between the imagery and bathymetry, without the need for ground-truth labels. This framework allows the prediction of one variable in the absence of the other; for example predicting areas on the bathymetry that are likely to contain imagery similar to a query image, and vice versa. Unsupervised learning provides an opportunity to learn from the vast quantities of data collected, however the use of supervised learning is still necessary in some applications, in order to force the model to learn the required relationship.

Habitat maps are a useful output of benthic AUV surveys, providing easy to interpret visualisation for key stakeholders in marine environment management. Remotely-sensed data can be collected efficiently from acoustic surveys and satellite/airborne data, however it needs ground-truth from in-situ observations. Machine learning

methods can be used to create habitat models that relate the broad-coverage remotely-sensed data to sparse in-situ observations. Neural networks have been shown to have impressive learning capabilities, with the ability to learn from complex data sources. They have the capability to learn informative features, that has demonstrated to outperform hand-crafted feature extraction methods. A key deficiency of neural networks is the lack of probabilistic output, meaning they are often overconfident in their predictions and they cannot identify out-of-distribution data. Probabilistic neural networks provide an uncertainty estimate associated with the prediction. This can be utilised to identify areas that are underrepresented in the training data and can be used to guide further sampling.

2.3 Bathymetry Collection

A bathymetric map is a depth map of the seafloor, that is useful for both navigation and scientific research. As imagery cannot be obtained for large areas underwater, the bathymetry is the main method of observing most of the seafloor. Originally bathymetry was collected using a 'line-and-sinker' approach, where a weight was attached to a piece of rope (later to piano string) and the length of the line when the weight touched the bottom was measured (Dierssen and Theberge, 2014). The accuracy and speed of measuring the bottom was limited using this method. The invention of sonar in 1914, where the return travel time for a sound wave to be transmitted from a sonar transmitter, deflected off an object (such as the seafloor) and received by the sonar receiver is measured, and using the speed of sound in water is used to calculate the distance to the object, as illustrated in Figure 2.2. This method, now known as single-beam echosounding was used to map much of the oceans, but to poor spatial resolution. The advent of multibeam echosounding combined with accurate localisation from the fusion of GPS and IMU sensors, enables the creation of very accurate bathymetry maps with high spatial resolutions. An example of this is the bathymetric dataset used in this chapter, covering the Fortesque region of Tasmania, collected by GeoScience Australia (Spinoccia, 2011) and is gridded at a

spatial resolution of 2 meters. An example of this data is presented in Figure 2.3. More recently, airborne LiDAR has been used to collect bathymetric data in shallower areas, with these systems being able to cover much larger spatial areas and to go to areas previously inaccessible by ships (Ohlendorf et al., 2011). LiDAR bathymetry is used in Chapters 3 and 4.

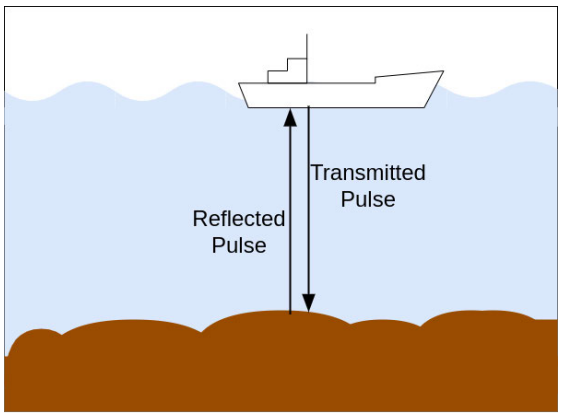
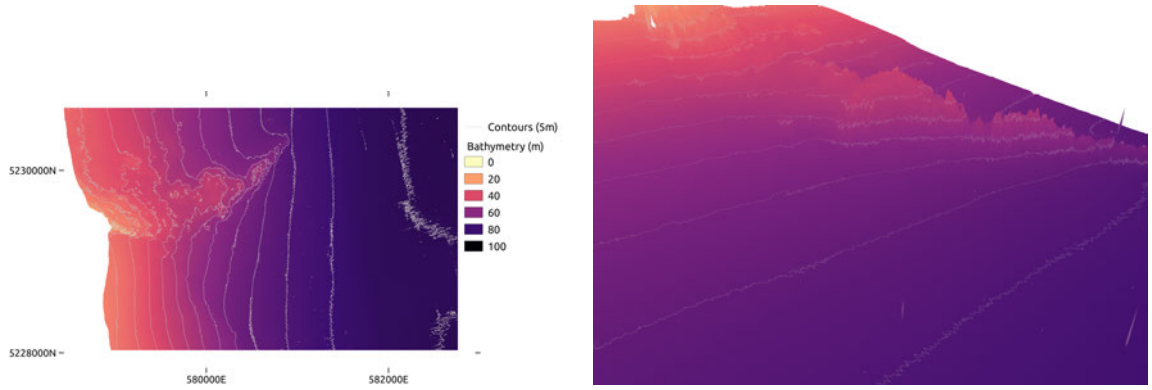


Figure 2.2 – A diagram of a ship collecting depth observations using sonar. The depth observations are combined with the location information to create a bathymetric map.

The collection and distribution of bathymetric data is a major priority for the global scientific community, with programs such as Seabed 2030 Project driving collaboration on worldwide efforts Wölfl et al. (2019). However, the bathymetric data does not contain enough information to properly observe the benthos, and in-situ sampling is needed. Therefore, after the ocean is mapped - the next question becomes what species inhabit the seafloor area and how does this relate to the bathymetric terrain.

2.4 Benthic Imagery and Habitat Classification

In-situ habitat observations can be efficiently obtained by an conducting a benthic AUV survey (Brown et al., 2011a), where downwards facing cameras collect imagery of the seafloor. The combination of accurate geo-referenecing and consistent imaging leads makes for ideal habitat observations. Annotating the large quantity of data is a key challenge for AUV surveys, as an AUV can capture tens of thousands of seafloor



(a) (b)

Figure 2.3 – An example bathymetric map of the O'Hara reef near Fortesque, Tasmania. (a) shows a 2D top-view of the reef, while (b) shows a snapshot from a 3D view of the O'Hara reef ridgeline, as viewed from the South-East. Coordinates are displayed in UTM zone 55S.

images per dive (deployment), which results in hundreds of thousands of seafloor images of the survey area. Manually labelling these images is infeasible and therefore machine learning approaches are utilised to annotate the images.

The feature extraction and clustering pipeline developed by Steinberg (2012) can be utilised to assign habitat labels with minimal supervision. Image features are extracted using the Sparse code Spatial Pyramid Matching (ScSPM) (Yang et al., 2009), which are then clustered using the Grouped Mixtures Clustering (GMC) method (Steinberg, 2012). After clustering is completed, the clusters are further grouped together to form habitat classes, based on examples from each cluster. This approach has been utilised for previous benthic habitat mapping approaches (Bender, 2013; Rao et al., 2017; Shields et al., 2020).

The unsupervised clustering then cluster assignment process is effective at generating habitat labels with minimal supervision, however as the cluster formation is completely unsupervised the clusters formed may not be homogeneous, that is multiple different habitat classes can occur within the same cluster. To solve this problem, Yamada et al. (2022) proposes to use minimal supervision where several candidate images from each class are selected to guide the supervision process. As a small number of images are needed to be labelled (<100), this process still only requires little

human intervention. The advantage of this process is the class boundaries are now guided by the habitat labels. To allow this process to work with small dataset sizes, Yamada et al. (2022) uses an autoencoder to compress the images into an informative feature space. A novel approach of this approach is the inclusion of contextual data from the images, including the images depth and location, with the proposition that images that are similar in location or at similar depths are more likely to be of the same habitat label. Labels for the Fortesque region have been calculated, with an 80% accuracy for the 12 classes of: Ecklonia, High Relief Reef, Low Relief, Reef, Patch Reef, Reef/Sand Ecotone, Screw-Shell Rubble, Screw Shell Rubble/Sand, Coarse Sand, Sand. This class resolution is too fine-scale to predict with bathymetry and hence these classes have been amalgamated into a 4 class dataset and a 6 class dataset, that will be the primary datasets used for habitat modelling in this chapter. The 4 class dataset has following classes: sand, screw-shell rubble, reef and kelp. The 6 class dataset has the following classes; sand, screw-shell rubble, patchy reef, low-relief reef, high-relief reef and kelp. Examples of these are displayed in Figures 2.4 and 2.5.

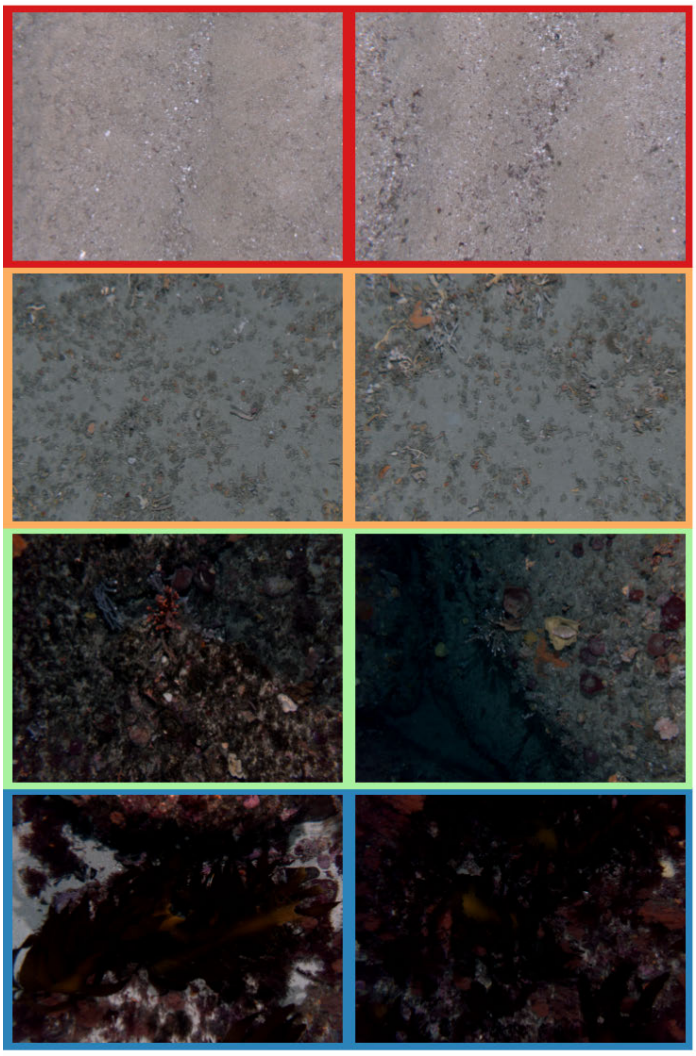


Figure 2.4 – Example habitat labels generated by Yamada et al. (2022) showing four benthic categories; sand (red), screw-shell rubble (orange), reef (green) and kelp (blue).

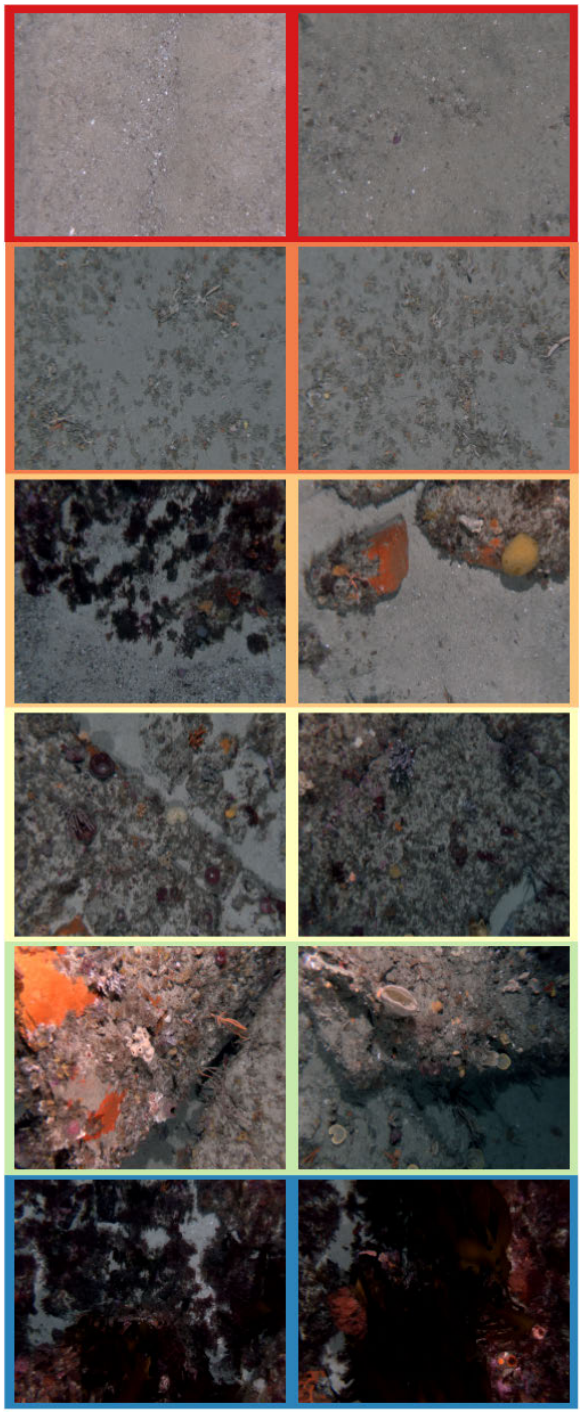


Figure 2.5 – Example habitat labels generated by Yamada et al. (2022) showing four benthic categories; sand (red), screw-shell rubble (dark orange), patchy reef (light orange), low-relief reef (yellow), high-relief reef (green) and kelp (blue).

2.5 Data

This chapter focuses on habitat modelling using AUV surveys conducted around the Fortesque region of Tasmania in 2008. This region was selected due to the high resolution bathymetry, interesting habitat composition and availability of previous research into this area. The bathymetry was collected by GeoScience Australia (Spinoccia, 2011) and is gridded at 2m. The AUV surveys were conducted by the ACFR using AUV Sirius. Figure 2.6 displays a bathymetric map of the Fortesque area, with each AUV survey. These dives are spread out over a large spatial extent. The class composition of each dive is presented in Table 2.1, showing that most dives in these areas visit most of the classes present.

Dive	Sand	Screw-shell Rubble	Patchy	Low Relief Reef	High Relief Reef	Kelp
Waterfall_05	3831	893	771	1091	2395	2375
Waterfall_06	1230	352	560	1978	811	1118
O'Hara_07	1314	2524	1210	1665	2382	1351
Patch Reef North_08	616	2542	541	247	1763	1
HippoN_09	415	3410	444	261	904	2
ChevronRockN_10	301	3388	500	463	862	1
LittlehippoN_11	993	1728	1340	1165	1120	23
LittlehippoSE_12	460	1061	815	1031	857	580
HippoS_13	349	2175	435	611	1080	33
ChevronRockS_14	166	1146	393	442	1327	1845
Blowhole_15	1035	632	2118	2434	174	4160
Sisters_16	2156	4293	2190	1147	4173	41
High_yellow_19	2599	3131	2650	5514	204	1064
O'Hara_20	925	1416	351	816	1667	748

Table 2.1 – Class counts for each dive in the Fortesque region using the 6 class labelling scheme from (Yamada et al., 2022).

This chapter focuses on a small subsection of the total Fortesque area, the reefs around O'Hara and Waterfall. These AUV surveys conducted around this area (*Waterfall05*, *Waterfall06*, *O'Hara07* and *O'Hara20*) have similar class compositions and as they are nearby geographically, the relationship between bathymetry and habitat class should be similar. This is important, as when validating different approaches to habitat

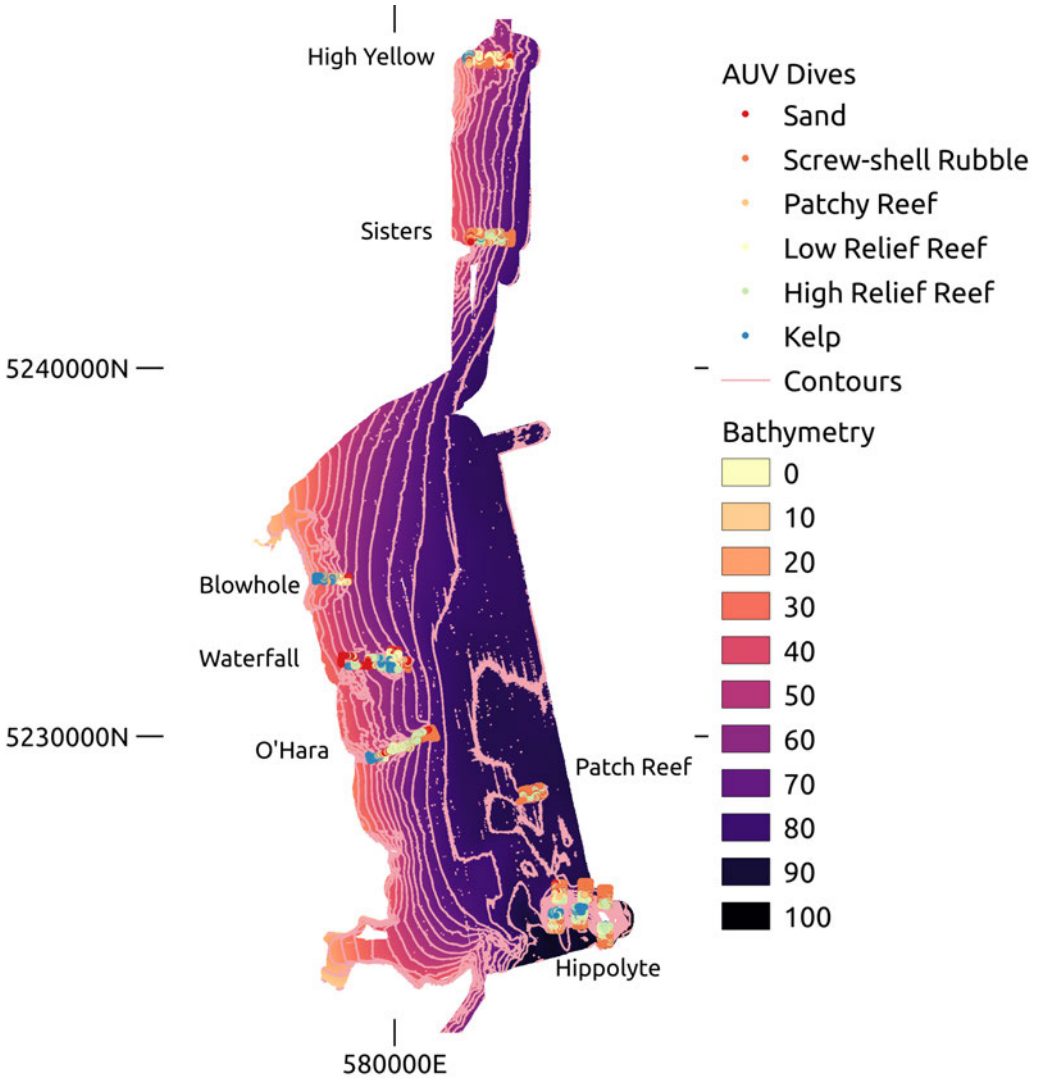


Figure 2.6 – AUV surveys conducted in 2008 by the AUV Sirius around the Fortesque region of Tasmania. The image locations are labelled according to the habitat label. Coordinates are displayed in UTM zone 55S.

modelling the data partitions (Train, Val, Test) should be of the same distribution. Figure 2.7a displays the four dives in the O'Hara / Waterfall region. Figures 2.7b and 2.7c display the same dives, with the colours now corresponding to the classes. The shallower, reef areas are dominated by kelp which only grows in shallow areas due to the need for sunlight. The deeper reef areas form the Patchy Reef, Low Relief Reef and High Relief Reef habitats. The flatter areas are generally the Patchy Reef, Sand and Screw-shell rubble classes. The screw-shell rubble class tends to occur in

deeper areas than the sand class.

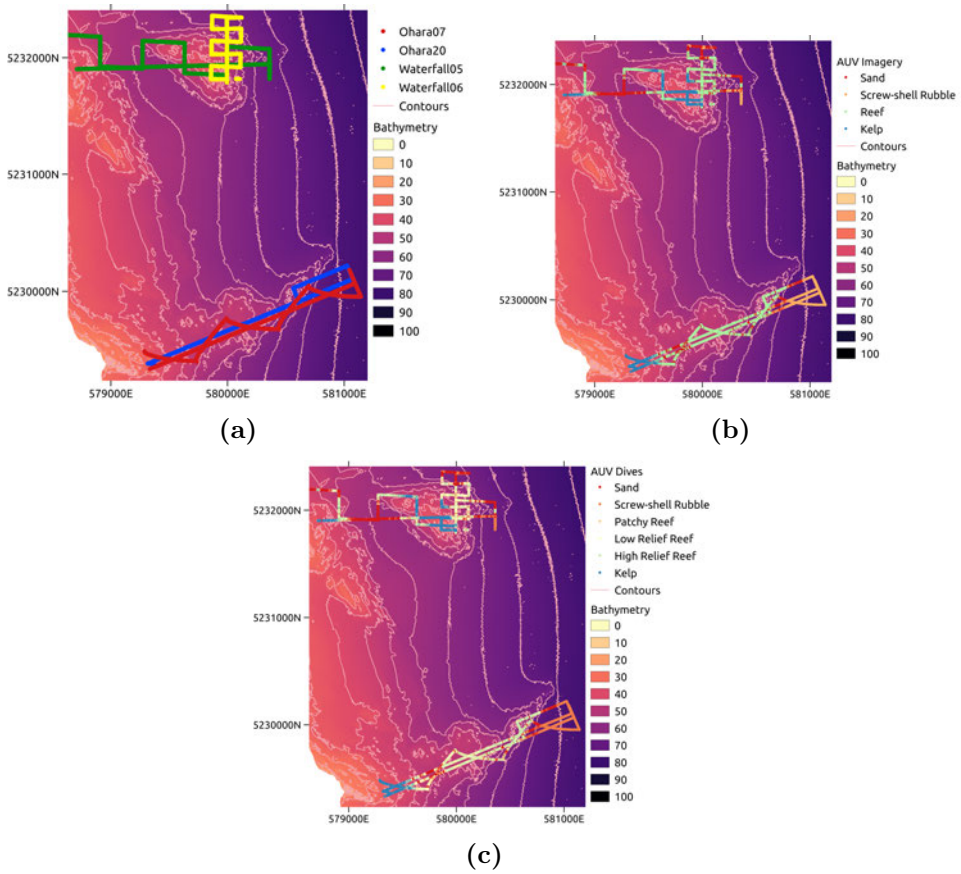


Figure 2.7 – The area of focus around the O'Hara / Waterfall reefs, a subsection of the Fortesque region. (a) colour codes each individual survey. (b) shows the habitat labels using 4 habitat classes. (c) shows the habitat labels for 6 habitat classes. Coordinates are displayed in UTM zone 55S.

2.5.1 Partition Filtering

Each dive is assigned to either training or validation / testing. This method of data partioning is preferred over partitioning every sample as it allows easier separation of training and testing data. Samples located close to each other have the same or very similar bathymetric feature space, and therefore should not be in different dataset partitions or else it would be contaminated. For the O'Hara / Waterfall region, two dives are allocated for training and two for validation / testing. The validation dataset is 200 samples randomly extracted from the test dataset. As displayed in Figure 2.7a,

the dives in the same area overlap one another, which would lead to contamination of the testing datasets. To avoid contamination, the testing dataset is filtered so that every sample in the dataset is not near any point in the training dataset. Figure 2.8a focusses on the O'Hara region, showing the two dives *O'Hara 07* and *O'Hara 20*. As shown, there are several overlaps between the dives. Figure 2.8b zooms in on the area furthest east of the dives, where there are several areas of overlap. Figure 2.8c shows this same area, after the location filtering has been applied to the *O'Hara 20* dataset. The areas of the *O'Hara 20* dataset (in blue) have been removed around the intersection points, as has the portion of the dive where the two dives ran parallel.

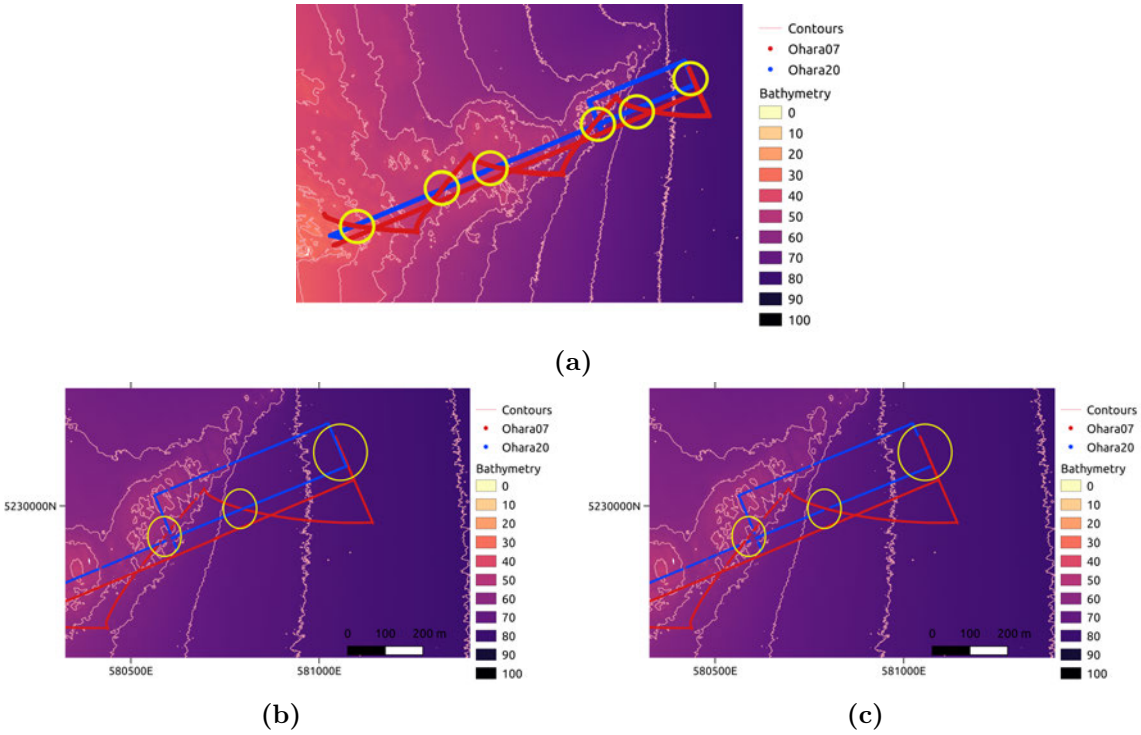


Figure 2.8 – Data filtering to avoid contamination of the train and testing splits. (a) shows the surveys on the O'Hara reef, circling the areas of proximity between the two surveys. (b) zooms in on the area to the East of the surveys, showing the areas of proximity before filtering has been applied. (c) shows the same area, after filtering has been applied, noting that labels have been removed from the blue dive when in proximity of the red path. Coordinates are displayed in UTM zone 55S.

2.6 Feature Extraction

The bathymetry is a depth map of the underwater environment, but the depth alone at a given location only provides limited information about the benthic habitat; it is the structure of the terrain around this location that provides greater insight into the benthic habitat. To summarise this terrain into a set of features that can be related to the benthic habitat, two methods are explored; geometric features and encoded features. The process for extracting each of these feature sets are described in the following sections (2.6.1, 2.6.2).

The introduction of 'deep learning' resulted in a shift-away from feature extraction and classification being separate processes. With the advent of deep neural networks, the raw input data (e.g. images) is fed directly into the network whose output is the classification (for a classification network) (Krizhevsky et al., 2012). The advantage of this method is that abstract but effective features can be learnt that directly relate to the output. However this approach requires large data quantities to train the networks. Due to the difficulty in collecting and labelling AUV data, methods that rely on large quantities of data should be avoided. Furthermore, having a small classification network is useful for developing the probabilistic models presented in Section 2.8. Hence, this work adopts the paradigm of separating the feature extraction and classification processes.

2.6.1 Geometric Features

With the bathymetry raster as input, we compose a geometric feature vector composed of the following attributes; depth, slope, aspect and ruggedness. These features are selected as they represent the structure of the seafloor, which is related to the benthic habitat (Wilson et al., 2007). The depth is simply the depth at the given location. The slope and aspect are calculated using Horn's algorithm (Horn, 1981), that uses the surrounding 8 raster cells to calculate the gradient, with each cell weighted according to its distance to the centre pixel. For ruggedness, the surrounding 8 cells.

The Terrain Ruggedness Index (TRI) (Wilson et al., 2007) is used to estimate the ruggedness, which is the mean difference between a cell and its surrounding cells.

Using Horn's method, the slope in the x-direction (p_x) and y-direction (p_y) can be calculated as:

$$p_x = \frac{(z_{-+} + 2z_{-0} + z_{--}) - (z_{++} + 2z_{+0} + z_{+-})}{8d},$$

$$p_y = \frac{(z_{--} + 2z_{-0} + z_{-+}) - (z_{++} + 2z_{+0} + z_{+-})}{8d},$$

where z_{00} corresponds to the centre cell, z_{-+} corresponds to the top left cell and z_{+-} corresponds to the top right cell in a 9-square grid. d is size of each raster cell. Following this, the magnitude of the slope (s) and aspect (a) can be calculated as:

$$s = \sqrt{p_x^2 + p_y^2}, \tag{2.1}$$

$$a = \arctan \frac{p_x}{p_y}. \tag{2.2}$$

The aspect is decomposed into the northing and easting components to avoid the angle discontinuity,

$$a_N = \cos a \tag{2.3}$$

$$a_E = \sin a \tag{2.4}$$

Using this notation, the ruggedness as calculated by TRI (r_{TRI}) can be written as:

$$r_{TRI} = \frac{1}{8} \sum_{i \in \{-,0,+\}} \sum_{j \in \{-,0,+\}} z_{00} - z_{ij} \tag{2.5}$$

The final feature vector at a given location is composed as follows:

$$\mathbf{z}_{\text{geometric}} = \begin{pmatrix} d & s & a_N & a_E & r_{TRI} \end{pmatrix}, \tag{2.6}$$

where d is the depth, s is the slope, a_N and a_E are the aspects in the northerly and easterly directions respectively and r_{TRI} is the ruggedness.

2.6.2 Learnt Features

Handcrafted features, while intuitive, can be too coarse to represent some benthic habitats, not necessarily representing the bathymetry efficiently and comprehensively. Learning a feature representation is an alternative that can develop a richer and more expressive representation. The benefits of a learnt feature representation are part of the successes of deep learning, where the feature extraction and classification modules are jointly optimised as part of a larger model. The subsequent convolutional layers together transform the input image into a complex but informative feature space. The convolutional layers adapt to optimise the feature space for the given task, for example creating features that are more able to discriminate between classes in a classification task.

While using supervised learning to approach this task is possible, it also is inefficient. The bathymetry is plentiful, but in-situ habitat observations are sparse. For this case, self-supervised learning can be used to develop a rich feature space, without the need for external observations. An autoencoder is one self-supervised method that can be applied to this problem.

2.6.3 Autoencoder for Bathymetric Feature Extraction

There are two components to an autoencoder; the encoder compresses the input into a latent space, whereas the decoder uses this latent space representation as its input and attempts to reconstruct the original input. An autoencoder is trained to minimise a reconstruction loss such as the MSE loss function:

$$L_{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \hat{\mathbf{x}}_i)^2, \tag{2.7}$$

where x_i is the input pixel at index i and $\hat{x}_i$ is the corresponding reconstructed pixel.

Convolutional neural networks are capable of extracting rich-features where there are local patterns to the data (Lecun et al., 2015). This makes them effective for bathymetric data. In this document, the autoencoder is trained on patches randomly

sampled from the entire bathymetric area. The bathymetric patches used as input to the encoder are centred around a given coordinate and are typically a square of width 11 or 21 pixels. The encoder consists for two convolutional layers with 512 filters in each, followed by one fully connected layer of 256 neurons, with a latent dimension of 16 (32 when using 21×21 as the input patch size). The decoder is the reverse of the encoder. This model is illustrated in Figure 2.9.

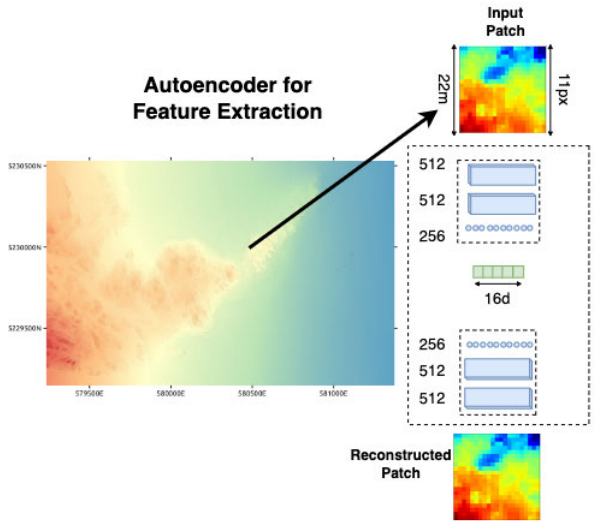


Figure 2.9 – The bathymetric autoencoder. Patches of bathymetry extracted from the raster are compressed into a feature space by the encoder. The decoder then aims to reconstruct the original patch using the latent space as input.

The learnt feature vector appends the mean depth value ($\bar{d}$ to the latent space of the autoencoder ($\mathbf{z}_{latent}$):

$$\mathbf{z}_{encoded} = \begin{pmatrix} \mathbf{z}_{latent} & \bar{d} \end{pmatrix} \tag{2.8}$$

The training of the autoencoder is a self-supervised process, which allows the encoder to become an effective feature extractor without the use of habitat labels. From the entire extent of the bathymetry, a geodetic coordinate (latitude, longitude) is selected. At this location, a bathymetric patch of a given size is extracted, centring on this point. The mean patch is mean-centred, where the mean value is subtracted from each value of the patch. This allows the autoencoder to focus on reconstructing the terrain of the patch. The loss function is minimised using the Adam optimiser

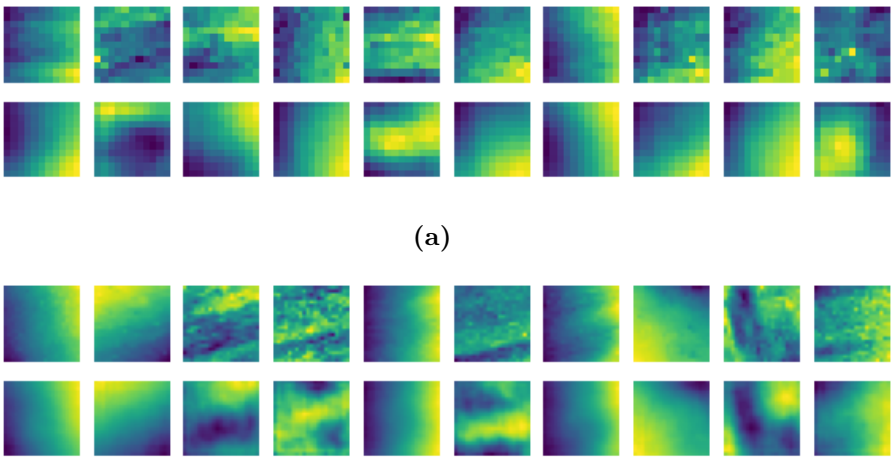
(Kingma and Ba, 2014).

When training the autoencoder on large areas that are dominated by barren habitats, the autoencoder will seemingly perform well, with a low reconstruction error. However, the autoencoder will primarily be very efficient at reconstructing flat terrain but will not be able to reconstruct rugose or varied terrain with the required accuracy. *Focus training*, which involves only back-propagating when the loss is high, can be used to increase the performance of the autoencoder in reconstructing a variety of terrain types. The improved performance when focus training is displayed in Figure 2.12.

2.6.4 Results

The autoencoder is an effective way to compress the features into the latent space, if the entire bathymetric patch can be instilled into the latent space of the autoencoder. If the decoder can recreate the input with reasonable accuracy, the feature space is a compressed version of the input space. The accuracy of the reconstruction is limited by the size of the latent space, with latent spaces that have too few dimensions making it infeasible to reconstruct the patch. As the objective of this feature extraction method is to generate an expressive but compact feature representation, the number of latent dimensions is set to a minimum while still being able to generate usable feature spaces. Figure 2.10 shows example reconstructions for both patch sizes of 11 and 21 for the Fortesque bathymetry.

Results are reported for autoencoders trained on a large area of bathymetry with a horizontal resolution of 2m, located off the coast of Fortesque, Tasmania, as collected by GeoScience Australia (Spinoccia, 2011). Two different patch sizes are considered, 11x11 and 21x21. For a patch size of 11x11, the latent space is 16 dimensions, whereas for the patch size of 21x21 the latent space is 32 dimensions. The latent dimensions were chosen empirically, with the objective being to minimise the dimensions of the latent space while still achieving good reconstruction errors. Table 2.2 provides the mean, maximum and minimum mean reconstruction errors for the au-



(a)

(b)

Figure 2.10 – Example reconstructions for patch sizes of 11 by 11 (a) and 21 by 21 (b). The top row of each figure is the original patch, while the bottom row is the reconstructed patch. Both show the reconstructed feature spaces having a close similarity with the original

toencoders. Flat, barren expanses dominate the Fortesque bathymetry and are easy for the autoencoder to replicate and skew the mean towards zero for this area. To complement the results for the entire Fortesque region, results are also presented for two smaller areas (O'Hara and Waterfall) that are partially covered by more rugose reef, that is harder to reconstruct. For both the patch sizes and across all three areas, the autoencoders can accurately reconstruct the input patches. The 'maximum' metric for the entire bathymetry is impacted by outliers. On the reefs of interest, there are small outliers and the maximum reconstruction error is very low. Figures 2.11a and 2.11b show box-and-whisker plots for the two different patch sizes. These demonstrate the accuracy of the reconstructions, with the entire box-and-whisker plot falling below 0.1 and 0.2 MSE.

Focus training, where the autoencoder is only trained on examples that have a high loss, offers a significant performance improvement. This is displayed in Figure 2.12, where the focus training (in orange) results in a reduction in the reconstruction error. As the bathymetry is unbalanced, with barren areas forming much of the area and harder to reconstruct errors forming a smaller area, focus training ensures the areas

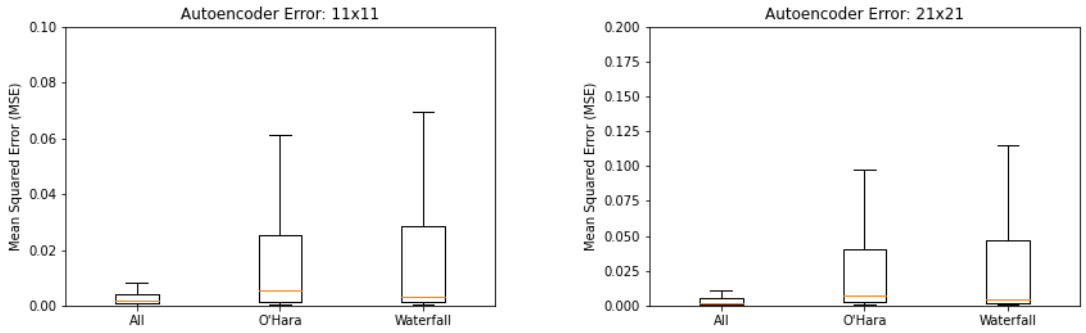
Patch Size	Area	Mean	Max	Min
11x11	All	0.02	14.7	0.0001
	O'Hara	0.03	1.6	0.0004
	Waterfall	0.03	2.1	0.0003
21x21	All	0.02	10.2	0.0002
	O'Hara	0.04	1.1	0.0005
	Waterfall	0.04	1.5	0.0005

Table 2.2 – Reconstruction error (Mean-Squared Error) for autoencoders trained on the Fortesque bathymetry. Although the max values do contain significant outliers, the overall reconstruction error is very low demonstrating the autoencoders are sufficiently compressing the bathymetric patches.

that are harder to reconstruct are prioritised.

Spatially mapping the reconstruction error provides a demonstration of how the autoencoder is able to successfully reconstruct input patches over the entire area. Figure 2.13 shows the reconstruction error for the entire Fortesque area, while Figure 2.14 shows the reconstruction error over the O'Hara region. These maps also demonstrate the improved performance when using focus training, with a noticeable difference on the O'Hara reef.

The autoencoder learns to distil the information in the bathymetric patch into the feature space, which can then be viewed as a feature set that completely describes the bathymetric patch. This is an effective method for developing an expressive feature space and a viable alternative to geometric features. Comparisons between this encoded feature space and geometric features for clustering can be found in Section 2.7.4 and habitat modelling in Section 2.10.4.



(a) **(b)**

Figure 2.11 – Box-and-whisker plot showing the reconstruction error for the autoencoder using 11×11 (a) and 21×21 (b) input patches. These plots demonstrate that the autoencoder is able to reconstruct the input patches with a very low mean-squared error.

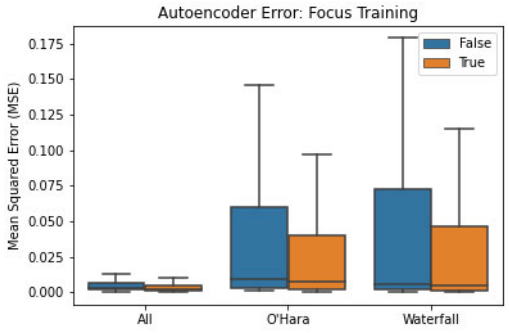


Figure 2.12 – Box-and-whisker plot highlighting the performance improvement when using focus training, where only patches with high loss are trained on. These results are with patch size of 21×21, where 'True' indicates focus training has been used, while 'False' indicates it has not.

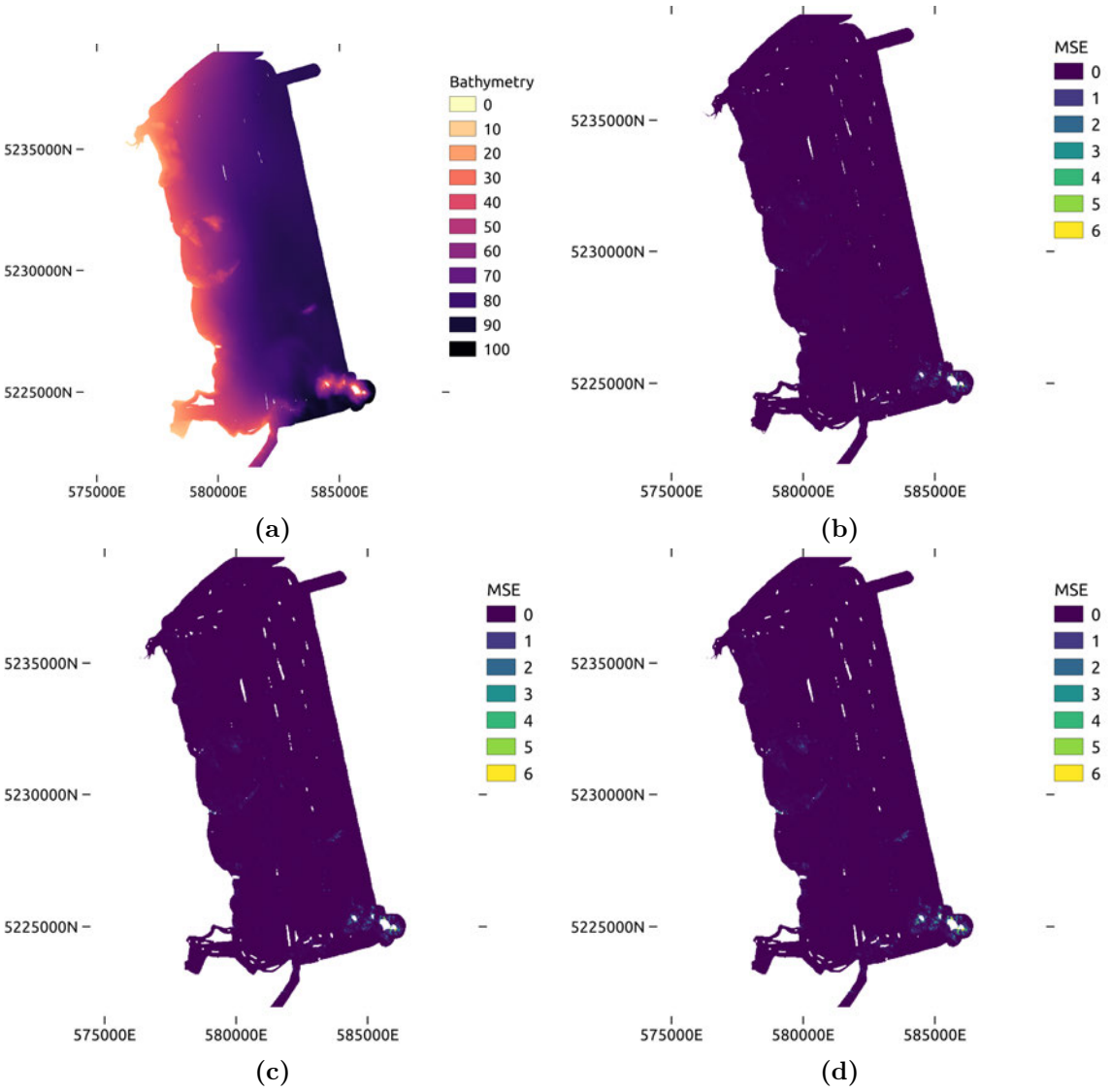


Figure 2.13 – Mapping the Mean-Squared Error (MSE) for the entire Fortesque area. (a) shows the original bathymetry. (b) shows the reconstruction error for the autoencoder using a 11×11 patch. (c) shows the reconstruction error for the autoencoder using a 21×21 patch, with focus training applied. (d) shows the reconstruction error for the autoencoder using a 21×21 patch, which has not had focus training applied. Coordinates are displayed in UTM zone 55S.

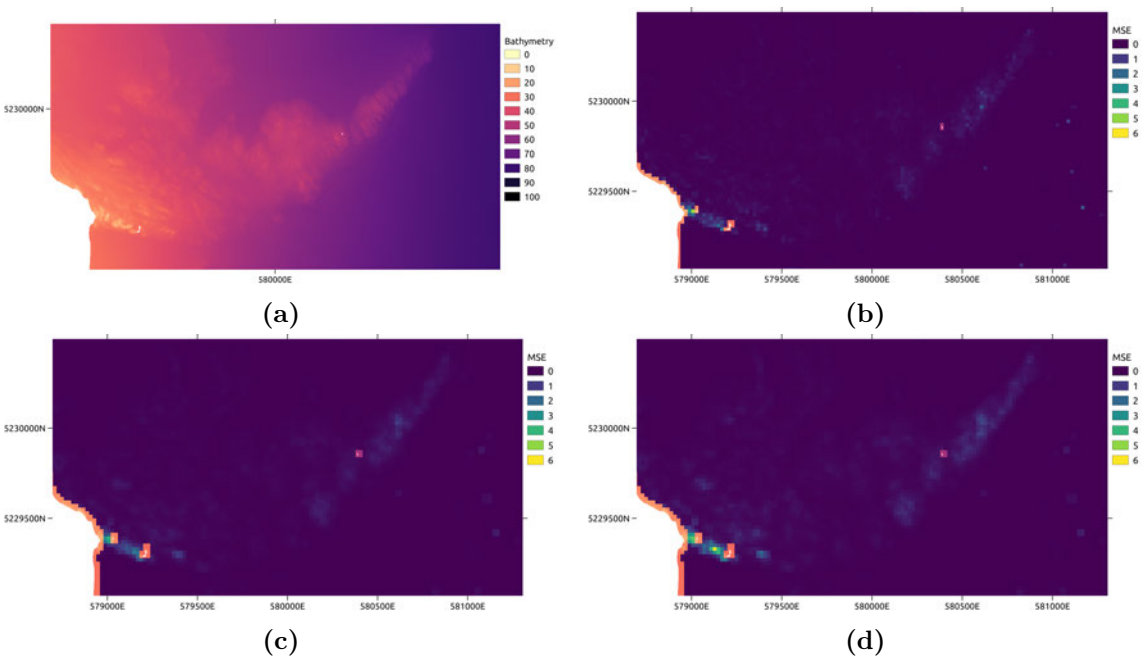


Figure 2.14 – Mapping the mean reconstruction error around the O'Hara reef. (a) shows the original bathymetry. (b) shows the reconstruction error for the autoencoder using a 11×11 patch. (c) shows the reconstruction error for the autoencoder using a 21×21 patch, with focus training applied. (d) shows the reconstruction error for the autoencoder using a 21×21 patch, which has not had focus training applied. When comparing (c) and (d) it becomes clear that focus training improves the reconstruction error. Coordinates are displayed in UTM zone 55S.

2.6.5 Variational Autoencoder

A Variational Autoencoder (VAE) conditions the latent space by imposing a multivariate Gaussian distribution on the latent space. This is trained using the Evidence Lower Bound (ELBO) loss function, as displayed in the following equation. The left term is equal to the mean squared error, while the right term is the Kullback-Leibler (KL) divergence term between the predicted distribution and a standard normal distribution (Kingma and Welling, 2019).

$$L_{ELBO} = \mathbb{E}_{q(z|x)}[\ln p(x|z)] - D_{KL}[q(z|x)||p(z)] \tag{2.9}$$

The conditioning of the latent space leads to a smooth and continuous feature space. This property is leveraged in Chapter 3 for exploration of the feature space.

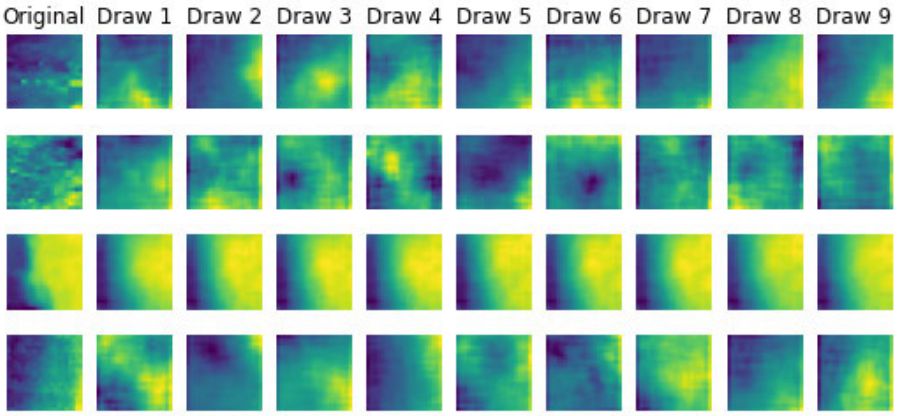


Figure 2.15 – Multiple draws from the latent space of a VAE. The first column shows the input patch, while the subsequent columns are the reconstructed patches from each draw from the VAE. The draws are all similar to the input patch, while also being different in individual ways.

2.7 Bathymetric Clustering

Clustering is a method that groups together similar areas of the feature space, which can provide insights into the benthic habitats in each of these groups. Plotting the

clusters spatially can identify areas of similar bathymetric terrain, that can be a useful tool in visualising the survey area and planning initial benthic surveys. When the relationship between bathymetric terrain and habitats is strong, the bathymetric clusters can align with habitat observations.

2.7.1 Clustering using Gaussian Mixture Models

Forming meaningful insights from data without supervision is difficult, and hence there are many approaches to unsupervised clustering and it is still an active area of research. A common issue is the unsupervised groupings not forming in similar ways to human intuition. As clustering relies on distance measures between feature vectors that degrade in higher dimensions, using a feature representation that is both expressive and compact is essential. While this is difficult for high dimensional data (such as imagery), it is achievable for bathymetric patches that have significantly fewer dimensions after feature extraction is performed.

The various clustering methods use the distance between features to group together similar methods. A simple approach to clustering is the K-Means algorithms, which positions K means (virtual data points) in the feature space such that the feature space is divided into Voronoi cells. A data point's cluster is the nearest mean point is nearest to in feature space. A significant limitation of this approach is that it only create clusters of circular shape (in 2 dimensions, spherical in 3 dimensions) and will not create intuitive clusters if the data doesn't match this representation. An alternative is Gaussian Mixture Models (GMM), that model the feature space as a weighted sum of k Gaussians. It is trained using the Expectation-Maximisation algorithm. This enables the clusters to fit non-circular data. The cluster of a data point for GMMs is the Gaussian that accounts for largest density at the given location. For both the K-Means and GMM methods, the number of clusters needs to be specified. Other methods, such as DB-Scan and VDP (Variational Dirichlet Processes) (Kim et al., 2006) use automatic methods for selecting the number of clusters. For this application, setting the number of clusters is an intuitive way of setting the resolution

of the clustering. For example, setting the number of clusters to 10 is easily interpretable visually. Furthermore, it can be intuitively set to approximately align with the number of expected types of benthic habitats. As the GMM clustering method is robust and versatile it is used for clustering the bathymetric feature space.

2.7.2 Clustering on the O'Hara Waterfall region

The O'Hara and Waterfall areas of the Fortesque area consist of two primary reef structures - O'Hara and Waterfall, surrounded by relatively barren sand/screw-shell rubble classes. The labelled AUV imagery from surveys conducted in 2008 over these areas show the habitat classes in these regions. The AUV surveys mostly examine these reef structures, but also extend beyond the reefs. Figure 2.16b shows clustering with GMM and 10 clusters. Clustering with 10 clusters allows the map to be easily interpretable. By comparing the map to the labelled AUV imagery in Figure 2.16a, the clusters show some correspondence to the habitat labels. Clusters 0, 1, 7 overlap with the sand and screw-shell classes. Cluster 9 is more prominent around interface areas, that are likely patchy reef. The remaining clusters overlap with the reef and kelp classes. The clustering has also identified artefacts in the bathymetry, known as track lines, where the multibeam has not been processed correctly. As depicted in Figures 2.16c and 2.16d, as the number of clusters increases, the map becomes less interpretable. The combination of the excessive number of clusters and the smaller spatial region for each cluster make it harder to interpret. When clustering on the bathymetry, the number of clusters should be set such that it is easily interpretable while also being able to distinguish major habitat groups.

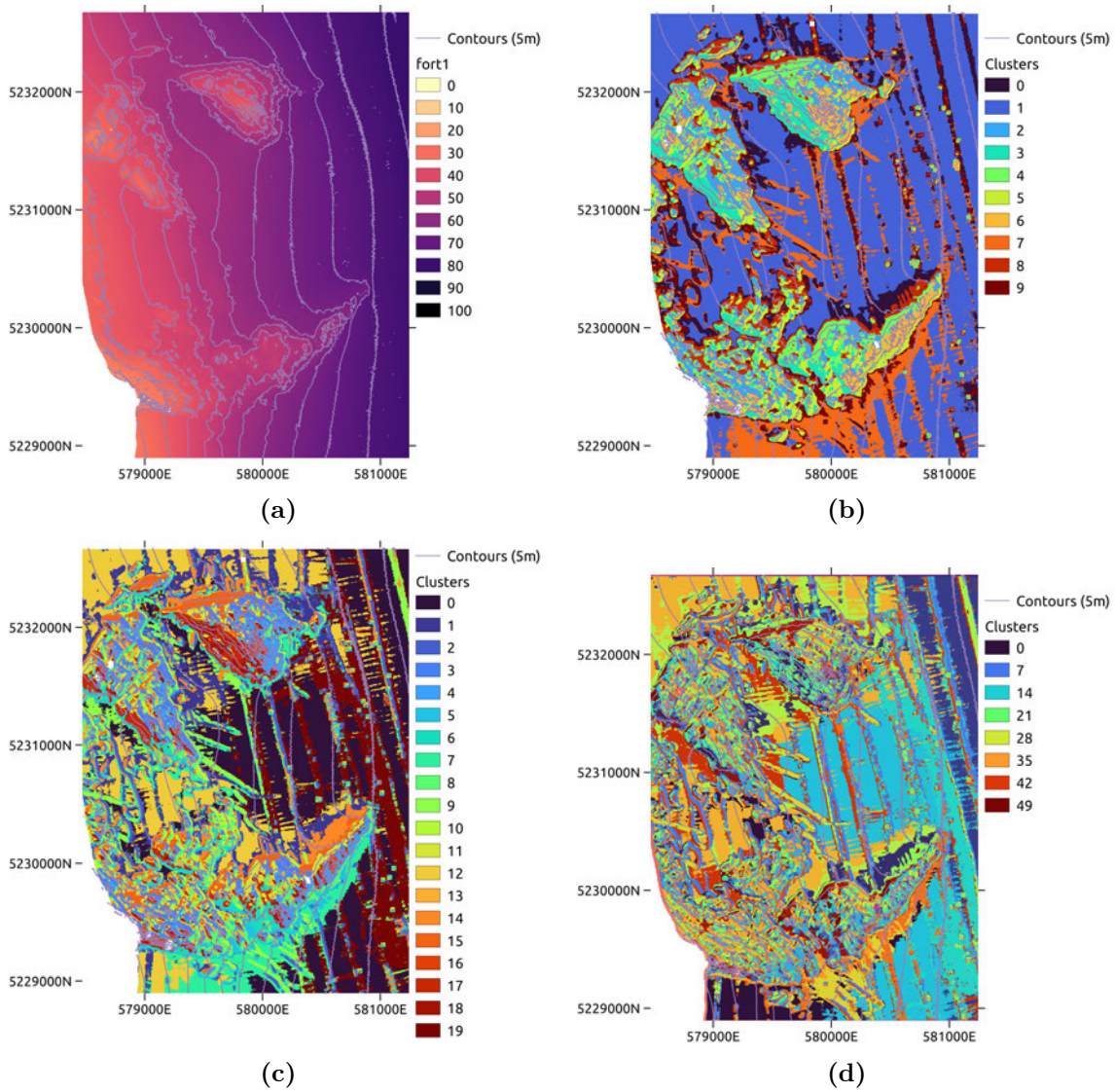
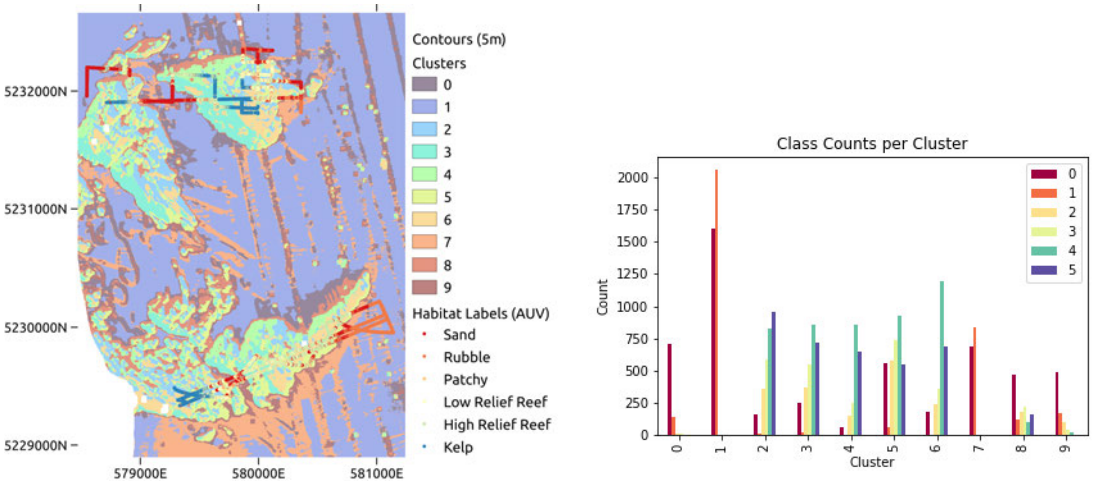


Figure 2.16 – Clustering with GMMs the bathymetric terrain of the O'Hara / Waterfall areas. (a) shows the bathymetry, (b) shows clustering with 10 clusters, (b)with 20 clusters and (d) with 50 clusters. Coordinates are displayed in UTM zone 55S.

2.7.3 Cluster Assignment

The bathymetric clusters can be assigned a habitat class to create a habitat map. The label of each class is selected by looking at the occurrence of each habitat classes (from a labelled AUV survey), in each bathymetric cluster. As shown in Figure 2.17a there is a correlation between the unsupervised clusters and the habitat observations. This is particularly evident for cluster 1 and the screw-shell rubble habitat. The number

of observations of each class in each cluster is presented in Figure 2.17b. For some clusters, including cluster 0 and cluster 9, the modal class in the cluster accounts for the vast majority of classes in that cluster. These are clusters that have a strong correlation with the underlying benthic habitat. In most of the others, there is a mixture of classes that occur in each cluster. The potential accuracy of this method is limited as the bathymetry is low resolution and the habitat class varies at a much higher rate. For the Tasmania dataset, a bathymetric cell is 2m and the bathymetric patches that are used for feature extraction are 11 by 11 cells, making the patch size 22 by 22 meters. There can be multiple classes of benthic habitat within a single bathymetric cell, and many more within a bathymetric patch. Although assigning labels to the bathymetric clusters is not expected to be accurate, it is a method that can be used as a benchmark to compare other habitat modelling methods to. It is also predicted to generate habitat maps with few habitat labels.



(a) **(b)**

Figure 2.17 – The overlap between bathymetric clusters and habitat labels. (a) shows the habitat labels overlaid onto the bathymetric clusters. (b) displays the counts of each habitat class, for each cluster. There is a definite correlation between clusters and habitat classes, but the clusters are from homogeneous. Coordinates are displayed in UTM zone 55S.

The clusters and the benthic habitat regions may not align, which is more likely when there are few clusters. Therefore when assigning clusters, increasing the number of clusters significantly (e.g. to 50) makes it more likely that the cluster boundaries will

correspond with benthic habitat boundaries. Figure 2.17 shows the O'Hara/Waterfall region clustered with 50 clusters and then the resulting habitat maps using the cluster assignment process. The *O'Hara 07* and *Waterfall 05* dives are used for training while the *O'Hara 20* and *Waterfall 06* validation (see Section 2.5 for more information). Both methods develop a habitat map that generally appears visually correct, the reef areas are correctly classified into either kelp or reef clusters while the flatter areas are mostly classified into sand or screw-shell rubble. On closer inspection, the clustered regions are too large to provide accurate habitat labels. For the 4 class dataset, the accuracy is 63%, while for the 6 class dataset the accuracy is 40%. Figure 2.19 presents confusion matrices to provide further insight into these results. For the 4 class dataset, there is significant confusion between the sand and rubble classes, and also between the reef and kelp classes. This implies that the border between these regions do not share the same border as the clusters. The performance on the 6 class dataset is significantly reduced. The habitat classes that are rarer in this dataset, patchy reef and low relief reef, are not predicted by the cluster model. As these classes occupy a smaller spatial region and there is not a separate cluster that represents them, the clusters they fall into have another modal class. Overall, the discriminatory ability of this method is too weak for classifying into these 6 classes.

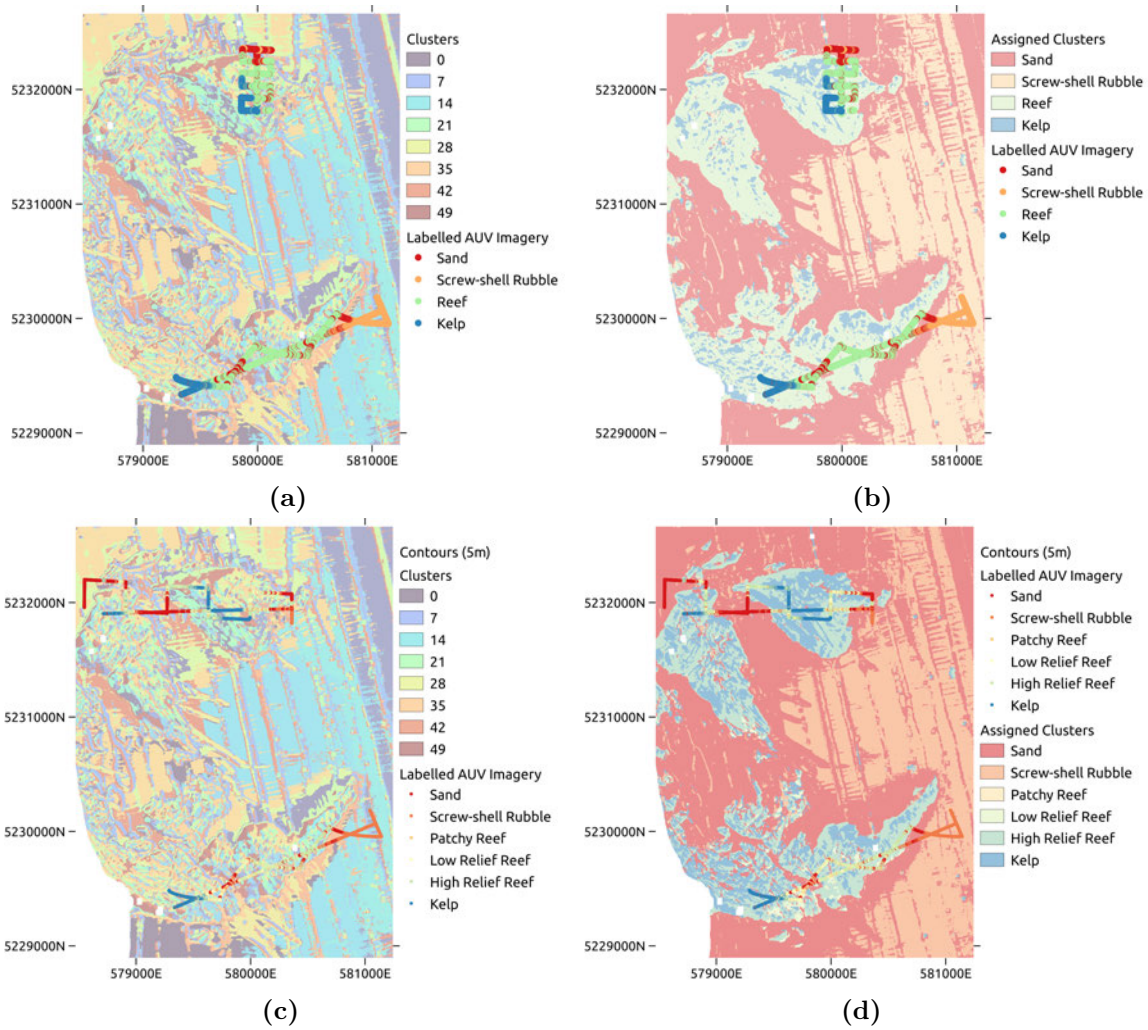
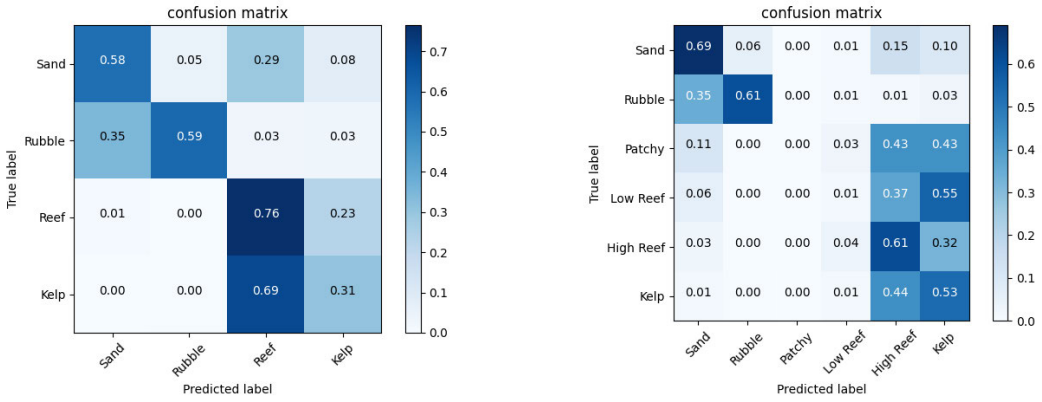


Figure 2.18 – Assigning habitat labels to bathymetric clusters. Top row uses four classes, while the bottom row uses six. This process is more effective with fewer classes. When there are more classes, classes with fewer samples are underrepresented in the final map. Coordinates are displayed in UTM zone 55S.



(a) **(b)**

Figure 2.19 – Confusion matrix when assigning clusters for 4 classes (a) and 6 classes (b). When used for habitat mapping, this process has low accuracy and could be improved by using more advanced machine learning methods.

2.7.4 Geometric versus Encoded features

The previously presented results used encoded features to cluster, however it is also possible to use geometric features (as presented in Section 2.6.1). Figure 2.20 displays clustering with encoded features and geometric features. It is evident that clustering with encoded features creates a more consistent, accurate and easy-to-interpret map. This is due to the improved information available in the encoded feature space. When using the cluster assignment method (6 classes) on the geometric feature space, the accuracy is 24% compared to 40% for the encoded feature space. This highlights the benefits of using autoencoders for feature extraction. This is further explored in Section 2.10.4.

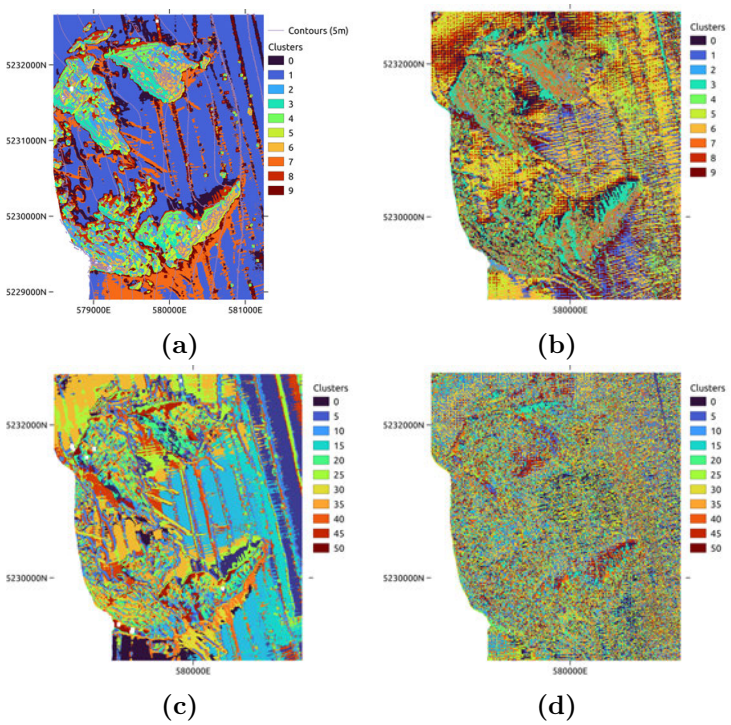


Figure 2.20 – Bathymetric clustering with encoded and geometric features. The left column shows clustering with encoded features, while the right column shows clustering with geometric features. The top row uses 10 clusters, while the bottom row uses 50 clusters. Geometric clustering results in a noisy and difficult to interpret cluster map. Clustering with encoded features creates an accurate and consistent map. Coordinates are displayed in UTM zone 55S.

2.8 Probabilistic Neural Networks

Neural networks have become a primary focus of machine learning research and applications, driven by ability to learn complex relationships between the input data and the outputs. A primary limitation to neural networks is that they operate as a black box and it is difficult to deconstruct what they have learnt. Several methods exist to try and understand the processes however it is largely an unsolved problem. This is a significant barrier to the application of neural networks, particularly in safety-critical applications. Furthermore, a neural network does not provide an uncertainty estimate associated with the predictions. For classification networks with a softmax output, the network's confidence in its prediction is misleading and these networks are typically overconfident in the predictions. For the case of out-of-distribution data, the model has no capability to predict this. Developing probabilistic neural networks is an active area of research. For this habitat modelling application, we investigate two approaches to probabilistic neural networks; Bayesian Neural Networks (BNNs) trained with Stochastic Variational Inference (SVI) (Blundell et al., 2015) and the BNN approximation Monte Carlo Dropout (Gal and Ghahramani, 2016).

2.8.1 SVI

A BNN, as popularised by Neal (1996), is a feedforward neural network, where a distribution is placed over each of the parameters (weights and biases) of the network. BNNs have two significant advantages over GPs; they explicitly handle multi-class / multi-output targets and they scale to large datasets.

Exact inference over complex Bayesian graphs (such as a BNN) generally involves intractable integrals in the posterior. To approximate the posterior, approximation methods are used. MCMC (Markov Chain Monte Carlo (Hastings, 1970)) uses sampling to estimate the posterior, however, for complex models sampling is computationally intensive. For large datasets and complex models, Variational Inference (VI) (Jordan et al., 1999) is a viable alternative for approximate inference.

The underlying idea behind VI is that a simplified approximate distribution q with variational parameters θ is used to approximate the actual distribution p (Blei et al., 2017). *Bayes by Backprop* (Blundell et al., 2015) utilises variational inference to adapt backpropagation to train a BNN. For a BNN where the approximate distribution q is placed over the weights of the network $\mathbf{w}$ and subject to training data D (Blundell et al., 2015):

$$q_\theta(\mathbf{w}|\mathcal{D}) \approx p(\mathbf{w}|\mathcal{D}) \tag{2.10}$$

The approximate distribution is calculated by minimising the KL divergence between p and q, which is framed as an optimisation problem. This is commonly known as the expectation lower bound (ELBO) (Blundell et al., 2015):

$$\theta^* = \arg\min_\theta KL[q(\mathbf{w}|\theta)||P(\mathbf{w}|\mathcal{D})] \tag{2.11}$$

Using the expanded form of the KL divergence:

$$\theta^* = \arg\min_\theta \int q(\mathbf{w}|\theta) \log \frac{q(\mathbf{w}|\theta)}{P(w)P(\mathcal{D}|\mathbf{w})} \tag{2.12}$$

Using log laws to separate $P(D|W)$ and then reconstructing the KL divergence:

$$\theta^* = \arg\min_\theta KLq(\mathbf{w}|\theta) - \mathbb{E}_{q(\mathbf{w}|\theta)}[P(\mathcal{D}|\mathbf{w})] \tag{2.13}$$

However the KL divergence is also intractable, as it involves a complex integral. *Bayes by Backprop* uses sampling to approximate the KL divergence. First it approximates the distribution q by learning its parameters and then samples from this approximate distribution q given the data. This is summarised in the optimisation objective (Blundell et al., 2015):

$$\theta^* = \sum_{i=1}^{n} \log q(\mathbf{w}^{(i)}|\theta) - \log\ P(\mathbf{w}^{(i)}) - \log\ P(\mathcal{D}|\mathbf{w}^{(i)}) \tag{2.14}$$

where $w^{(i)}$ represents a Monte Carlo sample drawn from the approximate posterior

$q(w^{(i)}|\theta)$. To provide gradient estimates for these distributions, *Bayes by Backprop* utilises the reparameterization trick from Kingma et al. (2015).

During inference, Monte Carlo sampling is performed to construct the posterior distribution:

$$\hat{p}(y|\mathbf{z}, \mathcal{D}) = \sum_{t=1}^{T} \text{softmax}(f^{\mathbf{w}_t}), \tag{2.15}$$

where $f^{\mathbf{w}_t}$ is the model parameterised with weights $\mathbf{w}_t$.

In this thesis, the BNN was implemented using PyTorch (Pazke et al., 2017) and Pyro (Bingham et al., 2019). It consists of 3 fully connected layers, all with 1024 units, followed by the logits layer. The approximate distribution q is parameterised by Gaussian distributions. Bayesian Neural Networks trained with SVI have known scalability issues (Ovadia et al., 2019), however in this application, the feature extraction is a separate process and therefore the network can be kept small.

2.8.2 MC Dropout

Monte Carlo Dropout (MC Droput) is another approach for approximating a Bayesian Neural Network (BNN). Dropout is the process of randomly setting a given number of outputs from each layer to zero. It is an effective regulariser of neural networks allowing it to reduce overfitting, as it forces each neuron in the network, to not rely on a single input but to find different pathways to learn the relationship from inputs to outputs. Gal and Ghahramani (2016) propose that by leaving dropout on during inference, and sampling multiple times to construct the posterior distribution. When left on at inference time, each sample from the model can be viewed as a separate model. By performing multiple draws, a set of model versions all process the same input data in different ways, which is akin to ensembling.

During inference, Monte Carlo sampling is used, as defined in Equation 2.8.1.

A benefit of MC Dropout is that it can be applied to existing neural networks that use dropout, with the precondition that it should be used after every layer. The effectiveness of MC Dropout as a probabilistic neural network is a source of debate

(Ovadia et al., 2019), with issues being identified with the scalability and uncertainty estimates.

2.9 Gaussian Processes

Gaussian Processes are sets of random variables, where each combination of these variables form a Gaussian distribution (Rasmussen and Williams, 2006). GPs provide a prior distribution of these set of random variables, that when exposed to data can be transformed into a posterior distribution. A GP is parameterised by a mean function $m(x)$, and a covariance function $\Sigma(x)$, which is also known as a kernel function $\Sigma_{ij} = k(x_i, x_j)$. The assumption with the kernel function is that when it is applied to input points that are similar, the output should also be similar. The mean and covariance functions of the GP are typically optimised to best fit the observed data using a technique such as Maximum Likelihood Estimation (MLE).

GPs scale poorly with the number of training data points. To address this, Sparse GPs utilise inducing points, which are used to approximate the training data points and reduce the computational complexity.

When the likelihood of a GP is Gaussian, such as for regression there is a closed-form linear algebra solution for inference. However, for tasks where the likelihood is non-Gaussian such as classification which uses a softmax likelihood, there is no closed-form solution and the likelihood needs to be approximated. Hensman et al. (2015) proposes a variational, inducing point framework for GP classification that uses Variational Inference (VI) to approximate the posterior. While previous classification models scaled poorly with increased training dataset sizes, this approach can be used with large dataset sizes (millions of points). This thesis uses this approach to benchmark the proposed probabilistic neural network approaches. The GPyTorch framework (Gardner et al., 2018) was utilised for this approach. A radial basis function (RBF) kernel was used with a length scale of e which has been empirically selected.

2.10 Results

The results presented here compare the previously introduced methods for habitat modelling; *BNN* (here referring to BNN trained with SVI), *Drop* (BNN using MC Dropout), *VGP* (Variational Gaussian Processes classifier) and *cluster assignment*. The objective is to assess each model on how accurate it is, its versatility, its data requirements and finally its measure of uncertainty. This will decide what habitat modelling method to use to plan further sampling (in Chapter 4) and will also be crucial to understanding how best to use these probabilistic models to plan further sampling. The assessment is first focused on dives from the O'Hara / Waterfall regions before applying it to dives from the entire Fortesque area. A list of these dives and the respective habitat class counts can be found in Table 2.1.

2.10.1 O'Hara and Waterfall

Each habitat modelling method is evaluated on dives from the O'Hara and Waterfall regions as these areas are close geographically and more likely to have similar bathymetry→habitat relationship. This allows for a fair comparison between the habitat modelling methods.

There are two dives in the O'Hara region (*O'Hara 07*), *O'Hara 20* and two in the Waterfall region (*Waterfall 05*, *Waterfall 06*). These are displayed in Figure 2.7a. Four training/testing partitions are created:

- Train: *O'Hara 07*, *Waterfall 05*; Validation: *O'Hara 20*, *Waterfall 06*
- Train: *O'Hara 20*, *Waterfall 06*; Validation: *O'Hara 07*, *Waterfall 05*
- Train: *O'Hara 07*, *O'Hara 20*; Validation: *Waterfall 05*, *Waterfall 06*
- Train: *Waterfall 05*, *Waterfall 06*; Validation: *O'Hara 07*, *O'Hara 20*

The results for training and testing on each partition for both four and six classes are displayed in Figure 2.21. The average of each of these partitions are displayed

in 2.3. The *BNN* and *Drop* methods perform similarly across all datasets. Overall, these methods outperformed the *VGP* method. Cluster assignment, which is used primarily as a benchmark, performs poorly across all the datasets, highlighting the need for more advanced methods. The accuracy for each method is higher when it is trained on a dataset from the same region as which it is evaluated, indicating there is a difference in the bathymetry-habitat relationship between the two regions.

	BNN	Drop	VGP	Cluster Assign
4 Class	0.75	0.76	0.73	0.57
6 Class	0.56	0.55	0.51	0.36

Table 2.3 – Accuracy of each method for habitat modelling on the O'Hara / Waterfall region

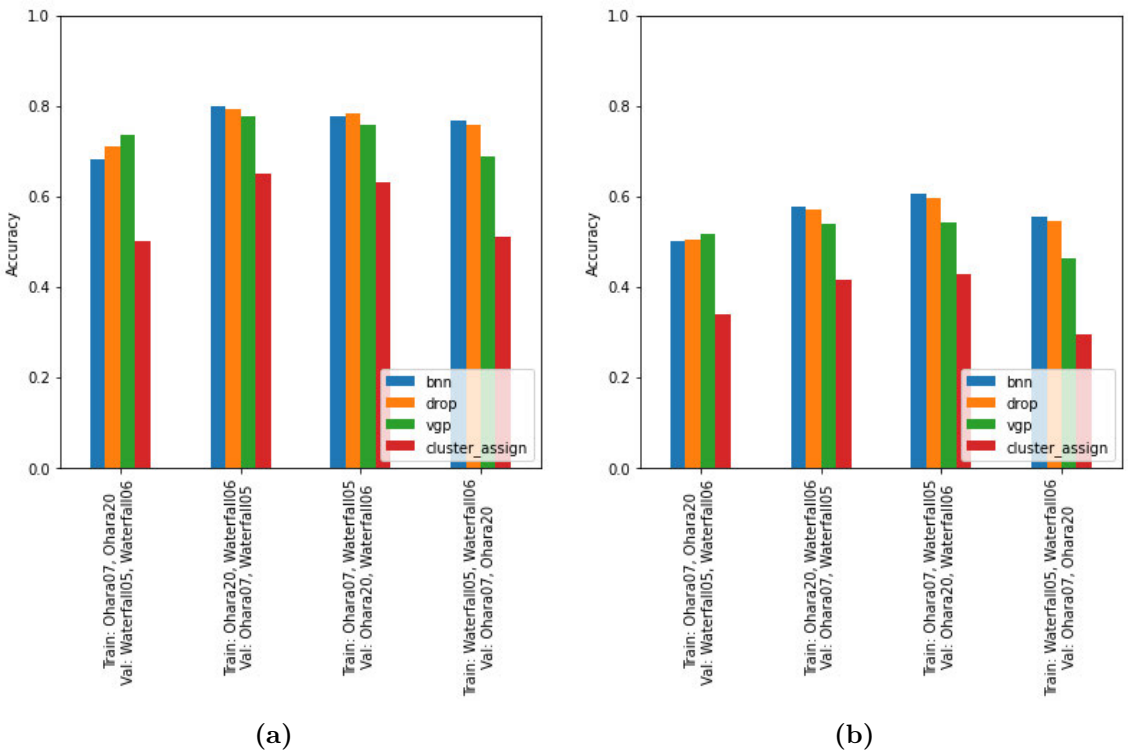


Figure 2.21 – Validation accuracy for different data partitions for dives in the O'Hara / Waterfall region.

To further analyse the accuracy of each habitat model, confusion matrices are presented in Figures 2.22 and 2.23. For the four class datasets (Figure 2.22) the primary sources of confusion are between the sand, rubble classes, and between the reef and

kelp classes. Overall, there is a relatively low level of confusion, given the difficulty in predicting the habitat class from the bathymetry. There is a significant accuracy drop for the 6 class datasets (Figure 2.23), as it becomes harder to resolve the finer resolution classes from the bathymetry. The added complexity of differing between the reef-classes; patchy reef, low relief reef and high relief reef is the primary driver of the accuracy drop. These class definitions can be hard to disambiguate even through imagery, making the classification task difficult when using bathymetry as the input.

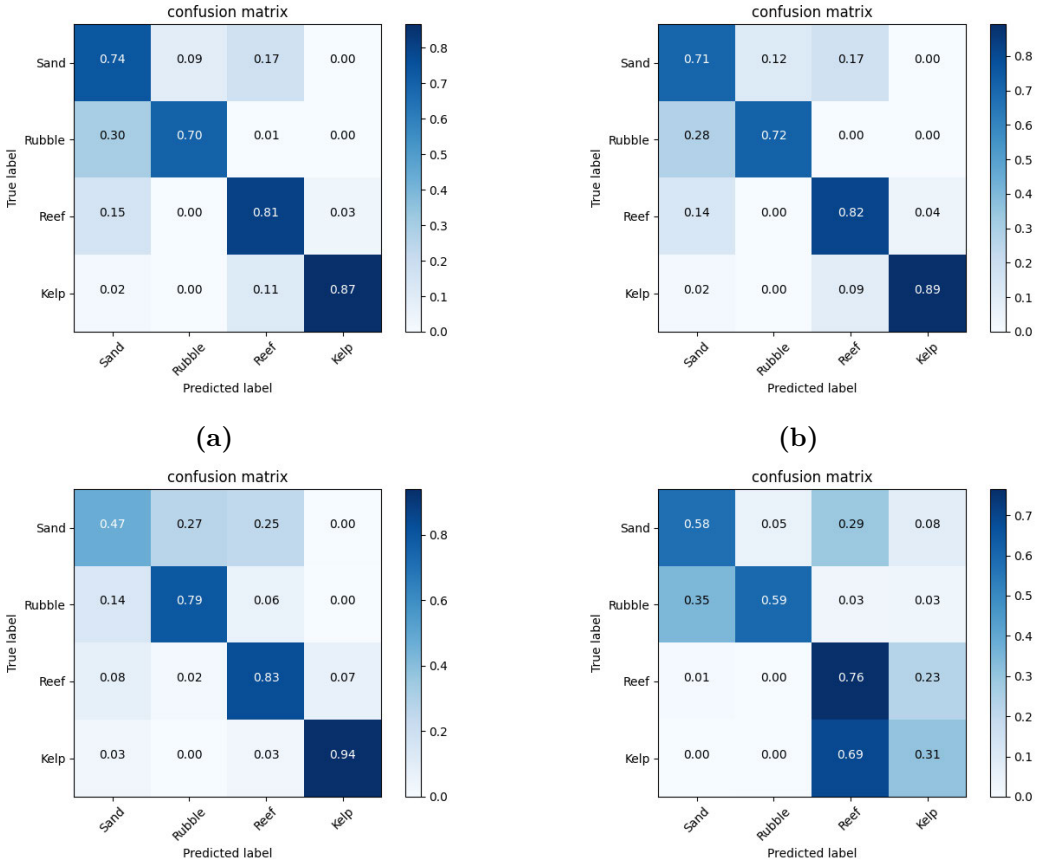


Figure 2.22 – Confusion matrices for each method for the 4 class dataset. (a) *BNN*, (b) *Drop*, (c) *VGP*, (d) *Cluster Assign*. Trained on *O'Hara 07 + Waterfall 05* - Tested on *O'Hara 20 + Waterfall 06*. The neural network methods are more accurate than the other approaches.

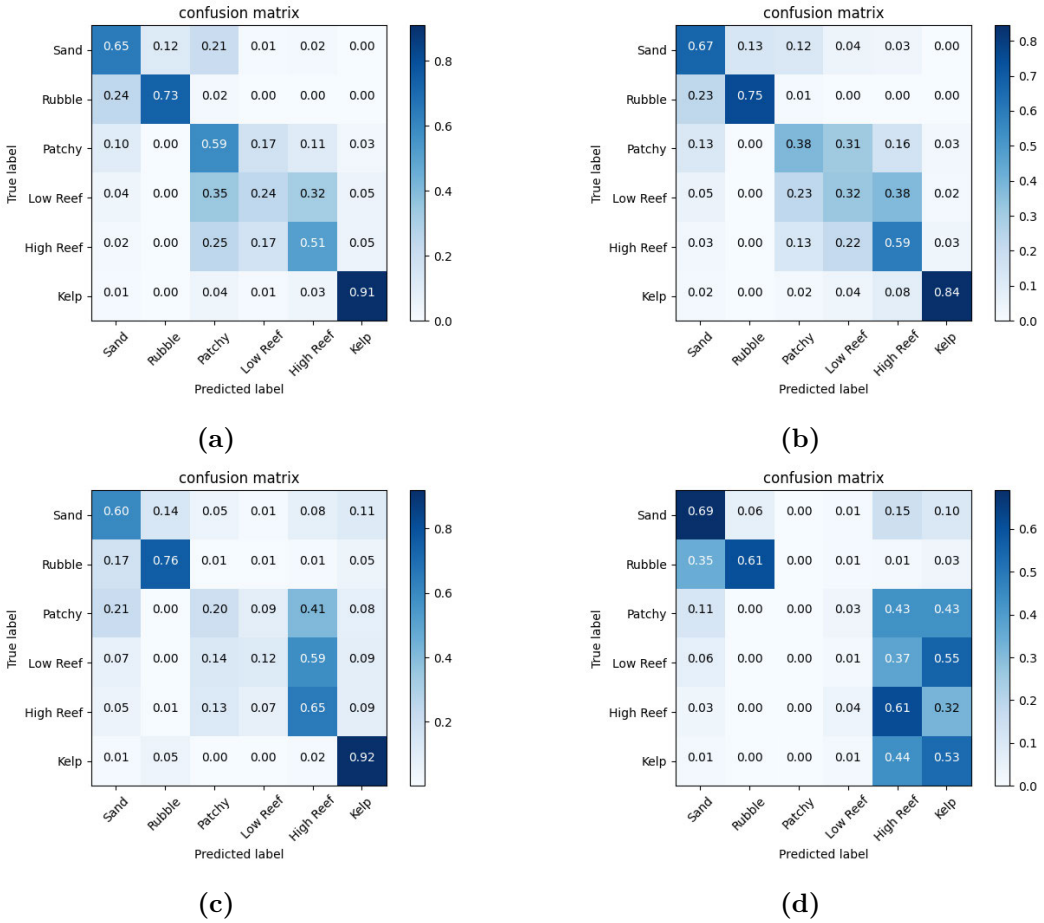


Figure 2.23 – Confusion matrices for each method for the 6 class dataset. (a) *BNN*, (b) *Drop*, (c) *VGP*, (d) *Cluster Assign*. Trained on *O'Hara 07 + Waterfall 05* - tested on *O'Hara 20 + Waterfall 06*. The neural network based methods are more accurate, particularly in the hard-to-disambiguate reef classes.

2.10.2 Extended Fortesque

To show the generalisation of the habitat models over a larger area, the habitat models are trained and evaluated on AUV surveys from the entire Fortesque region. The locations of these dives are shown in Figure 2.6. These dives are spread over a larger spatial distance and as such the relationship between bathymetry and habitat class may differ. Four groups of dives are composed, with each group containing a dive from each area. The composition of each group is shown in Table 2.4. From these groups, four training / testing combinations are composed:

- Combination 1 - Train: Group 1; Test: Group 2
- Combination 2 - Train: Group 2; Test: Group 1
- Combination 3 - Train: Group 3; Test: Group 4
- Combination 4 - Train: Group 4; Test: Group 3

Group 1	Group 2	Group 3	Group 4
O'Hara 07	O'Hara 20	O'Hara 07	O'Hara 20
Waterfall 05	Waterfall 06	Waterfall 06	Waterfall 05
Hippo 09	Hippo 13	Hippo 13	Hippo 09
Little Hippo 12	Little Hippo 11	Little Hippo 12	Little Hippo 11
Chevron 14	Chevron 10	Chevron 10	Chevron 14
Sisters 16	Blowhole 15	Sisters 16	Blowhole 15
High Yellow 19	Patch Reef 08	Patch Reef 08	High Yellow 19

Table 2.4 – Surveys from the Fortesque region arranged into groups, where different permutations of these groups are used for training and testing. These groups are formulated to ensure a relatively even spread of habitat classes and regions are present in each group.

The average accuracy across all the partitions is presented in Table 2.5, while the average accuracy for each partition is displayed in Figure 2.24. These results show a significant performance difference between the neural network based methods and the GP classifier. The classification problem when classifying dives from the entire Fortesque area is more difficult than classifying dives from the O'Hara / Waterfall region, as the bathymetry-habitat relationship differs more between the geographic areas. The result of this is more data-noise, where the network is seeing features that are similar that have different habitat labels, which is difficult for the classifier to resolve. For this classification problem, the improved discriminatory ability of neural networks are better able to resolve these differences, which is evident in the improved accuracy.

	BNN	Drop	VGP	Cluster Assign
6 Class	0.5	0.49	0.41	0.25

Table 2.5 – Accuracy of each method for habitat modelling using all the surveys in the Fortesque region with the six class datasets. There is significant variation between different areas within the Fortesque area which is reflected in the drop in accuracy compared to the O'Hara / Waterfall task.

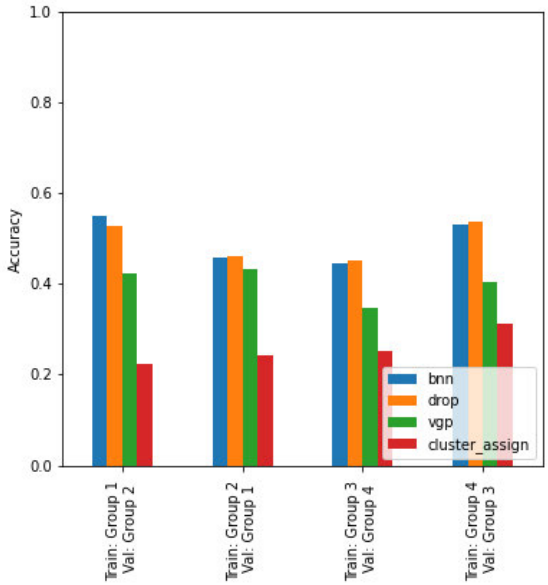


Figure 2.24 – Comparing the different methods for habitat modelling on all the dives in the Fortesque region. The neural network methods outperform the other methods on all dataset combinations.

2.10.3 Data Requirements

The process of conducting a survey and annotating the imagery is expensive and therefore assessing the data needed to create a habitat model is important. Methods that can create habitat models with fewer labelled samples are preferred, as they can significantly reduce the cost of habitat modelling. This section evaluates each habitat modelling method with different dataset sizes in order to gauge the data requirements for each model.

To create the dataset partitions of different sizes, each dataset is 'thinned', where the distance between samples is increased. This is repeated for each distance in the sequence $d \in \{1, 4, 8, 16, 64, 128, 256\}$ meters. The mapping from the distance

between samples to the number of samples for the *O'Hara 07* dataset is displayed in Table 2.6.

Distance (m)	Number of Samples	Number of Balanced Samples
1	3441	2424
4	1053	678
16	276	162
64	70	30
128	33	18
256	15	6

Table 2.6 – Dataset sizes for the *O'Hara 07* dataset with dataset thinning, where the distance between samples is increased.

Each habitat model method; *BNN*, *Drop*, *GP* and *Cluster Assignment*, is trained on each of the dive combinations, for each of the thinned dataset combinations. There are two dive combinations, with the first combination being trained on *O'Hara 07* and *Waterfall 05*, test on *O'Hara 20* and *Waterfall 06*, while the second is reversed. The testing accuracy for each of these combinations is displayed in Figure 2.25. With fewer samples, the MC Dropout method outperforms the other methods. The cluster assignment method performs well for small dataset sizes but the accuracy does not increase significantly for larger datasets. The variational inference based methods (*BNN* and *GP*) need larger dataset sizes before they become effective. Although the MC-Dropout method has significantly better accuracy, it can be overconfident for low dataset sizes, as analysed in Section 2.12.

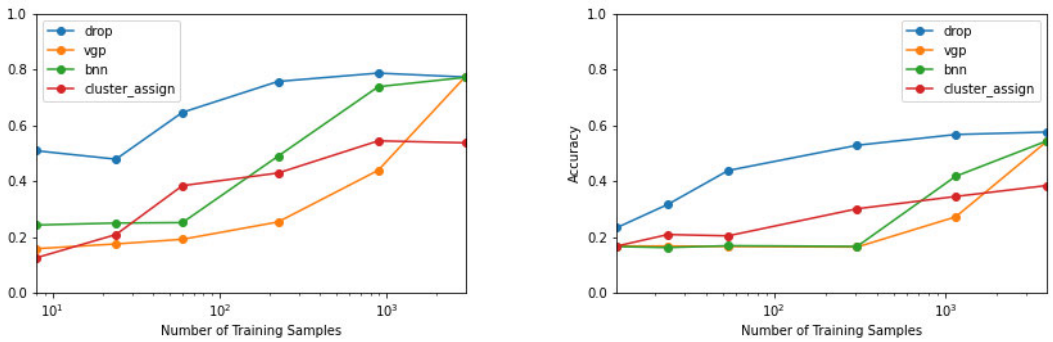


Figure 2.25 – Data requirements for 4 classes (a) and 6 classes (b). For small dataset sizes, the Drop method is more accurate than the other methods.

2.10.4 Feature Comparison

The habitat models input a vector of features extracted from the bathymetry. Section 2.6 presents two different methods for feature extraction; geometric features and encoded features. Geometric features use the depth map to extract features representing the shape of the bathymetry; slope, aspect and ruggedness. They have been traditionally used for habitat modelling (Brown et al., 2011a). Encoded features use an autoencoder to compress patches of bathymetry into a feature space. The aim is to generate a more expressive feature space. This section compares using each of these feature representations for habitat modelling, which is presented in Figure 2.26. The *BNN* model is used with the 6 class dataset on dives from the O'Hara / Waterfall region. The encoded features represent a significant performance improvement over the geometric features, suggesting there is added information in the encoded feature space that can be used for classification. Further investigation of geometric feature combinations could lead to improved performances (such as extracting features over multiple scales), however encoded features offer an effective, versatile and 'hands off' approach.

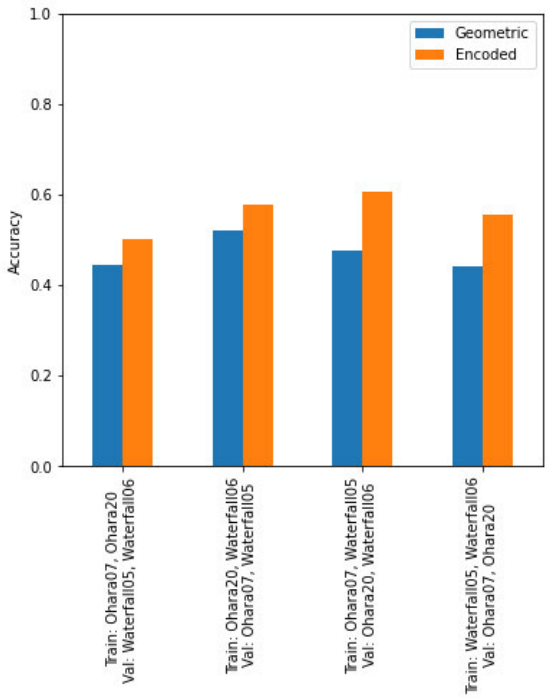


Figure 2.26 – Comparing geometric and encoded features for habitat modelling. Encoded features have a larger, but more informative feature space which enables more accurate models to be trained.

2.11 Inference and Uncertainty Estimation

2.11.1 Habitat Maps

The habitat model can be inferred over the entire bathymetry region, providing a classification of the benthic habitat for each raster cell. This is displayed for the O'Hara / Waterfall region in Figure 2.27, with the testing dataset overlayed to show the ground-truth labels. As displayed, there is a high level of correspondence between these ground-truth labels and the predicted category maps. The *BNN* and *Drop* methods produce similar habitat maps which is expected as they are similar network structures. The habitat maps generated from the neural network methods differ significantly to the map generated by the *VGP*. The extrapolation of the *VGP* has produced odd results, predicting there is kelp in the deeper section of the reef (which is not true and deeper than kelp occurs). Similarly, it predicts there is screw-shell

rubble in the shallow sections of the map, despite that only occurring in deeper areas in these datasets. These areas are out-of-distribution compared to the training set, but still show the difficulty in predicting with the *VGP* in areas that are not well-represented by the training dataset. This behaviour is likely behind the significant accuracy drop when using all the Fortesque dives compared to just the dives from O'Hara / Waterfall, which are of similar distribution.

2.11.2 Entropy

Understanding where habitat models are uncertain provides insights into the interpretation of these models and can be used to direct further sampling. The entropy of the habitat models highlights where the model is confused between the class of the prediction. The entropy for a prediction $\hat{\mathbf{y}}$ is given by:

$$\mathbb{H} = -\sum_{c \in \text{classes}} p(\hat{\mathbf{y}}_c) \log p(\hat{\mathbf{y}}_c), \tag{2.16}$$

where $p(\hat{\mathbf{y}}_c)$ is the probability of class c in the prediction.

The entropy is plotted spatially in Figure 2.28. These plots show the entropy is higher around the reef areas, where there are multiple classes that the sample could be. The entropy will highlight areas where there is class confusion but not necessarily where the model is truly uncertain about. If using the entropy to guide further sampling, the entropy will lead it to these areas of high aleatoric (data) uncertainty where there is confusion between the classes, even if these areas have already been explored by the habitat model and further sampling will not be able to further resolve the class confusion. Therefore probabilistic models should be used, and use the uncertainty from these models to drive further sampling.

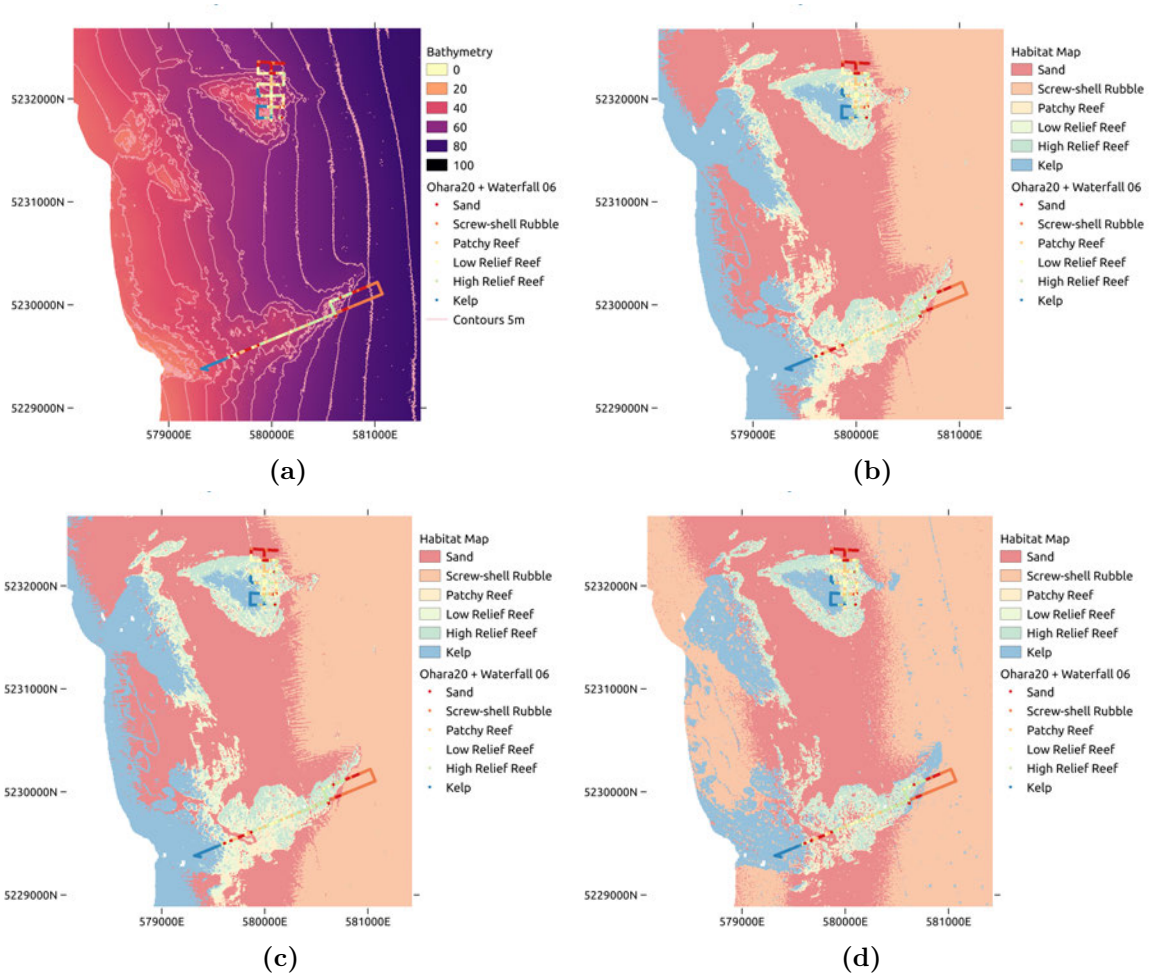


Figure 2.27 – Habitat maps for the O'Hara / Waterfall region, generated by each of the habitat modelling methods. The test dataset is overlayed with the same colour scheme, showing areas where the predictions are correctly and incorrectly classified. (a) shows the bathymetry, (b) *BNN*, (c) *Drop*, (d) *VGP*. Overall, the habitat maps appear accurate and follow the key habitat boundaries. On closer inspection, it is evident there are misclassifications, including where *VGP* predicts there is kelp towards the deeper section of the reef, and for all methods where there is confusion between the reef classes, particularly in the centre of O'Hara reef. Coordinates are displayed in UTM zone 55S.

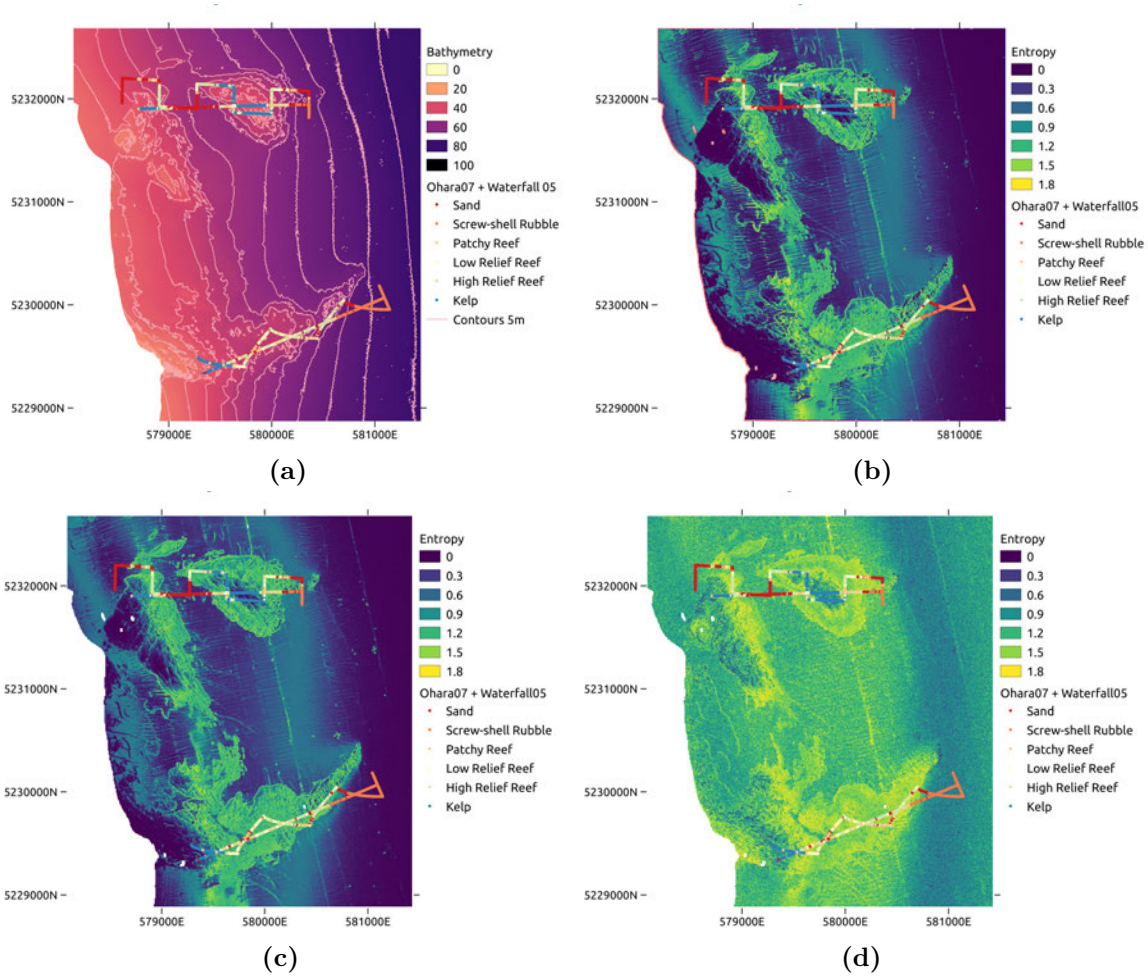


Figure 2.28 – Entropy maps for the O'Hara / Waterfall region, generated by each of the habitat modelling methods, with the class labels overlaid. (a) the bathymetry, (b) *BNN*, (c) *Drop*, (d) *VGP*. Coordinates are displayed in UTM zone 55S.

2.11.3 Deconstructing Uncertainty

A probabilistic classifier such as a BNN or GP classifier can estimate the uncertainty associated with a prediction. As these models are complex, exact inference is not possible and the posterior needs to be constructed from Monte Carlo sampling. With a BNN, for each Monte Carlo sample a version of the network is created by sampling from the weight distributions of the network. For a Monte Carlo Dropout based network, dropout is simply left on during inference, which creates a version of the network. By performing multiple draws of the network, the posterior distribution of the network can be approximated. From this posterior distribution, the network's uncertainty associated with the prediction can be deduced.

The uncertainty estimate of a probabilistic model can stem from either uncertainty in the data (aleatoric) or uncertainty in the model (epistemic). Using the BNN approximation technique Monte Carlo dropout (Gal, 2016), Kendall and Gal (2017) deconstruct the uncertainty into its epistemic and aleatoric components. A multi-head network is used, where one head is the predicted mean (class vector) and the other is the predicted variance. Both the predictive mean and predictive variance are trained, where the predictive variance is learned implicitly, not requiring uncertainty labels. Kwon et al. (2018) build upon Kendall and Gal (2017), by deconstructing the uncertainty without using an extra network head.

The total uncertainty is given by (Kwon et al., 2018):

$$\text{Var} = \underbrace{\frac{1}{T}\sum_{t=1}^{T}\text{diag}(\hat{y}_t) - \hat{y}_t^{\otimes 2}}_{\text{aleatoric}} + \underbrace{\frac{1}{T}\sum_{t=1}^{T}(\hat{y}_t - \bar{y})^{\otimes 2}}_{\text{epistemic}} \tag{2.17}$$

Where $\hat{y}_t$ is the prediction for each sample, t, of the model and $^{\otimes}$ is the tensor-wise product. The deconstructed uncertainties provide insight into how to sample next. As epistemic uncertainty highlights areas of model uncertainty, further sampling of these areas can improve the model, enabling a more comprehensive habitat model to be formed with less sampling. Aleatoric uncertainty represents noise in the data and cannot be reduced by further sampling.

The deconstructed uncertainty is plotted spatially in Figure 2.29 (aleatoric) and 2.30 (epistemic). The aleatoric map highlights areas of high data uncertainty which is focussed on areas where multiple classes are likely, in particular around the reef regions, where the three classes of reef (patchy reef, low-relief reef and high-relief) can be hard to disambiguate. The epistemic map highlights the areas of high model uncertainty, which should highlight areas that are not represented in the training datasets. As demonstrated in Figures 2.30b and 2.30c, the Bayesian neural network methods have higher epistemic areas for unexplored regions. These include the shallower reef areas towards the left of the map and the interface between the sand and rubble, towards the right of the map. The uncertainty deconstruction was originally designed for neural networks but can also be used when sampling from a GP. The epistemic map generated by the GP exhibits a generally higher level of uncertainty, with a slight increase in areas consistent with the neural network methods, including to the left of the map where it is shallower. Using probabilistic models is an integral tool for habitat modelling. The prediction of the uncertainty along with a prediction allows for more in-depth interpretation (i.e. to not trust the areas with high uncertainty) and the ability to direct further sampling to these areas of high uncertainty. Using the habitat model to drive further sampling is the focus of Chapter 4.

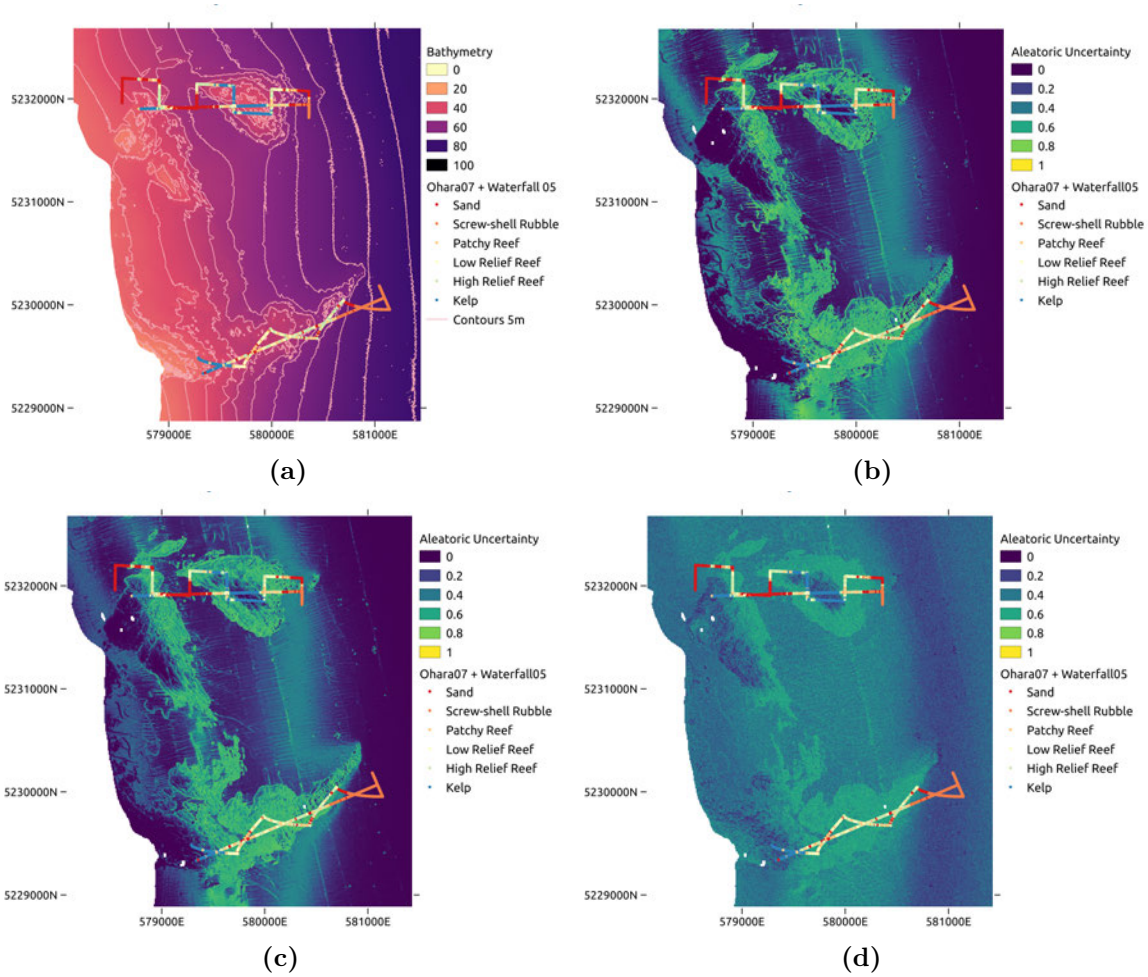


Figure 2.29 – Aleatoric uncertainty maps for the O'Hara / Waterfall region, generated by each of the habitat modelling methods, with the class labels overlaid. (a) the bathymetry, (b) *BNN*, (c) *Drop*, (d) *VGP*. The aleatoric uncertainty is highest in areas of high data uncertainty, where there is confusion between multiple classes. This is highest on the intersections between multiple habitats, such as in the centre of O'Hara reef, where there is confusion between the three reef classes. It is also moderate on the intersection between the sand and rubble class. Coordinates are displayed in UTM zone 55S.

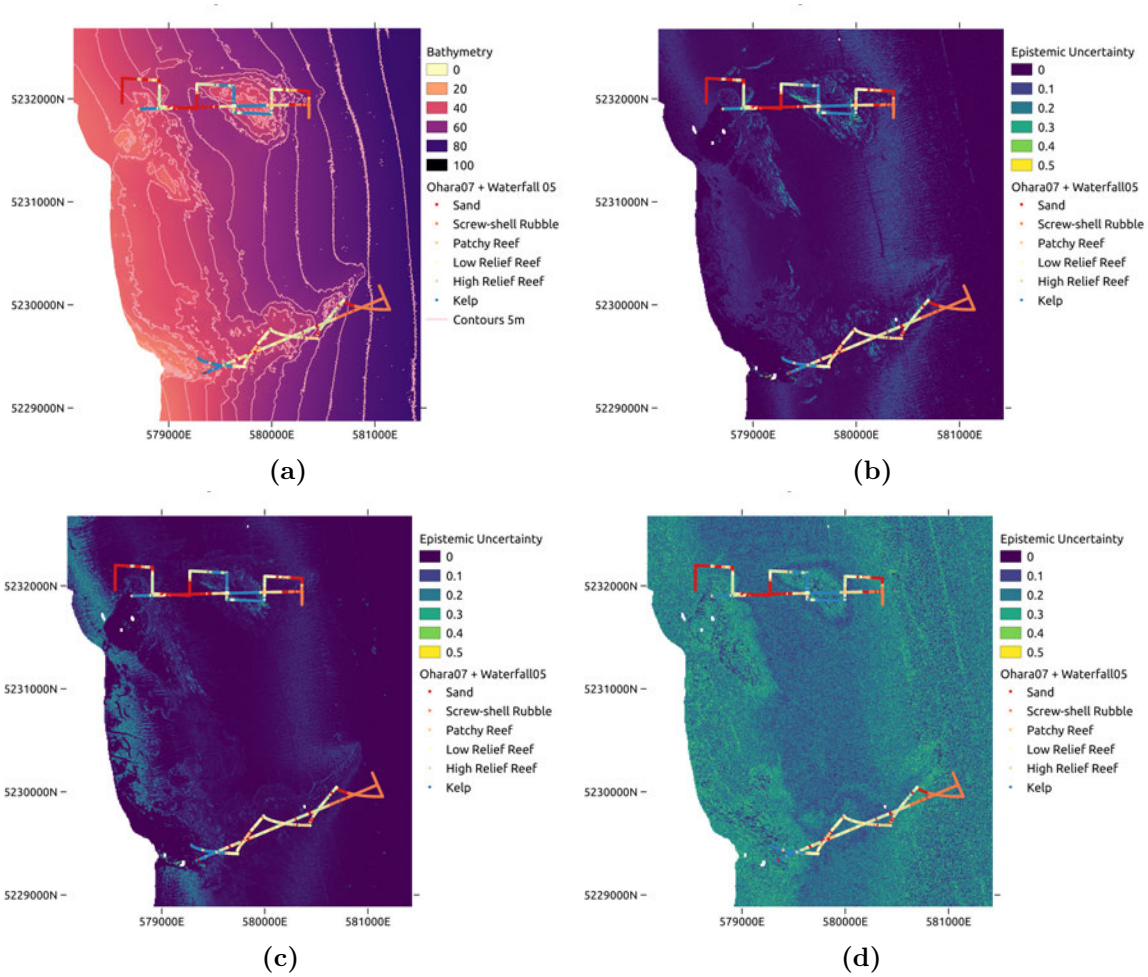


Figure 2.30 – Epistemic uncertainty maps for the O'Hara / Waterfall region, generated by each of the habitat modelling methods, with the class labels overlaid. (a) the bathymetry, (b) *BNN*, (c) *Drop*, (d) *VGP*. There should be higher epistemic uncertainty in areas that are different from those in the training dataset. Examples of these are the shallower section, which has higher epistemic uncertainty in all three methods, and in some of the reef sections. The sand / rubble intersection also has moderate epistemic uncertainty. Overall, the epistemic uncertainty is lower in areas that have been sampled. Coordinates are displayed in UTM zone 55S.

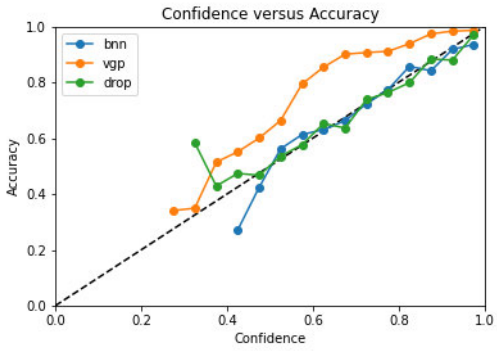
2.12 Validation of Uncertainty Estimates

The habitat models are probabilistic models that predict an uncertainty estimate associated with the prediction. This can be useful to provide feedback to end-users or to drive further sampling. This section explores and validates these uncertainty estimates, to identify if these models are appropriate for model-driven adaptive sampling.

Ovadia et al. (2019) compare various probabilistic classifiers and evaluate how the uncertainty estimates respond to Out Of Distribution (OOD) data, which is particularly important for safety critical applications. They simulate OOD data on various datasets. On the MNIST dataset, the digits are rotated, shifted and skewed, to progressively look less like the original digits. For the image based datasets (CIFAR and ImageNet), the images are progressively corrupted by several techniques including adding noise, blurring, pixelating and illumination changes.

They identify two metrics that can be used to summarise the performance of the probabilistic classifier; Expected Calibration Error (ECE) (Naeini et al., 2015) and Brier Score (Brier, 1950), which establish how well a model is 'calibrated'. The calibration of a model can be evaluated by plotting the confidence versus the accuracy (Lakshminarayanan et al., 2017), where the confidence of the prediction is the probability corresponding to the selected class. These plots are constructed by binning each confidence interval, and calculated the accuracy for each prediction in these bins. The confidence versus accuracy can then be plotted. A perfectly calibrated classifier should have the confidence matched with the accuracy, as shown in Figure 2.31. At 1.0 confidence, the perfectly calibrated classifier would be accurate 100% of the time, at 0.5 it would be accurate 50% of the time and so on. Figure 2.31a plots the confidence versus accuracy for each classifier, when trained on the *O'Hara 07* and *Waterfall 05* datasets, and tested on the *O'Hara 20* and *Waterfall 06* datasets. The *BNN* and *Drop* models are well calibrated as they fit the $confidence = accuracy$ line, which is different than vanilla neural networks with a softmax output, which are typically overconfident. This is likely a factor of the inbuilt regularisation and

the use of a statistically similar training and testing dataset. The *VGP* method is under-confident.

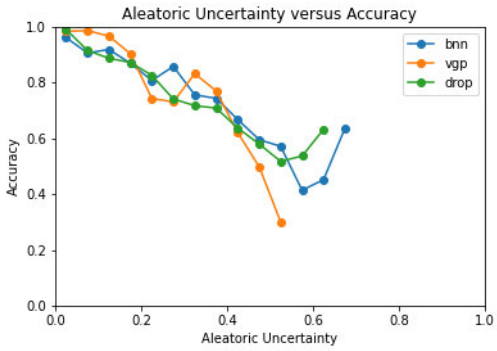


(a)

Figure 2.31 – Confidence versus accuracy, which analyses the calibration of a model. The black line represents a perfectly calibrated model, points above the line are underconfident, while points below the line are overconfident. Both the *BNN* and *Drop* methods are well calibrated for this dataset, while the *VGP* method being underconfident.

A similar approach can be used to evaluate the model's uncertainty estimations and the drivers of uncertainty. Figure 2.32a plots the aleatoric uncertainty versus accuracy for each of the classifiers. It shows a clear trend, as aleatoric uncertainty increases the accuracy decreases. For this dataset, there is a high level of label noise, which stems from the resolution difference between the bathymetry and benthic habitat. A single bathymetric patch can contain several different benthic habitats, which results in high label noise. The aleatoric (data) uncertainty is the primary driver of errors for this dataset.

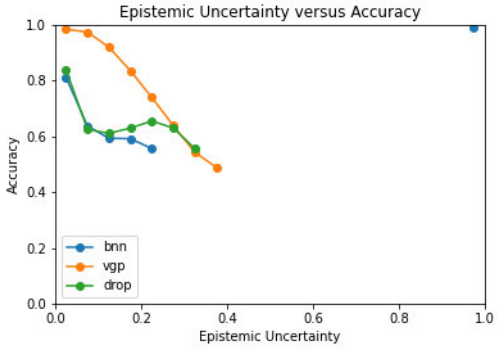
The model should predict high epistemic uncertainties for OOD, which is data that is underrepresented in the training set. In these cases, the model is extrapolating and this should be reflected in the uncertainty estimate. Figure 2.33a plots the epistemic uncertainty versus the accuracy. For each of the models, a higher epistemic uncertainty is correlated with a lower accuracy. The *VGP* classifier predicts higher epistemic uncertainties than the neural network methods. For zero epistemic uncertainty, the *BNN* and *Drop* methods report 80% accuracy, whereas ideally it should be 100%. This could be due to the networks being overconfident, or just that the



(a)

Figure 2.32 – The aleatoric uncertainty versus the accuracy. As uncertainty increases, the accuracy decreases, highlighting that data uncertainty is a driver for misclassifications.

aleatoric uncertainty was the key driver of error for this dataset. In the spatial epistemic plots (Figure 2.30) the neural networks predict higher uncertainty in areas that are significantly different to the training data paths. This is intuitive and highlights that the epistemic uncertainty estimate is meaningful. The *VGP* method predicts higher epistemic uncertainty in areas that appear similar to the training path, highlighting that the uncertainty is not being deconstructed effectively.



(a)

Figure 2.33 – Epistemic uncertainty versus accuracy. As expected, as the uncertainty increases, the accuracy decreases.

For probabilistic models to be used in active learning algorithms (which is the objective of Chapter 4), they need to be robust and produce valid uncertainties with a range of training dataset sizes. Figure 2.34 plots a histogram of the epistemic

uncertainty obtained from each model for different dataset sizes. The datasets are made smaller by increasing the distance between each sample (i.e. the 128m dataset is the smallest). The assumption is that there should be higher levels of epistemic uncertainty for a smaller dataset, and this should generally decrease as more samples are collected. This pattern is evident for the *BNN* method, which sees a distribution shift in the epistemic uncertainty as the dataset size is changed. This pattern is evident to a lesser extent in the *Drop* method and it is non-existent for the *VGP* method, although the *VGP* model was poorly trained. The *Drop* method reports higher accuracy for smaller dataset sizes (Figure 2.25) however it is not reporting higher epistemic uncertainties, highlighting that it is overconfident. Both the neural network methods have low counts of high uncertainty classifications for larger dataset sizes which indicates overconfidence. This is a significant issue for uncertainty-based active learning, which relies on valid reporting of uncertainties for all dataset sizes. This appears most evident for the *Drop* method, however this issue is present across all probabilistic classifiers (Ovadia et al., 2019).

Probabilistic classifiers construct the posterior distribution from Monte Carlo sampling, where during each pass, the model makes a prediction. For a prediction to have high uncertainty, it must predict different outcomes on each pass. For OOD data the model extrapolates. A classifier can become overconfident on OOD data if the data point is more similar to one class than all the other classes. This is demonstrated in Figure 2.36, where the new data point is significantly OOD, but closer in feature space to the yellow class than the other classes. This new data point could belong to a new class. If the model extrapolates to this new data point, it will be unlikely to predict the other classes resulting in low uncertainty. The possibility of a model becoming overconfident when exposed to OOD data, strengthens the need for alternate methods of identifying OOD data in an adaptive sampling algorithm. This is further examined in Chapter 4.

This chapter uses the relationship between the bathymetric structure as the only input to the habitat classification, and does not take into account the spatial proximity of samples into account. This can be an issue, as the relationship between the

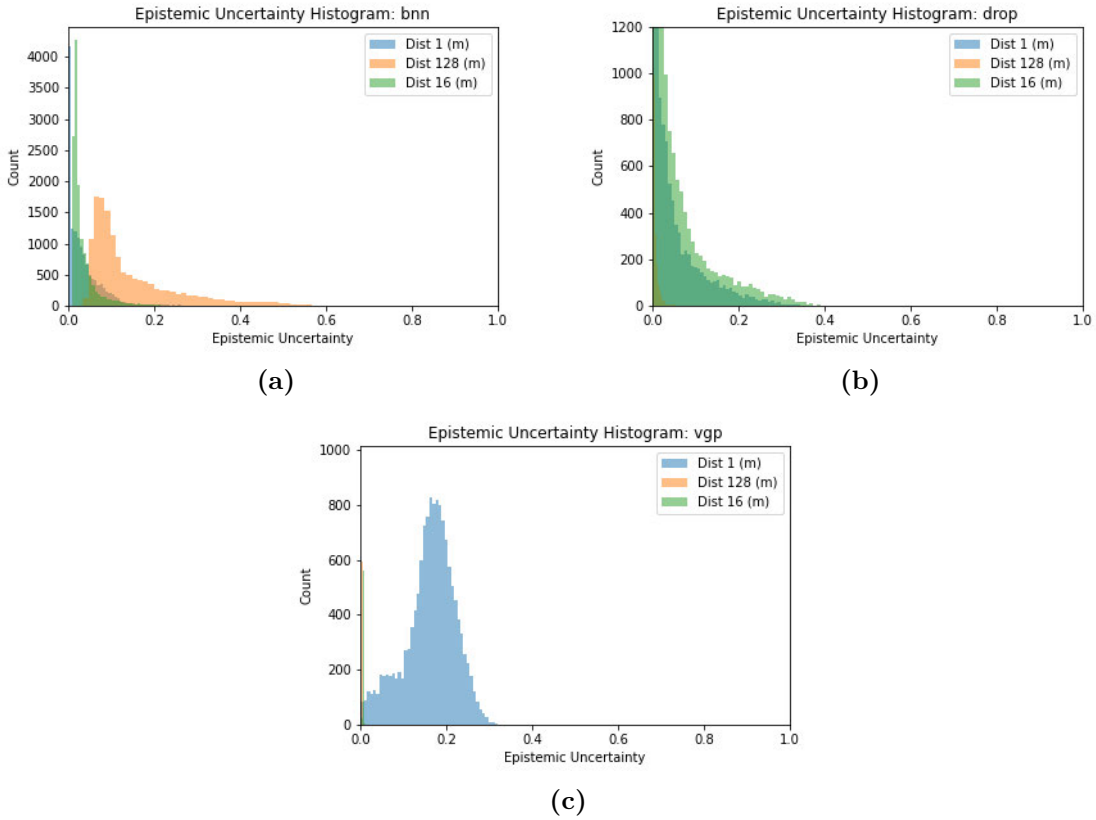
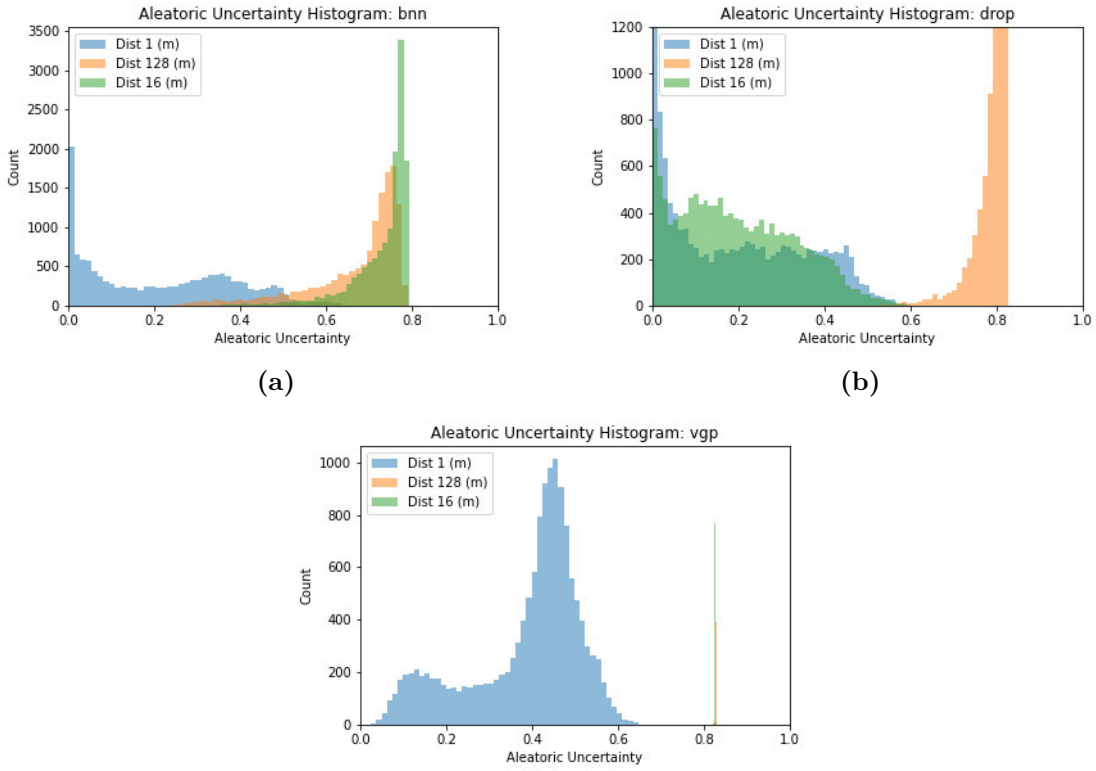


Figure 2.34 – Histogram of the epistemic uncertainty for models trained with different dataset sizes. These graphs should high counts of higher epistemic uncertainties for smaller dataset sizes (larger distance between samples). This trend is evident for the *BNN* method. Despite achieving the highest accuracy for small dataset sizes, the *Drop* method appears overconfident when there is little data. The *VGP* method was not properly trained on these small datasets and does not show a meaningful result.

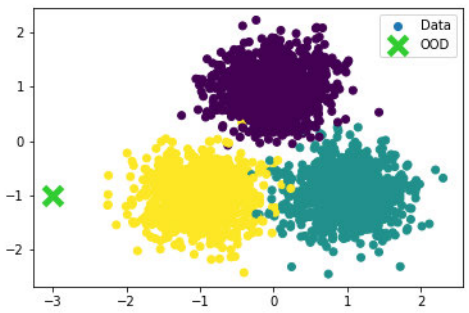
bathymetry to the benthic habitat can change between areas. For the AUV surveys of the Fortesque region in Tasmania that are used in this chapter, there is a difference in the bathymetry to benthic habitat relationship over different sites in this single region. Including location information in the feature vector could help the model identify that data from a new area is different to areas observed before. Alongside this location information, it is possible to include auxiliary information (current, temperature etc.) into the input feature vector, that can further refine the habitat estimation.



(a) **(b)**

(c)

Figure 2.35 – Histogram of the aleatoric uncertainty for models trained with different dataset sizes. As the dataset size decreases, the aleatoric uncertainty increases.



(a)

Figure 2.36 – A visualisation of Out Of Distribution (OOD) data. It shows a simple classification example, where there are three classes. The green cross shows a new data point, that is significantly different to existing data. A probabilistic classifier should identify this as highly uncertain, however if this point is most similar to the yellow class and not similar to the other classes, which make it unlikely that a model would not predict the yellow class.

2.13 Summary

Benthic habitat maps are a useful output of benthic scientific surveys, providing a visualisation and characterisation of the target survey area. Developing methods to create accurate habitat models with a limited sampling budget is desirable, as it can dramatically reduce the cost in conducting these surveys. This chapter investigated the use of probabilistic models for use in supervised habitat modelling. It demonstrated that Bayesian neural networks can produce accurate habitat predictions, while also producing a reasonable uncertainty estimate. This uncertainty estimate can be used to identify areas that are underrepresented in the training data. While these uncertainty estimates were reasonable, the models can also be overconfident in the predictions.

Autoencoders were developed to provide an alternative feature space to the geometric features commonly used in literature. These features were demonstrated to be more expressive, as the entire bathymetric space was summarised in the feature space. This resulted in higher classification accuracy for supervised habitat modelling.

Clustering on the bathymetric feature space enables the identification of similar areas of bathymetric terrain. This is a useful tool for visualising the benthic environment before any in-situ samples have been collected. Furthermore, it provides an avenue to plan initial benthic surveys. This is further explored in Chapter 3.

This chapter has demonstrated a model for habitat mapping that can be used for planning AUV surveys. Using the uncertainty estimate to drive further surveys is appropriate, as long as there are solutions to deal with the overconfident model in some OOD scenarios. Planning future surveys is investigated in Chapter 4.

Chapter 3

Initial Sampling

3.1 Introduction

In Chapter 2, the process of habitat modelling was investigated where the relationship between remotely-sensed data and in-situ observations of the seafloor is learnt, in order to predict the benthic habitat beyond where can be feasibly sampled. As this is a machine learning problem, the model should be trained on a data set that is representative of the overall input data distribution and contains all of the possible target classes. Therefore, for the habitat model to be representative of the survey area (area of interest), there should be visual observations of the benthic habitat in locations that cover the complete range of seafloor types evident in the remotely-sensed data, while also observing all of the different benthic habitats that exist in the survey area. Generally the areas to be characterised are vast (measured in millions to billions of square meters), and the image sensor footprint of the AUV is small (1-10 m^2), consequently only a minute portion of the survey area can be feasibly sampled. In order to collect data which is representative of the entire survey site, while satisfying the constraints of the vehicle capabilities, consideration must be given to the planned survey path. This chapter explores planning AUV trajectories such that an informative set of in-situ observations are collected, that can be used to train effective habitat models.

Currently, AUV surveys are mostly planned manually, with the survey planner identifying areas of interest while inspecting the remotely-sensed data. While expert knowledge can provide valuable insights when designing a survey, automatically generating a survey can provide a reliable and principled method for survey design (Foster et al., 2020). This chapter explores methods for autonomously planning a representative and comprehensive initial AUV survey by utilising prior bathymetry. Features are extracted from the depth map that represent the structure of the seafloor, which is indicative of the benthic habitat (Brown et al., 2011a). These features come from either established geometric processes or encoding bathymetric patches with an autoencoder. Geometric features represent the structure of the seafloor through a combination of depth, slope, aspect and ruggedness (Wilson et al., 2007). Encoded features are formed by compressing small patches of bathymetry with an autoencoder, with the latent space of the autoencoder being a compact representation of the input patch. The scale of both of these features changes according the the resolution of the underlying bathymetry, however these features can be extracted at several scales to complement the feature descriptor. The feature space is the collection of these features and is a representation of the range of bathymetric terrain found in the survey area. AUV paths are subsequently planned to maximise coverage of this feature space to ensure the AUV observes the complete range of bathymetric terrain and benthic habitats found in the survey area.

This chapter introduces the problem of exploring the feature space of remotely-sensed data by planning in physical space. It serves as an investigation into various approaches to address this problem. While this research focuses on the problem of visiting diverse bathymetric terrain with an AUV, this approach is applicable to any application where in-situ sampling is planned from remotely sensed data. Analogues to this problem exist in space exploration (Li et al., 2005) and agriculture monitoring (Ramin Shamshiri et al., 2018).

In this chapter, an information gathering framework is developed that consists of an objective function that rewards collecting features that are different to features already collected, and an evaluation function that scores paths based on the coverage

of the feature space. This framework is integrated into information gathering planners based on MCTS, RRT and placing survey templates. This is compared against a more conventional planning approach, that first proposes a set of points that represent the feature space and then plans a spatial path between them using set-TSP . These methods are evaluated based on how much of the feature space they explore given a set distance budget, with the results showing a significant increase in information collected compared to random sampling.

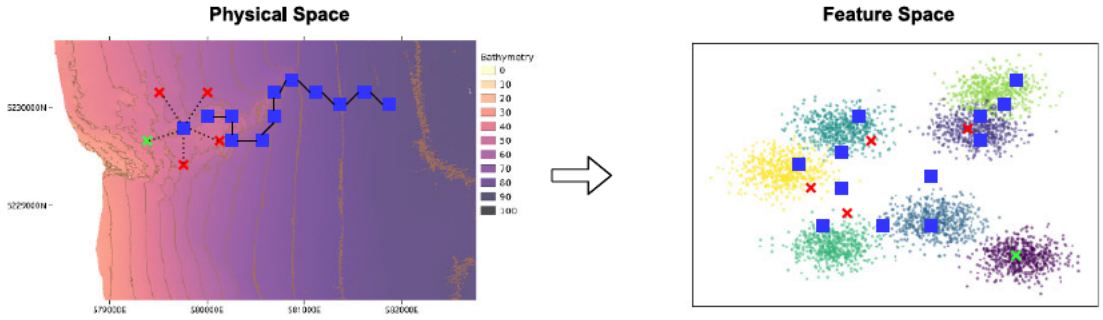


Figure 3.1 – An overview of linked feature space and physical space exploration, where the objective is to comprehensively explore the feature space given a budget in physical space. Features extracted from along the survey path are projected into the feature space. In this demonstration, the blue squares indicate features already on the path, with the planner making a decision about where to move physically (indicated by crosses). The planner chooses the green cross over the red cross as it occupies a more unexplored area of the feature space.

The primary contributions of this chapter are:

- The formalisation of the problem of exploring a feature space representation of remotely-sensed data for informative path planning of a mobile robot. This is presented in the case of exploring the bathymetric feature space for benthic survey planning. Following this, a novel approach to this problem is proposed that links feature-based active learning to robotic information gathering in order to plan AUV surveys that comprehensively explore the feature space. A suite of planners are developed, based on RRT, MCTS and setting survey templates that utilise a feature-based reward function to plan informative paths.

- Demonstration of this approach on several survey areas, each with different

sources of bathymetry, highlighting the versatility and value of this approach.

- Investigation of planning with different feature representations, showing that the methods proposed are suitable for various feature spaces and can be applied to analogue problems with different remotely-sensed data modalities.
- Development of evaluation criteria for initial surveys that do no not rely on visual habitat labels and use these criteria to evaluate each method's ability to plan initial benthic surveys.
- Integration of AUV motion and terrain constraints into the planning process to allow for safe operation of the AUV.

The remainder of this chapter is organised as follows: Section 3.2 provides an overview of the literature in this field; Section 3.3 outlines the methods for bathymetric feature extraction; Section 3.4 proposes a reward function for exploring the feature space and proposes a set of information gathering planners; Section 3.5 presents results for each of the proposed planners and Section 3.6 details the field trials conducted. Finally, Section 3.7 concludes the chapter and identifies avenues of future research.

3.2 Related Work

Robotic information gathering is path planning where the goal is to maximise an information reward. The need to autonomously explore an environment is multi-disciplinary while promoting many unique applications, including tracking biological hotspots with AUVs (Das et al., 2013), space exploration (Arora et al., 2019) and active perception and illumination (Sheinin and Schechner, 2016).

Traditionally, most robotic path planners would operate on discrete spaces, with the robot's environment decomposed into a grid. However, these approaches do not scale well to large or highly-dimensional environments, as the search space becomes too large. Sampling-based planners such as Probabilistic Road Maps (PRM) (Kavraki et al., 1996) and Randomly-expanding Information Gathering (RIG) trees (Karaman

et al., 2011) are able to operate in large and highly-dimensional environments while generating fast and usable solutions. More recently, Monte Carlo Tree Search (MCTS) planners have become popular, buoyed by the success of the go-playing algorithm, AlphaGO (Silver et al., 2016).

Hollinger and Sukhatme (2014) propose three information gathering algorithms; Rapidly-exploring Information Gathering (RIG)-Roadmap, RIG-Tree and RIG-Graph. These algorithms incrementally plan paths to maximise information given a budget constraint. RIG-Tree, which is based on RRT*, was shown to be the most effective. These algorithms were designed to be adaptable for different information goals by changing the reward function. Jadidi et al. (2019) use a RIG-Tree based planner to guide a robot to efficiently map a spatial field using a GP (Gaussian Process), where the reward function is derived from mutual information. They developed stopping criteria to allow planning to finish when the map is deemed to be sufficiently explored. This highlights the adaptability of RIG-Tree based planners to different information objectives.

MCTS-based information gathering planners provide an effective balance between exploration and exploitation, by using the UCT (Upper Confidence Bound 1 applied to Trees)(Kocsis and Szepesvári, 2006) to select promising nodes for expansion. Chen and Liu (2019) propose a multi-objective extension of the MCTS search process to optimise over multiple, possibly competing, objectives. This planner is utilised to plan AUV paths that maximise visiting environmental hotspots, such as algal blooms. To improve the operating utility in risk-prone environments, Ayton et al. (2019) extends MCTS by adding a constraint to minimise risk in the formulated plan. These examples show the versatility of MCTS-based informative path planners.

A key objective in the adaptive sampling literature is to characterise an environmental phenomenon with minimal samples. Krause et al. (2008) investigated this problem, focusing on approximating a spatial field (such as temperature) with a given number of observations. They model the spatial field using GPs and propose both maximum entropy and mutual information criteria that can be used to optimally place sensors such that the GP uncertainty over the spatial field is minimised. This approach has

been integrated into information gathering planners (Candela et al., 2021; Hitz et al., 2017; Jadidi et al., 2019; Singh et al., 2009). There has been research into modelling the bathymetry with GPs to enable efficient navigation or mapping, for example (Wilson and Williams, 2018). However this does not model the seafloor structure which is related to the benthic habitat.

Many of the information gathering algorithms discussed so far are used to map a spatial field using a minimal trajectory. Kodgule et al. (2019) takes a different direction, by planning a ground robot's in-situ spectroscopy measurements to aid in spectral unmixing of remotely-sensed hyperspectral data. MCTS with a differential entropy reward is used to plan the robot's path, with the results indicating this planner improves the accuracy of the unmixing process for a given budget. Another objective for information gathering planners is active classification, where the planner aims to collect information that will most improve an object classifier. Candela et al. (2021) uses hyperspectral satellite imagery for benthic habitat mapping in shallow water. Their approach leverages a Variational AutoEncoder (VAE) to extract features from the hyperspectral data, with these features used as input into a neural network that performs benthic habitat classification. They compare several planning approaches to collect in-situ observations to ground truth the remotely-sensed data, including MCTS, Bayesian Optimisation (BO) and Ergodic Optimal Control. They find that although computationally intensive, MCTS plans the most informative sampling trajectory. Patten et al. (2018) use a MCTS-based planner to plan a ground robot's path that will most improve its classification of objects with an onboard laser-scanner. A GP classifier is used to classify the laser scans and the information reward is predicted mutual information from the simulated path. For an Unmanned Aerial Vehicle (UAV) tasked with collecting imagery that will be used for terrain classification, Rückin et al. (2021) links active learning with informative path planning. A probabilistic segmentation network is used to classify each pixel of the image, while the uncertainty estimate from this network is used to reward the planner for visiting areas that will most improve the underlying model. This method is effective for improving a model with minimal samples, however it needs a model to

be trained on some initial data before this strategy can be used. This highlights the need for an additional method to collect informative data before any samples have been collected. Overall, this collection of research highlights the utility of information gathering planners for finding informative paths for a wide variety of objectives.

Recent advances have been made around using Reinforcement Learning (RL) to plan informative trajectories. Viseras and Garcia (2019) use deep RL to guide multiple robots to collect informative samples of a spatial field. The reward function is designed to both maximise information gain and avoid collisions between agents. Results demonstrate improved performance using RL over GP-driven informative sampling. However RL models can be excessively noisy and difficult to debug, as the planner is a black-box model. RL methods can be useful when there is a complex and or abstract relationship between the robot and the environment, however the added complexity of these approaches is unnecessary when a more direct solution is possible.

Active learning involves proposing training samples that will most improve the underlying model. It is primarily focussed on identifying which data points from a large data pool should be labelled (Settles, 2009). There are two classes of active learning algorithms; model predictive and feature based (or density based) approaches. Model predictive techniques involve training the model on an initial seed dataset, and evaluating the model's uncertainty when querying unlabelled points. Alternatively, feature based approaches identify a set of samples that most comprehensively represents the distribution of features in the dataset. Model predictive techniques rely on having an initial seed dataset on which to train the initial model, which is often randomly selected from the data pool. For this habitat mapping application, randomly sampling the environment is inefficient due to the difficulty and cost of deploying an AUV and the spatial correlation of benthic habitats, meaning that randomly-placed transects are unlikely to sufficiently represent the survey area. Collecting the initial dataset is known as the 'cold start' problem (Gong et al., 2019). As there is no need to query a trained model, feature based approaches can be used for cold start active learning. Fujii et al. (1998) postulate that queried samples should be both similar to unlaballed data points and dissimilar to collected samples. Nguyen and Smeulders (2004) use

clustering of the feature space and querying from each cluster to ensure the collected samples match the distribution of features in the unlabelled dataset. Geifman and El-Yaniv (2017) propose Farthest-First Active (FF-Active), which uses the intermediate output of a pre-trained neural network as the feature space and the next data point to collect should be farthest in feature space from the collected samples, in order to cover the distribution of features with minimal samples. Similarly, Sener and Savarese (2018) propose a core-set method, that solving the k-centre problem to comprehensively cover the feature space of the pool data using a minimal number of samples. By using a feature-based active learning approach to select where to sample, the AUV can collect informative data while minimising distance travelled. To classify benthic images captured by an AUV, Yamada et al. (2022) observes that using training samples that are distributed across the feature space results in higher accuracy than class-balancing the training samples, highlighting the importance of comprehensively sampling from across the feature space. A limitation of feature-based active learning is that it distributes features uniformly across the feature space, whereas to most improve the model it can be more effective to focus sampling in specific areas, for example around the decision boundaries. However this application focuses on the 'cold start' problem, where sampling is being allocated before any model has been trained. For this case, using feature-based active learning to collect samples that cover the feature space is the most effective strategy given the available data.

Using bathymetry as a prior, Bender et al. (2013a) optimises the placement of an AUV survey template in the environment. He proposes modelling the distribution of features using a Gaussian Mixture Model (GMM) and then minimises the Kullback Leibler (KL) divergence between the overall distribution of features in the environment and the distribution of samples in selected survey areas. For this application, aiming to replicate the overall distribution of features could result in suboptimal surveys, as there is often an overabundance of bare habitats. Furthermore, following a set survey template limits the habitat types that can be visited in a single dive.

Foster et al. (2014) analyse the impact of different AUV survey designs on the statistical bias of habitat models, when using these models to predict the percentage

cover of species observed through the AUV imagery. From this, a set of guidelines is developed for designing AUV surveys that will likely lead to a representative survey of the area of interest. Following from this, Foster et al. (2020) develop a method for placing AUV surveys, so that they are both spatially balanced and visit specific environmental features. These surveys take into account scientists' specifications to focus on the features of interest.

There is a research opportunity for creating free-form AUV trajectories that will collect a comprehensive seed dataset for a predictive habitat model (for example in Chapter 2). Sampling-based information gathering planners provide an adaptable and exploratory framework for robotic path planning. By taking a feature-based approach to active learning, the planner can be rewarded for exploring the feature space, to generate, prior to the first deployment, a plan that is expected to collect informative samples.

3.3 Feature Extraction

Chapter 2 (Section 2.6) provides information on two feature extraction methods, geometric and learnt features, that can be used to extract meaningful features from the bathymetry. These two feature extraction methods are used to create the feature space that represents the seafloor structures found in the survey area, which is in turn correlated to the benthic habitats found in the survey area (Friedman et al., 2012).

For reference, the feature vectors for the geometric and learnt features are repeated here.

The feature vector for the geometric features at a given location is composed as follows:

$$\mathbf{z}_{\text{geometric}} = \begin{pmatrix} d & s & a_N & a_E & r_{TRI} \end{pmatrix}, \tag{3.1}$$

where d is the depth, s, a_N is the northerly aspect component, a_E is the easterly aspect component and r_{TRI} is the terrain ruggedness index (Wilson et al., 2007).

The learnt feature vector at a given location is:

$$\mathbf{z}_{encoded} = \begin{pmatrix} \mathbf{z}_{latent} & \bar{d}, \end{pmatrix} \tag{3.2}$$

where $\mathbf{z}_{latent}$ is the latent space of the autoencoder and $\bar{d}$ is the mean depth of the bathymetric patch. For an input bathymetric patch size of 11×11, $\mathbf{z}_{latent}$ is of length 16, while for a bathymetric patch size of 21×21 it is of length 32.

3.4 Informative Path Planning by Exploring the Feature Space

The framework for informative sampling of an area of interest given prior remote-sensing data is summarised in Figure 3.2. The inputs to this process are the geographical bounds of the area of interest and the corresponding remotely-sensed data (such as bathymetry) covering the area. The proposed informative planner then jointly explores the physical and feature space in order to comprehensively explore the feature space given a travel budget in physical space. This approach is designed as an improvement on traditional survey planning techniques, where surveys are often planned based on the remotely-sensed data. In both cases, the plans are generated offline before the AUV is scheduled to deploy. As the planners do not interpret the collected imagery, there is no need for them to operate online. Having the AUVs trajectory available before deployment is also important for managing AUV operations. The output of the planner is a set of waypoints that describe a piecewise linear path the AUV is tasked to follow. The AUV is then deployed and captures imagery of the seafloor. The georeferenced imagery can then be used for downstream tasks such as habitat modelling.

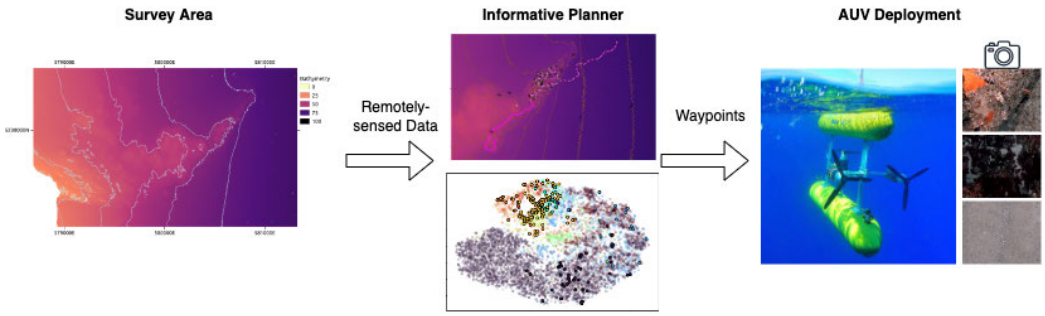


Figure 3.2 – An overview of the process for planning an informative initial survey. The inputs to this process are the survey area and corresponding remotely-sensed data. Using the remotely-sensed data, the planner designs a survey plan that approximately uniformly samples from the feature space while keeping to a total distance traveled budget in the spatial domain. The output of this process is a set of waypoints that the AUV is tasked to follow. Finally, the AUV is deployed and captures benthic seafloor imagery.

3.4.1 Problem Definition

The overarching goal is to explore the feature space, such that the AUV visits the complete extent of bathymetric features present in the bathymetry which in turn makes it more likely to observe each habitat/bottom type present in the survey area. This is done bounded by a total distance travelled constraint. Bender et al. (2013a) approaches the initial sampling by selecting the location of a template survey such that it is a 'representative survey', such that the distribution of bathymetric features of the survey should approximate the distribution of features in the environment. This approach is successful in planning initial benthic surveys, however it can lead to over-sampling of areas that are abundant in the environment. For example, for an area that is primarily composed of flat, relatively featureless terrain with smaller areas of reef, a survey planned to match this distribution would allocate the bulk of its samples to the featureless terrain.

This chapter takes a different approach, by proposing that an initial survey should visit the full range of bathymetric features present in the environment regardless of the density of these features. This is equivalent to uniformly sampling across the distribution of features. This is inspired by feature-based active learning, where the objective is to sample from the entire feature space (Sener and Savarese, 2018). A

criticism of uniform sampling is that it samples from sparse areas of the feature space, however this is a desirable quality in this application, as it avoids over-sampling of abundant areas (such as the bare, sand habitats common in marine environments) and focuses sampling on areas of diverse terrain that are likely to contain differing habitats. This approach, by definition, should yield representatives samples of the overall distribution in the sense that any point drawn from the feature space will be within a maximum distance to one of the representative samples.

The problem can be defined as selecting a subset of locations in the spatial domain to sample ($\mathbf{X}_{path}$) from the survey area ($\mathbf{X}_{area}$), such that the bathymetric features extracted at these locations ($\mathbf{Z}_{path}$) thoroughly covers the overall bathymetric feature space ($\mathbf{Z}_{area}$). Here, l is a loss function that estimates how well the feature space is explored.

$$\min_{\mathbf{X}_{path} \subset \mathbf{X}_{area}} \mathbb{E}_{\mathbf{X}_{path}}[l(\mathbf{Z}_{path}; \mathbf{Z}_{area})] \tag{3.3}$$

3.4.2 Information Reward

For the planner to sample informative areas, the reward function needs to encourage visits to features that are different to features already observed along the path so far. A reward function based on the Mahalanobis distance between features is used. The Mahalanobis distance (Mahalanobis, 1936) measures the distance between two points in multivariate space. If the variables are correlated and scaled (as is the case for an autoencoder), the Euclidean distance does not produce meaningful distances, whereas the Mahalanobis distance accounts for possible correlations and scaling differences among the dimensions of the feature vectors. This method is adaptable to any distance measure.

The Mahalanobis distance is given by:

$$D_m = \sqrt{(\mathbf{z}_a - \mathbf{z}_b)\mathrm{Cov}[Z]^{-1}(\mathbf{z}_a - \mathbf{z}_b)^T}, \tag{3.4}$$

where $\mathbf{z}_a, \mathbf{z}_b \in \mathbb{R}^n$ are the two feature points and n is the number of dimensions in the feature space. $\mathrm{Cov}[Z]^{-1}$ is the inverse covariance matrix of the data. As these

features are extracted from the remote sensing data, assumed to be available for all of the area of interest, the covariance can be estimated for the entire feature space corresponding to the area of interest.

Given that the goal is to explore the feature space, the reward function should prioritise features which are distinct from the samples collected so far. In the active learning literature, this is known as Farthest First active learning (Geifman and El-Yaniv, 2017). Therefore, the reward is calculated as the minimum feature distance to any collected feature:

$$R_m = \min_{\forall \mathbf{z}_i \in \mathbf{Z}} D_m(\mathbf{z}_a, \mathbf{z}_i), \tag{3.5}$$

where R_m is the reward, $\mathbf{z}_a$ is the feature currently being evaluated, Z is the set of features being compared against and D_m is the Mahalanobis distance.

3.4.3 Evaluation of a path

To evaluate the informativeness of a path, the coverage of the feature space should be considered. Using the feature reward (Equation 3.5) on the path is not sufficient, as this reward evaluates the diversity of features found on the path, but does not evaluate how this path approximates the survey area. Here multiple metrics are proposed to evaluate the informativeness of a path, each with a different perspective.

To assess the diversity of samples on a path, the mean of the paired feature distance between samples is used.

$$M_{PD} = \frac{1}{N^2} \sum_{\mathbf{z}_i \in \mathbf{Z}_{path}} \sum_{\mathbf{z}_j \in \mathbf{Z}_{path}} D_m(\mathbf{z}_i, \mathbf{z}_j), \tag{3.6}$$

where $\mathbf{z}_i$ and $\mathbf{z}_j$ are a pair of features collected along the candidate path ($\mathbf{Z}_{path}$), N is the number of features on the path and D_m is the Mahalanobis distance outlined in Equation 3.4. Although this metric captures the diversity of samples observed along the path, it does not capture how well the path characterises the environment. For example, a path that visits a diverse range of terrain but does not visit a large area of

similar terrain (e.g. a bare, sandy habitat) will likely score highly using this method, even though it fails to characterise a large area of the environment.

To ensure the evaluation function is capturing how well the path characterises an environment, the path should minimise the feature distance from the features found in the environment and one of the points on the path. As there is a very large number of pixels (on the scale of millions) in the remotely-sensed data, to reduce computational burden, the environment's features are approximated by spatially sampling a large set of points and extracting the feature vector at each of these points. The following metric finds the closest feature distance from features randomly extracted from the environment, to one of the features collected on the path.

$$M_{SD} = \sum_{\mathbf{z}_i \in \mathbf{Z}_{space}} R_m(\mathbf{z}_i; \mathbf{Z}_{path}), \tag{3.7}$$

where $\mathbf{z}_i$ is one of the features randomly extracted from the environment ($\mathbf{Z}_{space}$) and $\mathbf{Z}_{path}$ is the set of features extracted from the path. R_m (Equation 3.5) finds the closest feature distance between the given feature ($\mathbf{z}_i$) and the set of reference features ($\mathbf{Z}_{path}$). While this method maximises the area that is characterised by the path, it is biased towards large spatial areas of a similar seafloor structure. Using this metric, a path that only visits a large area of similar terrain (e.g. a large sand area) will score higher than a path that visits a range of terrain but does not explore as much of the similar terrain. Therefore this metric is not used in the evaluation of the surveys.

To reduce the bias towards large similar regions in Equation 3.7, this research proposes the following evaluation function M_K. First, the feature space is clustered using the K-means algorithm to select K centre points ($\mathbf{Z}_K$) that represent the feature space. K is selected by visually analysing the cluster maps (see Section 2.7) with different numbers of clusters, with an appropriate number being where the feature space forms many distinct regions around the bathymetric features without becoming noisy. The evaluation metric is the sum of the closest feature distances for each of the K centre

points to the features visited along the path.

$$M_K = \sum_{\mathbf{z}_k \in \mathbf{Z}_K} R_m(\mathbf{z}_k; \mathbf{Z}_{path}), \tag{3.8}$$

where $\mathbf{z}_k$ is one of the centres ($\mathbf{Z}_K$) that represent the latent space, $\mathbf{Z}_{path}$ represents the features extracted from the path being evaluated and R_m is the reward function presented in Equation 3.5. This metric and the transition from physical to feature space is displayed in Figure 3.3.

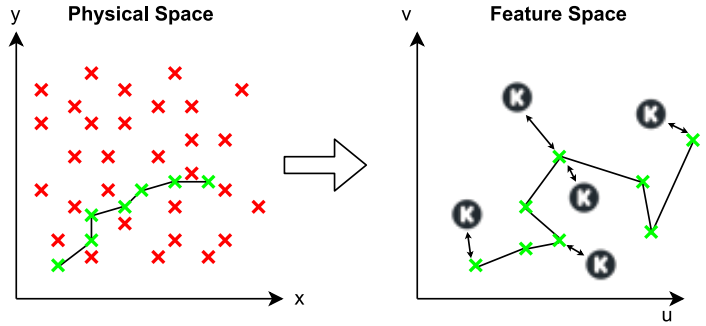


Figure 3.3 – Visualising the process for evaluating a path. On the left it shows the physical (spatial) space and the robot's path. The green crosses are points sampled on the robots path, while the red crosses are points randomly sampled from the survey area. The right plot shows the robot's path in feature space. The red crosses are clustered in feature space to form the k centres. For each of the k centres, the distance to the closest feature from the path is found. The total evaluation score is the sum of all these distances.

3.4.4 RRT-based Informative Path Planner

The algorithm presented here is a planner designed to use the information reward (proposed in Section 3.4.2) to explore the feature space. It is primarily based on RIG-Tree (Hollinger and Sukhatme, 2014), which is in turn based on RRT*, which promotes exploration of the entire environment. This algorithm incrementally grows a tree by expanding branches towards high value areas. Branches of the tree can be expanded until they reach the distance budget. For this application, the ability to query the entire survey area of the robot, rather than just within the sensing radius makes the *aim* (outlined below) functionality very effective. The algorithm is

summarised in Algorithm 1 and illustrated in Figure 3.4. The key extensions to RIG-Tree / RRT* are the use of the novel reward function outlined in Equation 3.5, that allows for incremental building of informative trajectories and consideration of the entire trajectory in the information reward. A novel interpretation of the *aim* function is developed to direct the tree towards areas that have unique features compared to those already collected by the tree. Furthermore, a method to establish the starting position of the tree is proposed that considers the informativeness of the region.

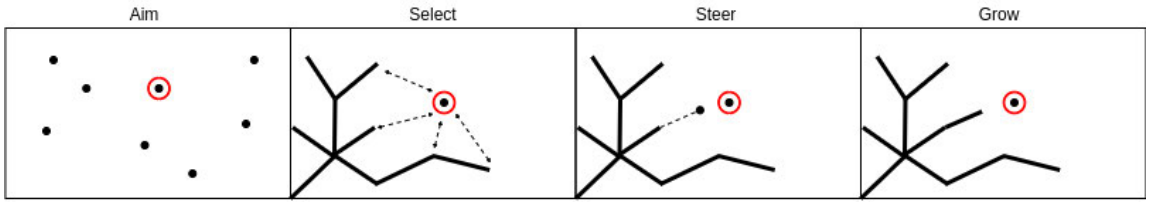


Figure 3.4 – InfoRRT Process. First, the planner selects a target point from the larger survey area. This target point is selected either at random or is the highest reward point from many candidate target points. Next, nodes on the search tree that are closest to the target point are evaluated for expansion. This node is selected such that it will be the highest value, when combined with the target point. A new node is created by steering the node in the direction of the target point, and finally this node is added to the search tree.

Find Starting Position

The starting position has a significant impact on the overall path, especially if the path budget is small compared to the size of the survey area. The AUV will not be sampling efficiently if it has to traverse a significant distance to get to informative areas. Therefore, the starting location should be chosen so it places the AUV in an informative region. To estimate the informative starting region, random points are sampled from the environment and features are extracted at each of these points. A random subset of these points is selected as potential starting positions.

For each of these potential starting positions, all the points in its vicinity are used to estimate the region's value. The M_K metric is used to estimate the informativeness, which is calculated as the sum of the minimum feature distances from any one of the points in the vicinity, to K-centres distributed throughout the feature space. This method is successful in selecting informative starting positions and typically selects

the locations around the interface between bathymetric terrain, for example between reef and sand.

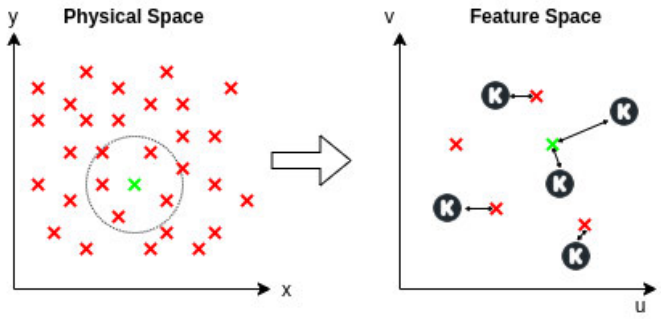


Figure 3.5 – Selecting informative starting positions. Features are extracted from points within a search radius of each candidate starting position (green). The value of the starting position is the sum of the feature distance between all K-centres that approximate the latent space and the closest feature.

Aim

Select a target point either at random or towards a goal point, as is the case in RRT. A goal point is found by first sampling spatial points randomly sampled from the environment. For each of these points, the reward is calculated between the point and all the other points currently in the tree. The point with the maximum reward is selected as the goal point. The objective is to direct the tree towards informative areas.

$$\mathbf{x}_{target} = \underset{\mathbf{x}_i \in \mathbf{X}_{space}}{\arg\max}[R_m(\mathbf{z}_i; \mathbf{Z}_{tree})], \tag{3.9}$$

where $\mathbf{X}_{space}$ are locations randomly sampled from the search space, $\mathbf{z}_i$ are the features sampled from the location $\mathbf{x}_i$ and $\mathbf{Z}_{tree}$ is the set of features already collected from the tree.

Select

Given the target point, find the nearest (spatially) k nodes on the tree. For each of these nodes, assess the reward gained if the tree was to expand to this target point. The planner selects the node that has the maximum reward and that will not exceed the maximum distance budget.

$$v = \underset{v_i \in V_{nearest}}{\arg\max} \left[R_m(\mathbf{z}_{target}; \mathbf{Z}_{path}) \right], \tag{3.10}$$

where $V_{nearest}$ is the k nearest nodes to the target point, $\mathbf{x}_{target}$. $\mathbf{z}_{target}$ are the features extracted from the point $\mathbf{x}_{target}$ and $\mathbf{Z}_{path}$ are the features collected on the path to node v_i.

Steer

Extend the selected node towards the target point, while checking for collision with any obstacles. For this application, these obstacles are usually areas that are either too shallow or deep for the AUV to operate. Constraints on the vehicle dynamics can also be added during this step, for example to prevent the vehicle attempting to travel over steep slopes.

Grow

Create a new node at this new point and add it to the tree.

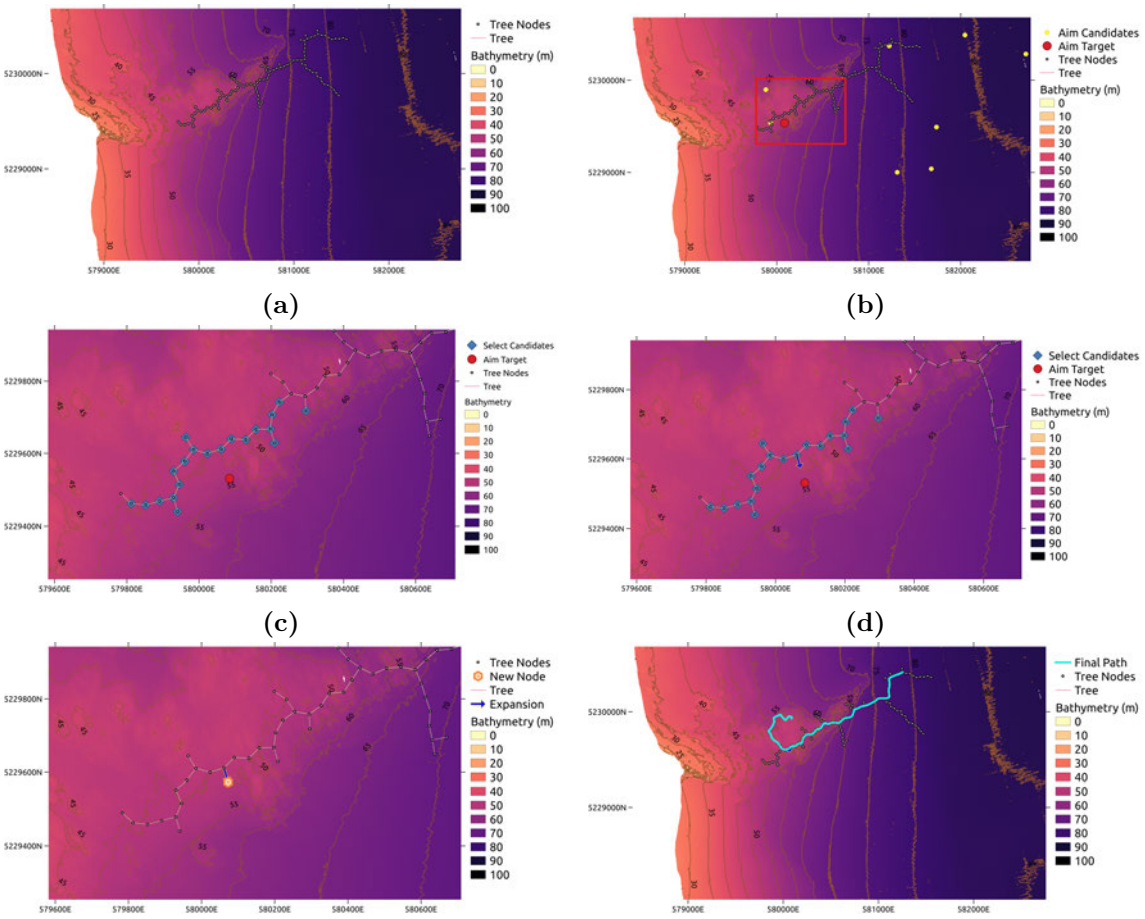


Figure 3.6 – An iteration of informative planning using *RRT*. (a) shows the tree before expansion. (b) details the *aim* process, where several candidates from the search space are selected. A target node is selected that has the maximum feature wise distance to nodes already in the tree. (c) details the *select* process, where candidate nodes from the tree are evaluated for expansion. (d) outlines the *expand* process, where the tree is expanded from the selected node towards the target node. (e) shows the new node as part of the expanded tree. (f) shows the final path (that does not include the branch added in the previous steps).

Algorithm 1 InfoRRT

function AIM
  if Aim Towards Informative Area **then**
    $x_{target} \leftarrow \arg\max_{x_i \in X_{space}} [R_m(z_i; \bar{Z}_{tree})]$ ▷ Select the location that maximises the feature distance to points already in the tree
  else
    $x_{target} \leftarrow RandomLocation()$
  end if
  return x_{target}
end function

function SELECT(x_{target})
  $V_{nearest} \leftarrow$ NearestKNodes(x_{target}, k)
  $v_{selected} \leftarrow \arg\max_{v_c \in V_{nearest}} [Rm(z_{target}, Z_{path})]$ ▷ Select the node to expand that maximises the information reward
  return $v_{selected}$
end function

function STEER(v, x_{target})
  $x_{new} \leftarrow ExtendTowards(x_v, x_{target}, d_{expand})$ ▷ Extend towards the target location by the expansion distance
  if CheckConstraints **then**
    return True, x_{new}
  else
    return False, x_{new}
  end if
end function

function GROW(x_{new})
  $R \leftarrow R_m(z_i, Z_{path})$
  $c \leftarrow c + d_{expand}$
  $v_{new} \leftarrow$ Node(x_{new}, R, c)
end function
procedure INFORRT
  FindStartingPosition() ▷ Selects a starting position
  while Computating Budget Remains **do**
    $x_{target} \leftarrow$ Aim() ▷ Selects a target point to aim towards
    $v \leftarrow$ Select(x_{target}) ▷ Selects a node to expand towards the target point
    valid, $x_{new} \leftarrow$ Steer(v, x_{target}) ▷ Expands the selected node towards the target point
    if valid **then**
      Grow(x_{new}) ▷ Creates a new node and adds it to the tree
    end if
  end while
end procedure

3.4.5 MCTS-based Informative Path Planner

Monte Carlo Tree Search (MCTS) is a planning method that incrementally builds a search tree of states by taking random samples in the decision space (Browne et al., 2012). At each iteration, MCTS attempts to select the optimal node (potential waypoint) for expansion, based on either exploration (searching in areas that are undersampled) or exploitation (searching in areas that have a higher reward). The value of an action is estimated by performing simulations of further AUV trajectories given this action, using a default policy.

MCTS provides a solid base for an information gathering planner as it is capable of finding an acceptable solution in large decision spaces and effectively balances exploration and exploitation. This research extends MCTS to explore the feature space by guiding the planner using the reward function outlined in Equation 3.5. Furthermore, natural exploration is promoted by occasionally aiming towards high value areas, by incorporating the *aim* functionality from RRT*. Many MCTS implementations operate with discrete actions where the possible actions from a specific node are predefined, such as in game-playing (Go, chess etc.) or where a continuous space is divided into a grid. This implementation of MCTS uses continuous spaces to build its search tree. It manages the increased complexity by limiting the number of expansions per node and also preventing a random action from a specific node being too similar to a previous action. Operating on continuous spaces allows the planner to better fit the environment while also allowing easier integration of robot constraints.

3.4.6 MCTS Process

Find Starting Position

The method for finding the start position is identical to that use in the *RRT* method, which finds an informative region in which to expand the tree.

Selection

MCTS aims to expand nodes based on a 'best first' policy. The node to expand is

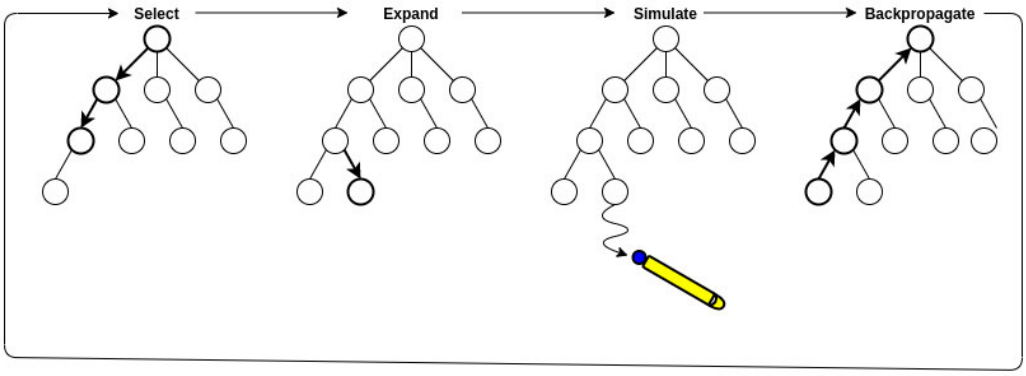


Figure 3.7 – The four stages of MCTS. First, a node is selected for expansion. Next, the selected node is expanded according to the expansion policy, which is to aim towards high value areas. From this expanded node, further actions are simulated to provide an estimate of the node's expected future value. Finally, this value estimate is backpropagated up the tree, updating the value of each node that led to this action.

based on the UCB-1 algorithm (Kocsis and Szepesvári, 2006) that is found to balance exploration and exploitation.

$$v = \underset{v_c \in v_{children}}{\arg\max} \left[Q_v + C_{uct} \sqrt{\frac{\ln N_v}{N_{v_c}}} \right], \tag{3.11}$$

where v is the node evaluated, Q_v is the expected future reward given from visiting node v, C_{uct} is the exploration vs exploitation constant, here with a value of $\sqrt{2}$ as determined by (Kocsis and Szepesvári, 2006) to effectively balance exploration and exploitation. N_v is the number of times the parent node has been visited and N_{v_c} is the number of times the child node has been visited.

Expansion

Inspired by our RRT approach, the selected node is expanded either randomly or towards high value areas. Both types of node expansion have to satisfy constraints, including the direction change being within an acceptable range to avoid excessive jitter (i.e. turning) in the path.

The expand function can be used to direct the AUV towards informative areas. To encourage exploration, the informative areas are identified as areas that are different to features already in the tree. The expand function selects the location that

maximises the minimum feature distance to all nodes currently in the tree.

$$\mathbf{x}_{new} = \underset{\mathbf{x}_i \in \mathbf{X}_{space}}{\arg\max}[R_m(\mathbf{z}_i; \mathbf{Z}_{tree})], \tag{3.12}$$

where $\mathbf{x}_{new}$ is the new location of the expanded node, $\mathbf{x}_i$ is a location from $\mathbf{X}_{space}$ that has been randomly sampled from the environment, $\mathbf{z}_i$ is the feature extracted from the remotely-sensed data at point $\mathbf{x}_i$ and $\mathbf{Z}_{tree}$ is the set of features collected in the entire search tree.

During this process, the path is checked to ensure there are no obstacles. If an obstacle does exist, the expansion process is skipped and the planner begins the selection process again.

Simulation

In the simulation phase, the value of an action is estimated by performing multiple simulations of future actions. For this application, AUV paths are generated until the distance budget is exceeded or until the simulated path hits an obstacle. The simulation follows the same action policy as the expansion step. The simulation can be run multiple times to provide a better estimate of the expected future rewards. The value for the simulation becomes:

$$r_{sim} = \frac{1}{N_s} \sum_{i=0}^{N_s} \sum_{\mathbf{z}_j \in \mathbf{Z}_{path}} R_m(\mathbf{z}_j; \mathbf{Z}_{path}), \tag{3.13}$$

where N_s is the number of simulations run, $\mathbf{Z}_{path}$ is the set of features collected during each simulation run.

Backpropagation

After a simulation is performed, this evaluation is then used to update the search tree. Starting from the recently expanded node and moving up the tree until it reaches the root node, each node value is re-evaluated. As there are incremental rewards for subsequent nodes, the valuation function from Patten et al. (2018) is used:

$$Q_v = \eta^{\tau} R_v + \frac{1}{N_v} \left(r_v + \sum_{v_c \in v_{children}} N_{v_c} Q_{v_c} \right), \tag{3.14}$$

where Q_v is the estimated value of the node, η is the discount factor, τ is the depth of the current node, R_V is the immediate reward for visiting the node, N_v is the number of visits to the current node, r_v is the simulation reward for the current node, v_c is a child node of the node currently being evaluated, N_{v_c} is the number of visits to the child node and Q_{v_c} is the estimated value of the child node.

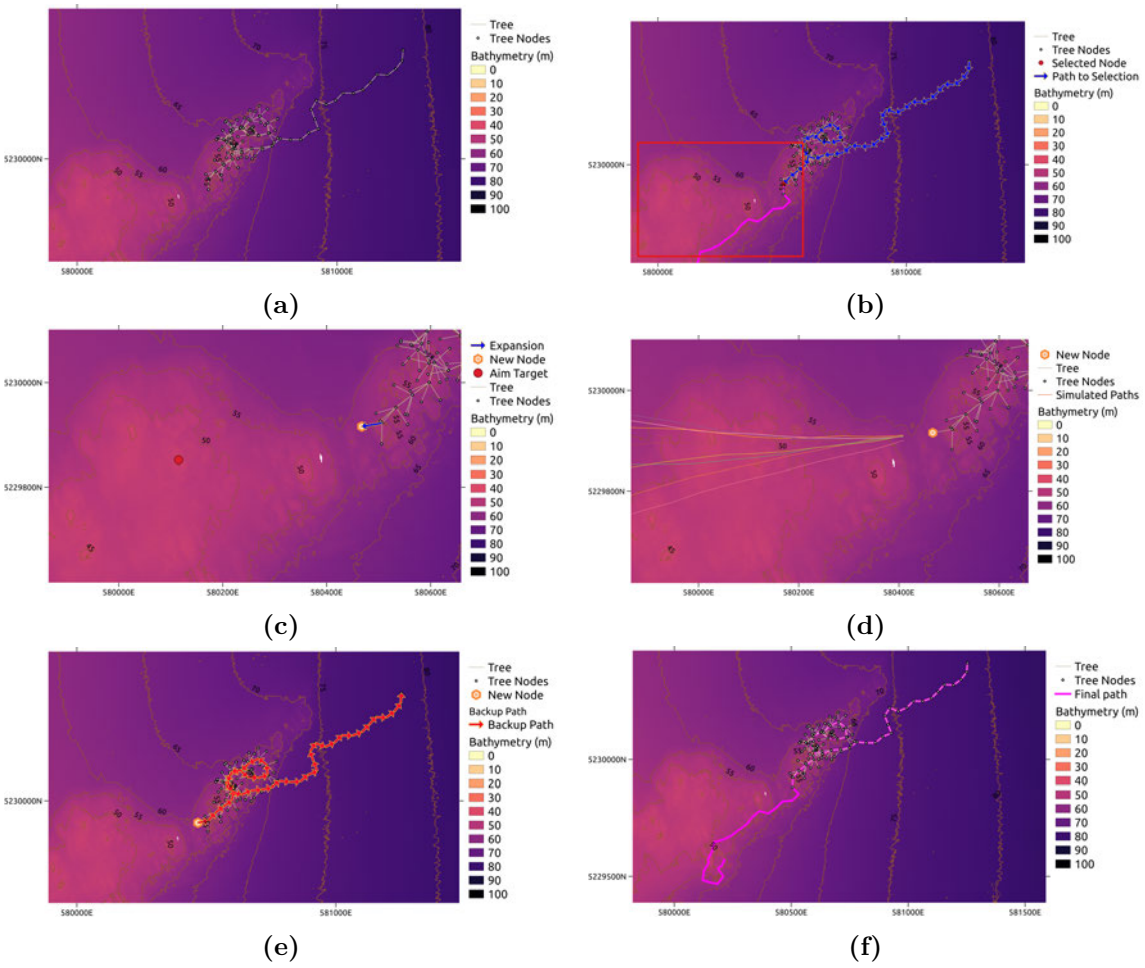


Figure 3.8 – An iteration of informative planning using *MCTS*. (a) shows the tree before expansion. (b) highlights the selection process, where a node is selected for expansion based on the UCB-1 algorithm (Browne et al., 2012) (c) details the expansion process, where the tree is expanded from the selected node, to an informative target (d) shows the simulated future rollouts, (e) shows the backpropagation process, where the value of the tree is updated from the simulation result. (f) shows the final path.

Algorithm 2 InfoMCTS

function SELECT
  $v \leftarrow RootNode()$
  while NotTerminal **do** ▷ Starting at the root node, select a node for expansion
    $v \leftarrow \arg\max_{v_c \in v_{children}} [Q_v + C_{uct}\sqrt{\frac{logN}{N_{v_c}}}]$
  end while
  return v
end function

function ACT(v)
  if Aim Towards Informative Area **then**
    $X_{space} \leftarrow ManyRandomLocations()$
    $x_{target} \leftarrow \text{argmax}_{x_i \in X_{space}} [R_m(z_i; Z_{tree})]$ ▷ Select a target location that maximises the feature-wise distance compared to features already in the search tree
    $x_{new} \leftarrow ExtendTowards(x_v, x_{target}, d_{expand})$ ▷ Extend towards the target location by the expansion distance
  else
    $x_{new} \leftarrow RandomlyExtendPath()$ ▷ Extend the path by the expansion distance
  end if
  return x_{new}
end function

function EXPAND(v)
  $x_{new} \leftarrow \text{Act}(v)$
  $R \leftarrow R_m(z_i; Z_{path})$
  $c \leftarrow c + d_{expand}$
  $v_{new} \leftarrow Node(x_{new}, R, c)$ ▷ Create a new node, with a new state, reward and cost
  return v_{new}
end function

function SIMULATE(v)
  for $s = 1, 2, \ldots, \text{Simulations}$ **do**
    $c \leftarrow Cost(v)$
    $x \leftarrow State(x)$
    while $c < DistanceBudget$ **do** ▷ Simulate until budget is exhausted
      $x \leftarrow Act(v)$
      $c \leftarrow c + d_{expand}$
      $r \leftarrow r + R_m(z; Z_{path})$
    end while
  end for
end function

39:
40: **function** BACKPROPAGATION(v)
41: **while** v is not v_{root} **do** ▷ Move back up the tree, updating the expected reward
42: $Q_v \leftarrow \eta^\tau R_v + \frac{1}{N_v}(r_v + \sum_{v_c \in v_{children}} W_{v_c} Q_{v_c})$
43: $v \leftarrow v_p$
44: **end while**
45: **end function**
46:
47: **procedure** INFOMCTS
48: $x_{start} \leftarrow FindStartingPosition()$ ▷ Selects a starting position
49: $v \leftarrow Node(x_{start}, 0, 0)$ ▷ Creates the starting node, with zero reward and cost
50: **while** **do**
51: $v \leftarrow Select()$ ▷ Select a node for expansion
52: $v \leftarrow Expand(v)$ ▷ Expand the selected node
53: $Simulate(v)$ ▷ Simulate further actions from this node
54: $Backpropagation(v)$ ▷ Adjust the tree values
55: **end while**
56: **end procedure**

3.4.7 Informative Survey Template Placement

Survey templates are routinely used for planning benthic surveys. A survey template is beneficial as it is easily interpreted by human operators and guarantees a spatially balanced survey (Foster et al., 2014). However, the set template may not correspond with the interesting areas of the terrain. Survey templates are traditionally planned manually to cover areas of interest. Alternatively, these templates can be informatively placed to ensure the template maximises its exploration of the feature space. This approach has been pioneered by Bender et al. (2013a), who optimised the placement of a pre-defined survey shape (i.e. template) so the distribution of features on the survey matched the distribution of features found in the overall survey area. An informative placement method is proposed that evaluates many different candidate placements with the evaluation function outlined in Equation 3.8, in order to

uniformly sample the feature space of the survey area. This can be defined as:

$$\mathbf{X}_{path} = \underset{\mathbf{X}_{path} \in \text{RandomPlacements}}{\arg\min} [M_K(\mathbf{Z}_{path}; \mathbf{Z}_{area})], \tag{3.15}$$

where $\mathbf{X}_{path}$ is a candidate survey with features $\mathbf{Z}_{path}$, that is evaluated with the metric M_K using the reference features for the entire survey area ($\mathbf{Z}_{area}$).

The starting location, orientation and type of survey template are the degrees of freedom when placing a survey. There is an endless array of survey templates to choose from. In this chapter, a broad-grid consisting of five equal segments is used, with each of the sides being one-fifth of the total distance budget in length. This is designed to satisfy the guidelines for spatially-balanced surveys presented in Foster et al. (2014). The informative placements of surveys is displayed in Figure 3.9, showing a selection of randomly placed surveys and the most informative survey presented in green.

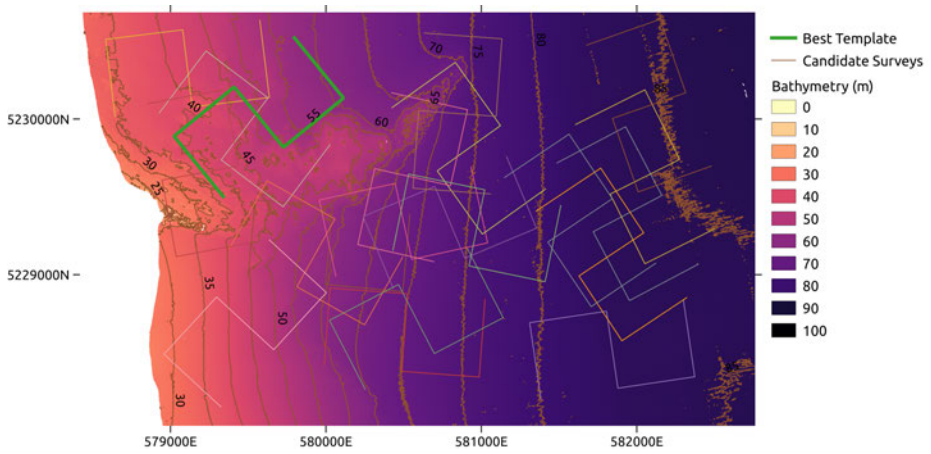


Figure 3.9 – Shows the informative placement of survey templates. Many surveys are randomly placed in the survey area, by altering the location and orientation. The thin lines are a subselection of the candidate surveys. Here 20 surveys are shown, while in the planning process at least 1,000 surveys are used. The green indicates the most informative survey. Coordinates are displayed in UTM zone 55S.

3.4.8 Cluster-TSP

This method proposes a set of points that together represent the feature space and plans a path that visits these points. The bathymetry for the entire planning space is

first projected into the latent space by the bathymetry encoder. The corresponding features (i.e. latent space coordinates) are clustered using the Gaussian Mixture Models (GMM) clustering algorithm (Reynolds, 2009). GMMs fit a given number of Gaussians to the training data using the Expectation-Maximisation algorithm. The GMM is used to assign a cluster label to each of the bathymetry patches, which can then be visualised is the spatial domain, where the pixel value is the cluster assigned at the spatial coordinates for that bathymetric patch. The clusters in feature space tend to form several disjoint patches in the spatial domain. Each spatial cluster grouping in the spatial domain is represented as a polygon.

Representative points for each cluster in the latent space are randomly sampled from each corresponding cluster polygon in the spatial domain. To explore the feature space, at least one of each cluster type should be visited in the spatial domain. Typically there will be several cluster polygon choices to visit for each cluster type. This problem can be framed as a set-TSP (Travelling Salesman Problem). Set-TSP is a generalisation of the travelling salesman problem, where the goal is to visit one of each type of node. To solve the Set-TSP problem the OR Tools library (Perron and Furnon, 2019) is leveraged, which uses hill-climbing or simulated annealing to optimise the routing between the nodes, where the nodes are points selected from each spatial cluster grouping.

Both geometric and encoded features can be used with this method, however the clusters generated when using geometric features are excessively noisy in the spatial domain and do not form well defined spatial clusters. Hence only the encoded features are used for this method.

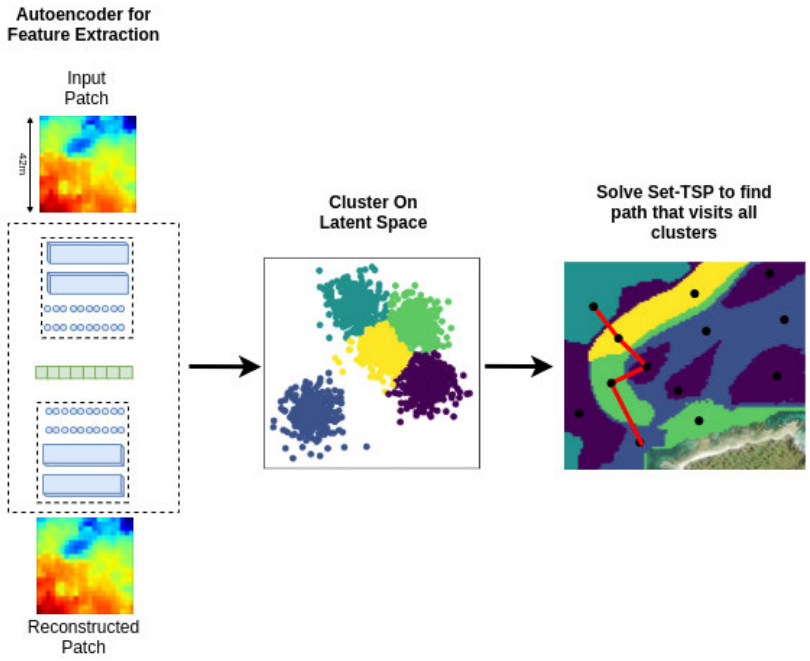
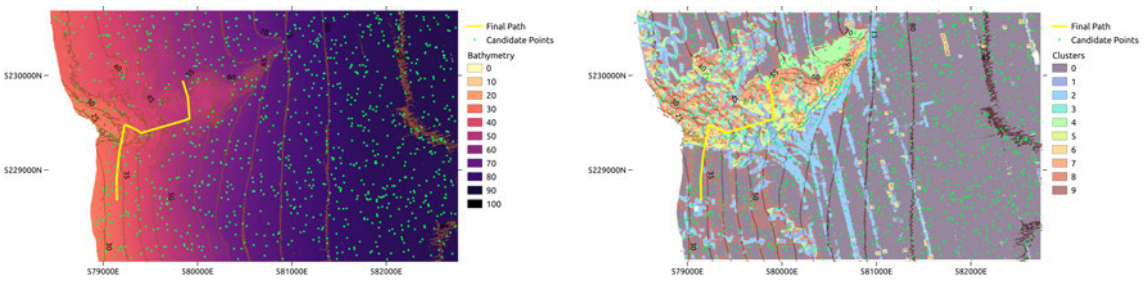


Figure 3.10 – Process diagram for the Cluster-TSP method. Bathymetric patches of the entire survey area are extracted from the raster and projected into the latent space of the autoencoder. Clustering is then run on this feature space. Finally, a set-TSP problem is solved to find an AUV path that visits each cluster type and hence explores the feature space.



(a) **(b)**

Figure 3.11 – Placement of nodes and the final path on the O'Hara dataset (see Section 3.5). Nodes are distributed uniformly across the survey area. The final path focusses on cluster groups in and around O'Hara reef to minimise the path length.

3.5 Results

This section presents results for the proposed feature exploration framework. It is demonstrated on three datasets, all with unique sources of bathymetry and scales,

highlighting the adaptability of this approach. The *O'Hara* dataset (Figure 3.13) focuses on O'Hara Reef near Fortesque in Tasmania, Australia. It consists of ship-borne bathymetry gridded at 2m collected by GeoScience Australia (Spinoccia, 2011). Figure 3.12 shows AUV imagery collected in this region in 2008 and the habitat classes overlaid onto the bathymetry and clusters. This provides context into the likely habitats found in this region. A well planned survey should visit all these habitat classes, or at least as many as can be differentiated using the bathymetry. The *Vincentia* dataset (Figure 3.14) is focused on the area off Vincentia, Jervis Bay, Australia, with the bathymetry derived from LiDAR and gridded at 5m (Environment and of Planning Industry and, 2018). For the *Trimodal* dataset (Figure 3.15), an AUV collected full coverage imagery of a section of Trimodal Reef, Lizard Island, Australia. The bathymetry is extracted from the photogrammetric reconstruction at 0.1m. Furthermore, three classes of benthic habitats; sand, rubble and reef, have been coarsely labelled on the mosaic, to demonstrate how the proposed planners visit all the habitat classes.

To benchmark these proposed planners, random transects are placed across the environment, with the length of these transects being equal to the budget set for the planners. The number of transects is set to 100 to ensure the environment is adequately sampled. The benchmark, L_B is the mean value of each of the validation criteria across all the transects. This benchmark is expected to perform well on the datasets selected, as the bathymetry is cropped to the area of interest so that any transect placed across the environment is likely to visit an interesting area.

Each path is evaluated using the criteria established in Section 3.4.3. M_{PD} captures the diversity of features of the path, which should be maximised. M_K uses features randomly sampled from the environment to measure how well the path characterises the environment. It summarises the feature space using clustering to prevent bias towards large spatial regions. Finally, M_C measures the ratio of bathymetric cluster types visited, compared to the total number of cluster types:

$$M_C = \frac{\text{Clusters Visited}}{\text{Total Clusters}} \tag{3.16}$$

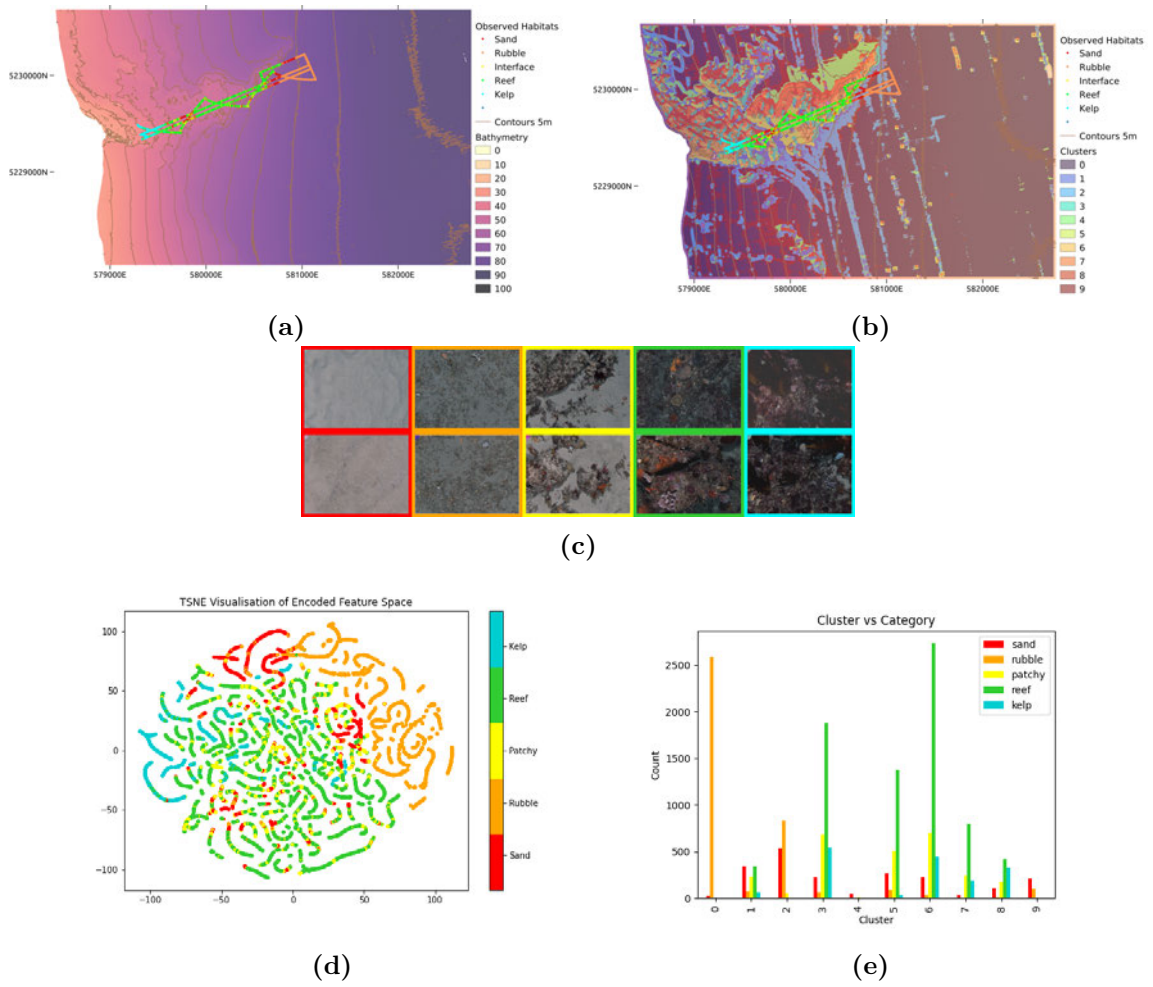


Figure 3.12 – Habitat labels from AUV imagery collected at O'Hara reef, Tasmania in 2008. (a) shows the bathymetry with the labelled AUV dive. (b) is the same labelled dive overlayed onto the bathymetric clusters. The colours along the AUV path on the map correspond to the colours of the frames on the example images in (c). This shows the distribution of benthic habitats in the area. (d) 2D t-SNE (Van Der Maaten and Hinton, 2008) projection of the bathymetric feature space, color coded with the habitat labels, which shows that similar areas of bathymetric terrain are likely to be similar benthic habitats. (e) the counts of each habitat class within each bathymetric cluster. Coordinates are displayed in UTM zone 55S.

The budget for each dataset is set according to an initial pass by the Cluster-TSP method. This is used to set the budget as the feature-space ca be regarded as sufficiently explored when all the clusters have been visited. The budget is set at 2500m for the *O'Hara* dataset, 2500m for the *Vincentia* dataset and 60m for the *Trimodal* dataset, given its smaller size.

The parameters used for each planner were empirically selected to achieve the best performance. For each run, the *MCTS* and *RRT* planners start three times, each for 10,000 iterations, with the best path automatically selected. The multiple starts helps prevent the planners proposing a poor path. The expansion distance for these methods was set relative to the resolution of the bathymetry and to the overall budget. For the *Templates* method, 1,000 candidate paths were evaluated.

Results are presented for planning using the geometric features in Table 3.1 and for encoded features in Table 3.2. As each method is stochastic, each experiment is run ten times. The starting positions for the *MCTS* and *RRT* are kept the same for each run, and are selected using the method for starting in informative regions, outlined in Section 3.4.4.

The survey planners that utilise the remotely-sensed data (*MCTS*, *RRT*, *TSP*, *Templates*) significantly outperform random transects over the area of interest (L_B). As expected, L_B performs relatively well as the operational area is cropped around the areas of interest. As the operational area becomes larger, randomly placing transects (L_B) is expected to perform poorly as a random transect will be less likely to intersect interesting areas of the feature space.

The ability for the set of planners to propose informative paths for both the geometric and encoded feature representations highlights the generalisation of the approach to different feature representations. Both feature representations direct the AUV to similar informative areas; relatively flat areas that are likely to be sand, more rugose areas that are commonly reef, and interface areas that exist between the habitat types. This is expected as both the representations are capturing the geometric features of the bathymetry, albeit at different resolutions. However the performance of this approach for higher-dimensional feature representations may suffer as it relies upon the feature distance, that is less reliable in higher dimensions due to sparsity (Aggarwal et al., 2001). Similar samples can still have large feature distances between them which may not represent large underlying differences in benthic habitat. This can be observed in the paths planned with the encoded features, which focus too much on the highly rugose areas of the reef, where there can be large feature distances

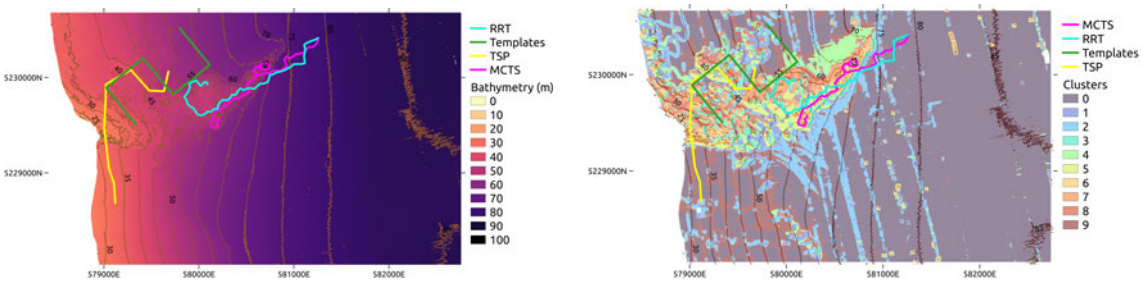
between similar and co-located samples. This leads to oversampling of these areas at the expense of others areas. Therefore, care should be taken to construct a feature space that is both compact and descriptive in representing the important features of the underlying data. This may limit this approach from being used for planning informative surveys using highly-dimensional remote data such as satellite imagery.

Both the freeform planners, *MCTS* and *RRT* are able to comprehensively explore the feature space. These planners choose where next to sample in order to maximise the distance from previously collected samples, which effectively maximises the metric M_{PD}. This does not directly optimise approximating the feature space of the target area. It is too computationally intensive to directly optimise the M_K method for these planners. However when planning with geometric features, the strong performance on the M_K metric indicates that the approach of selecting samples that maximise the distance from existing samples leads to a thorough coverage of the feature space. The performance on the M_K metric is lessened when planning with encoded features, due to the higher dimensions of the encoded feature space. This increased sparsity means that moderately different samples can have large feature distances between them, and the freeform planners will focus sampling on these areas. On the *O'Hara* and *Trimodal* datasets, the freeform planners consistently visit most or all clusters present in the dataset with M_C over 0.96 (an average 9.6 out of 10 clusters visited on each run), which is a key indication of an informative initial survey. Example paths can be seen in Figures 3.13, 3.14, 3.15.

The *Templates* method, which involves placing a set survey template using the information metrics, scores highly across all the datasets presented, highlighting the benefits of utilising the bathymetry for positioning the survey template. For these experiments, a 2D broad-grid is considered that is shown to be effective using the guidelines developed in Foster et al. (2014). In practice, using a set survey template can limit performance, if the template cannot be aligned to interesting features in the environment. For example a square-grid would not be well suited to the long reef in the *O'Hara* dataset. Evaluating multiple candidate surveys could improve results, providing a greater chance a template will be well matched to the target area. The

current method is a brute-force method, where many randomly-placed candidate surveys are evaluated. Dynamically fitting survey templates to informative points could lead to surveys that provide more complete coverage of the feature space. Similarly, a secondary local optimisation of the survey template position could further increase the information collected.

The *TSP* method is designed to visit all of the clusters, hence scoring highly on the M_C metric. This method also performs well on the M_K metric, however the feature distance methods (*RRT*,*MCTS*,*Templates*) generally score better, highlighting the importance of exploring the feature space within a cluster. *TSP* is an effective method for initial feature space exploration, however its reliance on clustering and lack of flexibility hinder its ability for planning benthic surveys. For some datasets, clustering can lead to suboptimal clusters, with clusters forming that are either too large or small, which will impact the ability of the planner to explore the latent space. Artefacts in the bathymetry, such as the track lines present in the *O'Hara* dataset (Figure 3.13b), can lead to clusters that are not reflective of the habitat. Finally, the *TSP* method does not have a pathway to continue planning beyond the initial survey, while the other methods can continue planning indefinitely.



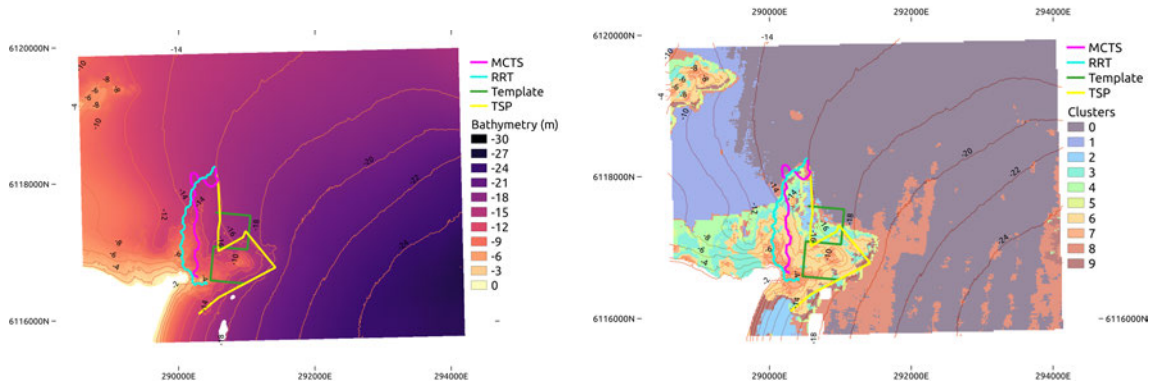
(a) **(b)**

Figure 3.13 – Planned paths using encoded features for O'Hara Reef, Fortesque, Tasmania, Australia overlaid onto the bathymetry (a) and clusters (b). The budget for each path is 2500m. The informed paths are able to collect features representative of the entire area by focusing on the areas around the O'Hara Reef. Coordinates are displayed in UTM zone 55S.

Noise or artefacts in the remotely-sensed data can lead to incorrect estimates of the seafloor structure. For acoustic bathymetry, these artefacts are either from sensor noise or errors in the localisation of the sensor. This is evident in the bathymetry for

		MCTS	RRT	Templates	TSP*	L_B
O'Hara	M_{PD}	**2.04 ± 0.34**	1.75 ± 0.26	1.54 ± 0.18	1.54 ± 0.17	1.23 ± 0.02
	M_K	1.20 ± 0.00	**1.19 ± 0.00**	1.20 ± 0.01	1.31 ± 0.00	1.53 ± 0.00
	M_C*	0.96 ± 0.10	**0.98 ± 0.06**	0.97 ± 0.05	0.97 ± 0.07	0.73 ± 0.01
	D (m)	2481 ± 12.07	2418 ± 150.57	2500 ± 0.00	1881 ± 361.22	2500 ± 0.00
Vinc.	M_{PD}	1.57 ± 0.46	**2.25 ± 0.27**	1.55 ± 0.44	1.34 ± 0.25	0.86 ± 0.04
	M_K	1.31 ± 0.01	**1.31 ± 0.01**	1.34 ± 0.01	1.32 ± 0.02	1.72 ± 0.00
	M_C*	0.87 ± 0.09	0.79 ± 0.07	0.79 ± 0.12	**0.99 ± 0.03**	0.38 ± 0.03
	D (m)	2485 ± 13.00	2309 ± 278.60	2500 ± 0.01	3150 ± 730.10	2500 ± 0.00
Trim.	M_{PD}	1.24 ± 0.08	**1.61 ± 0.26**	1.40 ± 0.15	1.40 ± 0.12	1.20 ± 0.02
	M_K	**0.55 ± 0.00**	0.61 ± 0.00	0.56 ± 0.00	0.62 ± 0.00	0.61 ± 0.00
	M_C*	0.90 ± 0.08	0.98 ± 0.04	0.90 ± 0.05	**0.99 ± 0.03**	0.85 ± 0.01
	D (m)	59 ± 0.43	48 ± 9.12	60 ± 0.00	51 ± 7.71	60 ± 0.00

Table 3.1 – Results for exploring the feature space, where the feature space is composed of the geometric features (Section 2.6.1). ± refers to plus or minus one standard deviation. Each value is the mean value over ten runs. Vincentia is abbreviated to Vinc and Trimodal is abbreviated to Trim. The blue columns indicate paths planned using the information metric proposed in Section 3.4.2, which are compared to Cluster-TSP in the yellow column (Section 3.4.8) and random transects over the area (L_B) in the orange column. M_{PD} and M_C should be maximised, while M_K should be maximised. As the clustering process does not produce usable clusters with geometric features, * indicates methods and metrics that use encoded features.



(a) **(b)**

Figure 3.14 – Planned paths using encoded features for Vincentia, Jervis Bay, NSW, Australia overlaid onto the bathymetry (a) and clusters (b). The budget for each path is 2500m. Coordinates are displayed in UTM zone 56S.

the Fortesque region and is highlighted in the clusters (Figure 3.13b), which shows the track lines that correspond to the vessels path when collecting the bathymetry. These artefacts can make the planner to unnecessarily traverse to these regions, however this

		MCTS	RRT	Templates	TSP	L_B
O'Hara	M_{PD}	13.80 ± 1.02	$\mathbf{14.70 \pm 3.15}$	8.60 ± 1.46	9.56 ± 1.19	3.87 ± 0.20
	M_K	$\mathbf{8.96 \pm 0.24}$	9.06 ± 0.05	8.97 ± 0.03	9.26 ± 0.03	9.36 ± 0.00
	M_C	0.96 ± 0.07	0.97 ± 0.05	$\mathbf{0.99 \pm 0.03}$	0.97 ± 0.07	0.72 ± 0.02
	D (m)	2492 ± 11.92	2311 ± 247.45	2500 ± 0.00	1881 ± 361.22	2500 ± 0.00
Vinc.	M_{PD}	16.23 ± 5.71	$\mathbf{18.00 \pm 3.20}$	11.02 ± 2.53	8.49 ± 2.17	2.87 ± 0.25
	M_K	9.07 ± 0.01	8.96 ± 0.09	$\mathbf{8.68 \pm 0.05}$	8.78 ± 0.02	9.14 ± 0.00
	M_C	0.80 ± 0.09	0.76 ± 0.08	0.82 ± 0.06	$\mathbf{0.99 \pm 0.03}$	0.38 ± 0.02
	D (m)	2493 ± 11.59	2401 ± 174.95	2499 ± 0.00	3150 ± 730.10	2500 ± 0.00
Trim.	M_{PD}	5.45 ± 0.83	$\mathbf{5.99 \pm 0.96}$	3.98 ± 0.81	5.14 ± 0.46	4.07 ± 0.04
	M_K	2.98 ± 0.02	2.93 ± 0.03	$\mathbf{2.90 \pm 0.01}$	3.09 ± 0.06	3.20 ± 0.00
	M_C	0.98 ± 0.04	$\mathbf{0.99 \pm 0.03}$	0.93 ± 0.05	$\mathbf{0.99 \pm 0.03}$	0.86 ± 0.01
	D (m)	59 ± 0.54	54 ± 4.75	60 ± 0.00	50 ± 7.71	60 ± 0.00

Table 3.2 – Results for exploring the feature space, where the feature space is latent space of an autoencoder trained on bathymetry patches (Section 2.6.2). ± refers to plus or minus one standard deviation. Each value is the mean value over ten runs. Vincentia is abbreviated to Vinc and Trimodal is abbreviated to Trim. The blue columns indicate paths planned using the information metric proposed in Section 3.4.2, which are compared to Cluster-TSP in the yellow column (Section 3.4.8) and random transects over the area (L_B) in the orange column. As the encoded feature space has more dimensions than the geometric feature space (17 vs 5), the values of M_{PD} and M_K are larger than in Table 3.1.

		MCTS	RRT	Templates	TSP	L_B
Geometric	M_H	0.97 ± 0.10	$\mathbf{1.00 \pm 0.00}$	0.93 ± 0.13	$\mathbf{1.00 \pm 0.00}$	0.89 ± 0.02
Encoded	M_H	$\mathbf{1.00 \pm 0.00}$	0.97 ± 0.10	0.93 ± 0.13	$\mathbf{1.00 \pm 0.00}$	0.89 ± 0.02

Table 3.3 – The average habitat visits for the densely sampled *Trimodal* dataset over ten runs. ± refers to plus or minus one standard deviation. These results highlight how using informed planning leads to observing all the habitats. As this dataset is small compared to the budget, it restricts straight transects to a smaller area making it likely to hit most of the classes.

was not evident in these experiments.

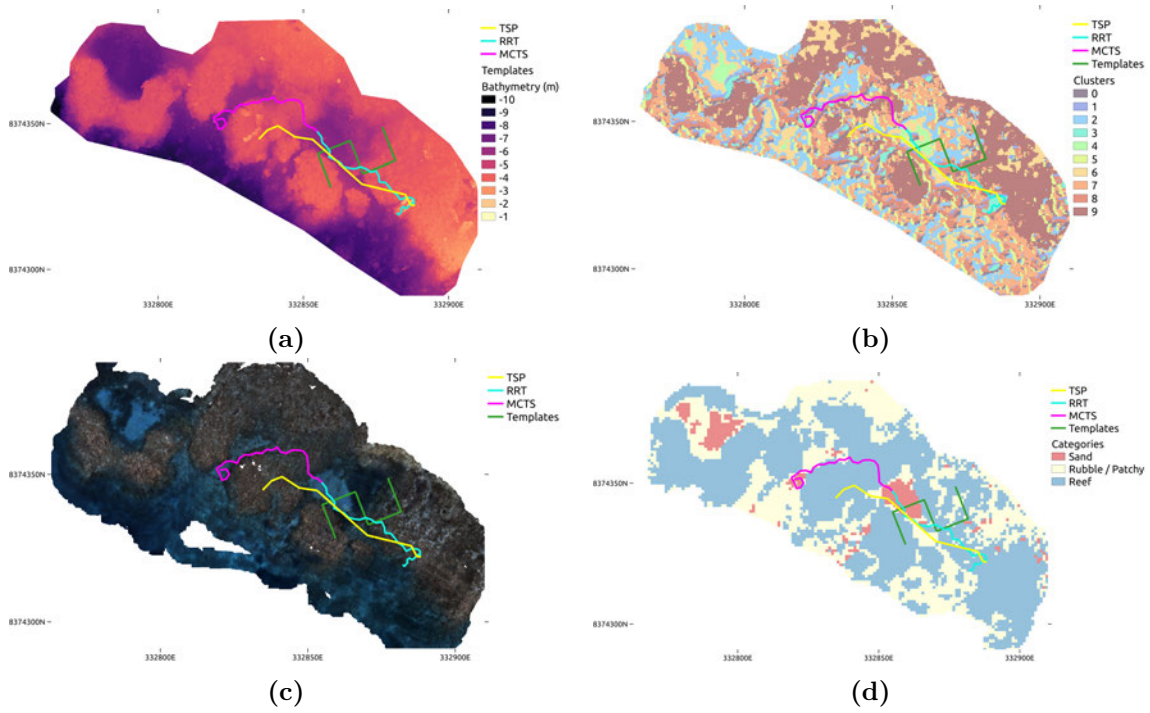


Figure 3.15 – Planned paths using encoded features for Trimodal Reef, Lizard Island, Queensland, Australia overlaid onto the bathymetry (a), clusters (b), image mosaic (c) and habitat labels (d). The budget for each path is 60m. The habitat labels are sand (red), rubble / patchy reef (yellow) and reef (blue). Coordinates are displayed in UTM zone 55S.

3.5.1 Computation Considerations

The planning time is not considered an important factor when comparing these approaches, as the planning time is insignificant compared to the time taken to complete a survey. For dives that will take several hours for the AUV to traverse, the planners take several minutes to plan. Table 3.4 shows the time taken to plan a survey on the O'Hara dataset, with all methods producing an informed plan within several minutes. All runs were performed on the O'Hara dataset, running on a 12-core Ryzen 3900X CPU at 3.8GHz coupled with a Nvidia 2080Ti GPU. Parameters, such as the expansion distance and number of reward samples, have an impact on the time taken to complete the plan and hence they are kept the same as those used to complete the results. The *MCTS* and *RRT* methods are capped at 10,000 iterations, whereas the *Templates* method uses 1,000 candidate surveys. The time budget for the *TSP*

method is set at 180.0 seconds. All these methods are 'any time' planners and can instead be given a planning time budget. The time is reported for planning with geometric features. The time taken to plan with encoded features is the same, by pre-extracting the features at every point on the raster and creating a new geotiff with these features.

The reward function uses the history of features in its calculation and hence will become slower as the number of samples increases, however this is not significant in testing. For the *MCTS* method, performing a 'heavy playout' (Browne et al., 2012) by increasing the number of simulations performed during each cycle significantly extends the execution time. Four iterations per cycle were empirically found to balance execution time and quality of the resulting survey. For the *RRT* method, the parameter that most increases execution time is the number of nearest neighbours to evaluate during the *Select* process. In these experiments 50 nearest neighbours were used.

	MCTS	RRT	Templates	TSP
Time (s)	164	116	454	180

Table 3.4 – Time (in seconds) to plan one run for each method.

3.5.2 Feature Visualisation

To demonstrate the features visited along the paths, Figure 3.16 shows a 2D projection of the encoded feature space using the t-SNE algorithm (Van Der Maaten and Hinton, 2008). For this example, the O'Hara dataset is used and the example paths present in Figure 3.13. 10,000 randomly sampled features from the area of interest are plotted, coloured as the clusters from Figure 3.13 and partially transparent. Features visited on the planned paths are overlaid as solid points with a dark border. Although this 2D representation does not preserve feature distances, it shows the planned paths collecting features from all regions of the feature space.

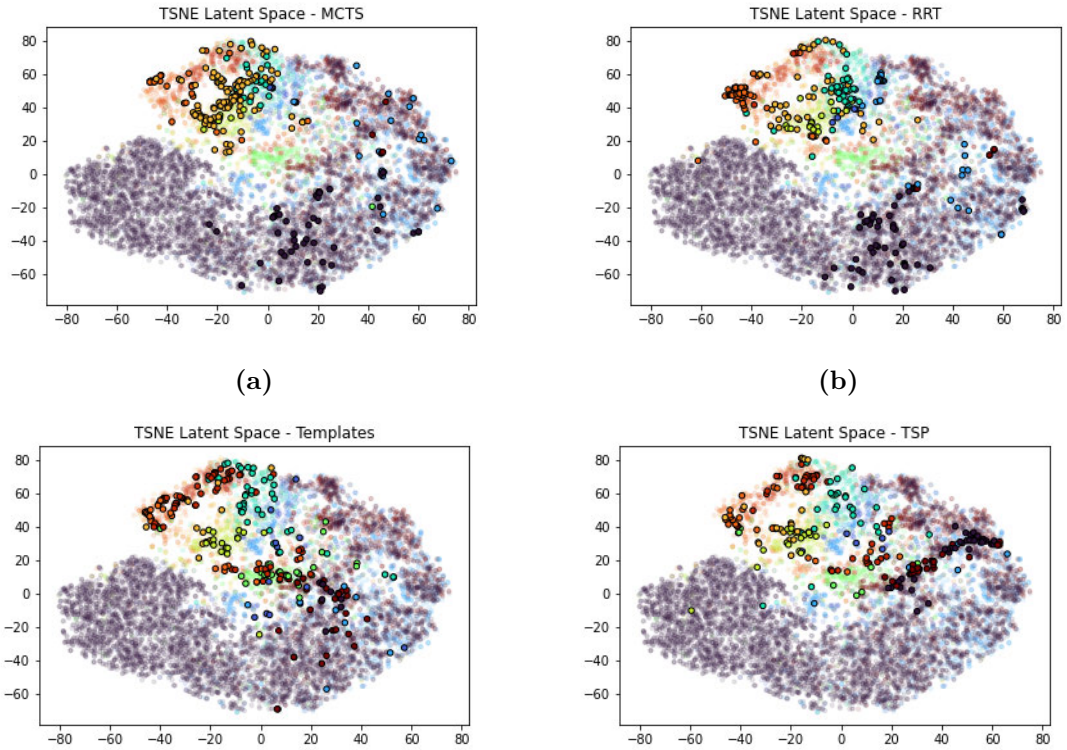


Figure 3.16 – 2D t-SNE projection of the encoded feature space for the O'Hara region, and each path's respective exploration of this feature space. The partially transparent points are features randomly sampled from the area with the colours corresponding to the clusters outlined in Figure 3.13b. The opaque points with the black border represent the features extracted from the respective paths.

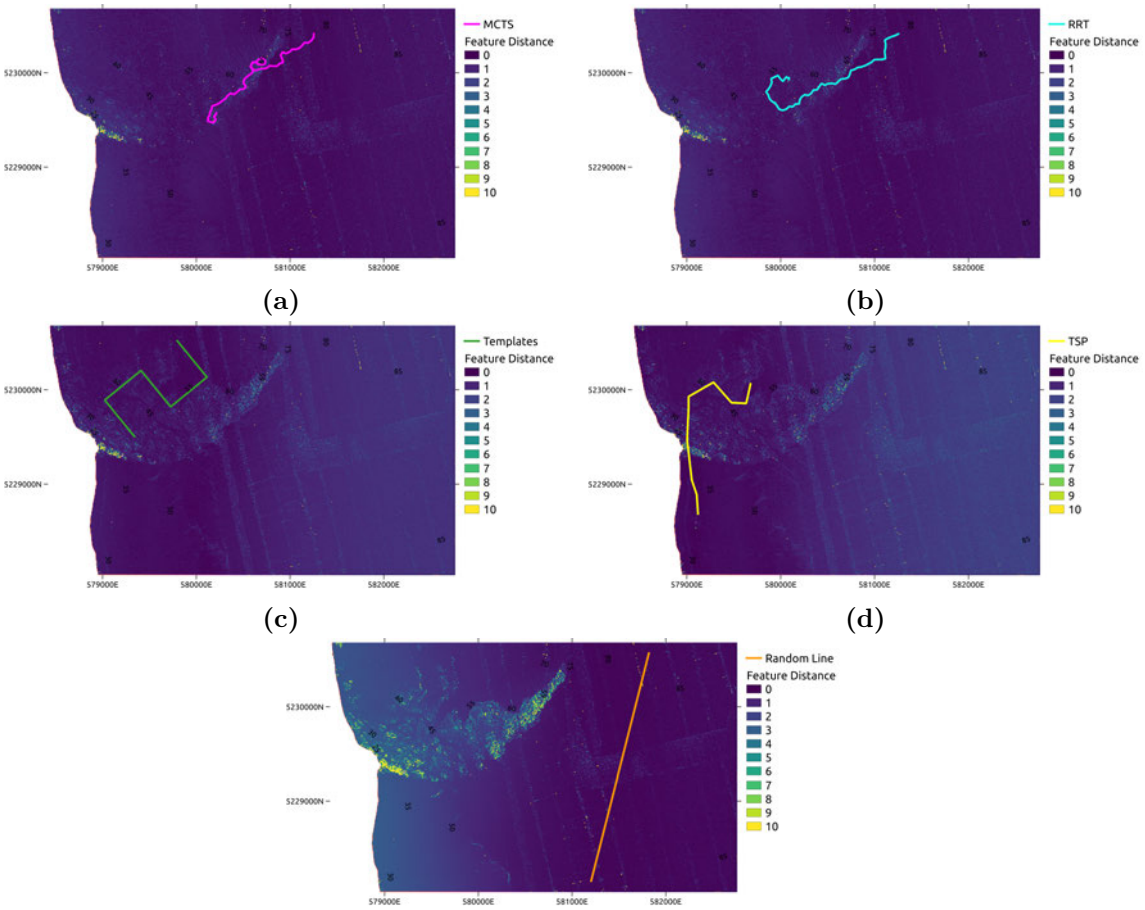


Figure 3.17 – Mapping the geometric feature distance from the feature at a given location, to planned survey paths for the O'Hara region. (a) *MCTS*, (b) *RRT*, (c) *Templates*, (d) *TSP* and (e) a random line. Areas that have a small feature distance are well explained by features collected on the respective paths. The *MCTS* and *RRT* surveys exhibit lower feature distances over the survey area. Coordinates are displayed in UTM zone 55S.

3.5.3 Relative Feature Distance

The objective of these initial planning surveys is to collect an initial survey that best captures the range of bathymetric terrain present in the area. Providing a set of training examples that comprehensively explore the feature space enables informed habitat models of the area to be created. An effective way to visualise this is to map the feature distance (Equation 3.5) for each point in the survey area, to features collected on the respective initial surveys. This is visualised in Figure 3.17 for geometric features. For the L_B benchmark method, a single random transect is used (Figure 3.17e). The dark areas indicate a smaller feature distance, which means that features from that area are similar to features collected on the path. As captured in Figure 3.17 the paths planned with *RRT* and *MCTS* minimise the feature distance over the entire survey area. The maps in Figure 3.17 highlight the benefit of informed sampling. An uninformed, randomly-placed transect (Figure 3.17e) explains the large, sparse area that dominates the survey area, but fails to sample from the reef area that is likely to contain varying benthic habitats. Informed sampling, as demonstrated by the *RRT*, *MCTS*, *Templates* and *TSP* is able to explain the large, sparse area despite only sampling there briefly. These maps are plotted as histograms in Figure 3.18, where the feature distances are binned and counted. The histograms show that informative surveys have a larger number of small feature distances and fewer large distances, highlighting that the informative surveys are a better representation of the entire survey area.

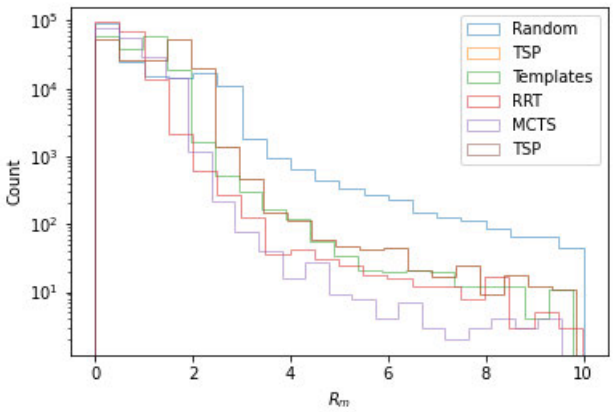


Figure 3.18 – A histogram representation of the feature distance maps in Figure 3.17. The log-scaled y-axis is the count of pixels corresponding to each feature distance bin. The surveys planned with *MCTS* and *RRT*, have a larger count of smaller feature distances and lesser counts of higher feature distances, indicating a better representation of the feature space. All the informatively planned surveys better characterise the feature space than the random survey.

When increasing the dimensions of the feature space, the differences in feature distance can be harder to perceive due to increased sparsity and distance concentration. Features with seemingly small differences between them can have large feature distances. While increasing the feature space dimensions can be advantageous, by prompting the robot to explore interesting areas, it can also lead to the robot over-exploring habitats whose features are diverse, at the expense of sampling from other habitats. Figure 3.19 presents the feature distance for each point in the survey area from the features collected on each respective path. Using this feature representation, there is an advantage of using the *RRT* path (Figure 3.19b) which is able to best approximate the reef dominated areas of the O'Hara Reef and the vast, bare terrain that exists beyond it.

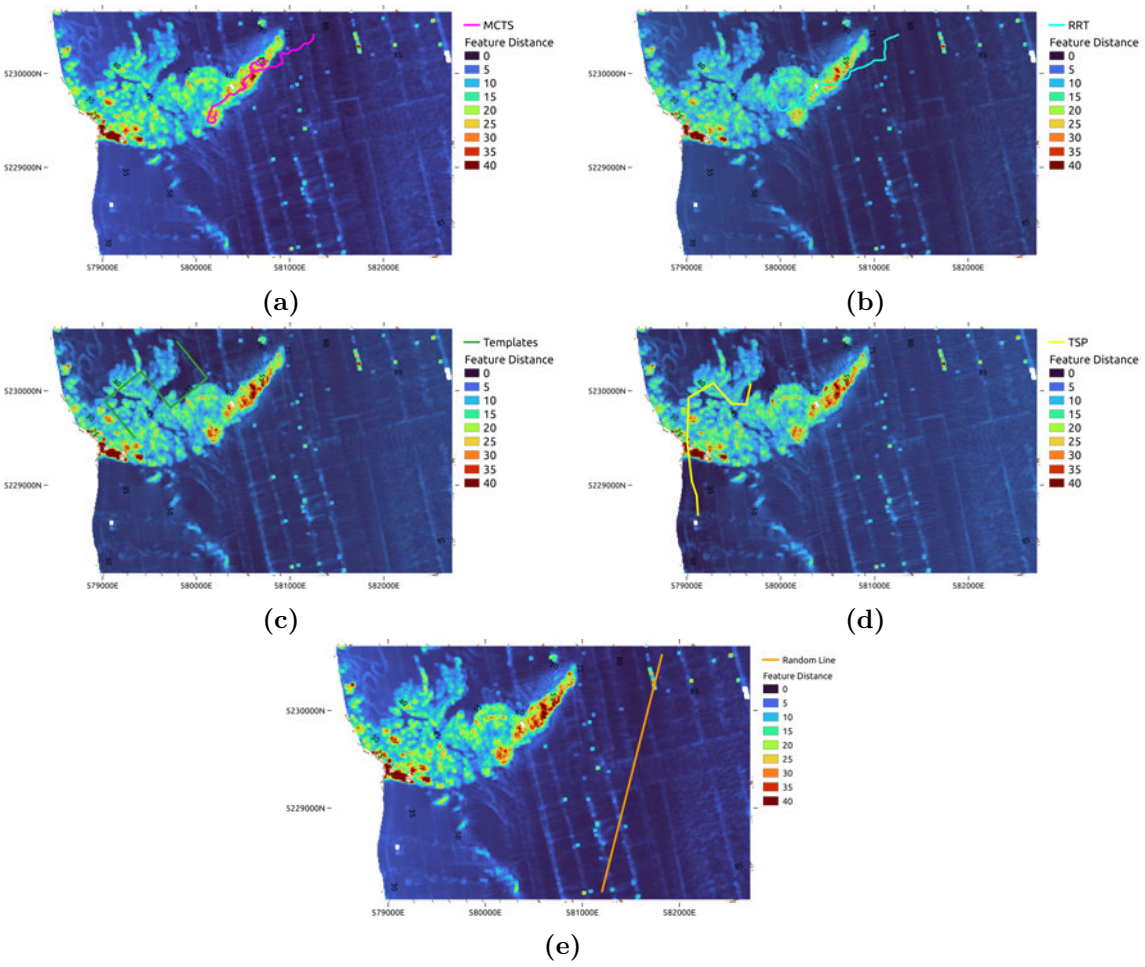


Figure 3.19 – Mapping the encoded feature distance from the feature at a given location, to planned survey paths for the O'Hara region. The layout mimics Figure 3.17. For reference the plots correspond to (a) MCTS, (b) RRT, (c) Templates, (d) TSP and (e) random transect. Due to increased sparsity and distance concentration in higher dimensions, the difference between each map is harder to perceive. The 'turbo'colour map is used to increase contrast. Coordinates are displayed in UTM zone 55S.

3.5.4 Budget Considerations

For the results presented in Section 3.5, the distance budget is set using the *TSP* method to provide a reasonable distance in which to explore the area. For the deployments (in Section 3.6) the budget was set to allow for time for two deployments in a single day. Without this operating constraint, the optimal operating distance budget should be determined that sufficiently characterises the environment. As the

paths consider an entire trajectory together, simply using sections of a longer path is not sufficient, as the path may be traversing a low value area to get to a high value area. This experiment looks at planning with several different budgets, to identify at what point further sampling is not effective. The information reward methods (*RRT*, *MCTS* and *Templates*) methods are compared. While a budget limit can be set to *TSP*, it will aim to find the shortest path and so will produce a path under the budget if possible. The budgets selected were $\{1,000, 2,500, 5,000, 1,0000\}$ metres, which offer a range of path lengths than can potentially visit all the different areas of terrain. Paths shorter than 1,000m are likely to be ineffective as this distance is too short to visit all the different areas of terrain. Paths longer than 1,0000m were not evaluated as the set survey template of the *Templates* method cannot fit within the survey area without altering it. It is not possible to compare the benchmark method L_B as the survey area is too small to fit a 5,000m or a 10,000m line. The planned paths for each method and distance budget are displayed in Figure 3.20. The *RRT* and *MCTS* follow similar trajectories for the smaller budgets, moving from the deeper flatter areas to explore the reef outcrop, making their way towards the shallower rugose areas.

Figure 3.21b shows the proportion of clusters visited for each path at each distance budget. With a budget of 1,000m, the paths fail to visit all the clusters, suggesting this path length is too short. For a budget of 2,500m, the *RRT* visits all the clusters while the *MCTS* and *Templates* methods visit on average 96% and 97% of clusters respectively. At 5,000m, all the paths visit all of the clusters and there is a significant elbow point in the M_K metric for all the methods. For the *Templates* method, the larger survey template of the 10,000m path could not effectively align with the terrain, while the *RRT* method offered no improvement with the 10,000m survey. However, the 10,000m surveys result in a further reduction in the M_K metric for the *MCTS* method, highlighting the benefit of further exploration. This shows that further exploration can be beneficial, even when the path has visited all the clusters. When trying to identify an effective budget, proposing a value of M_K at which the environment is completely explored would be arbitrary and misleading, however evaluating a range

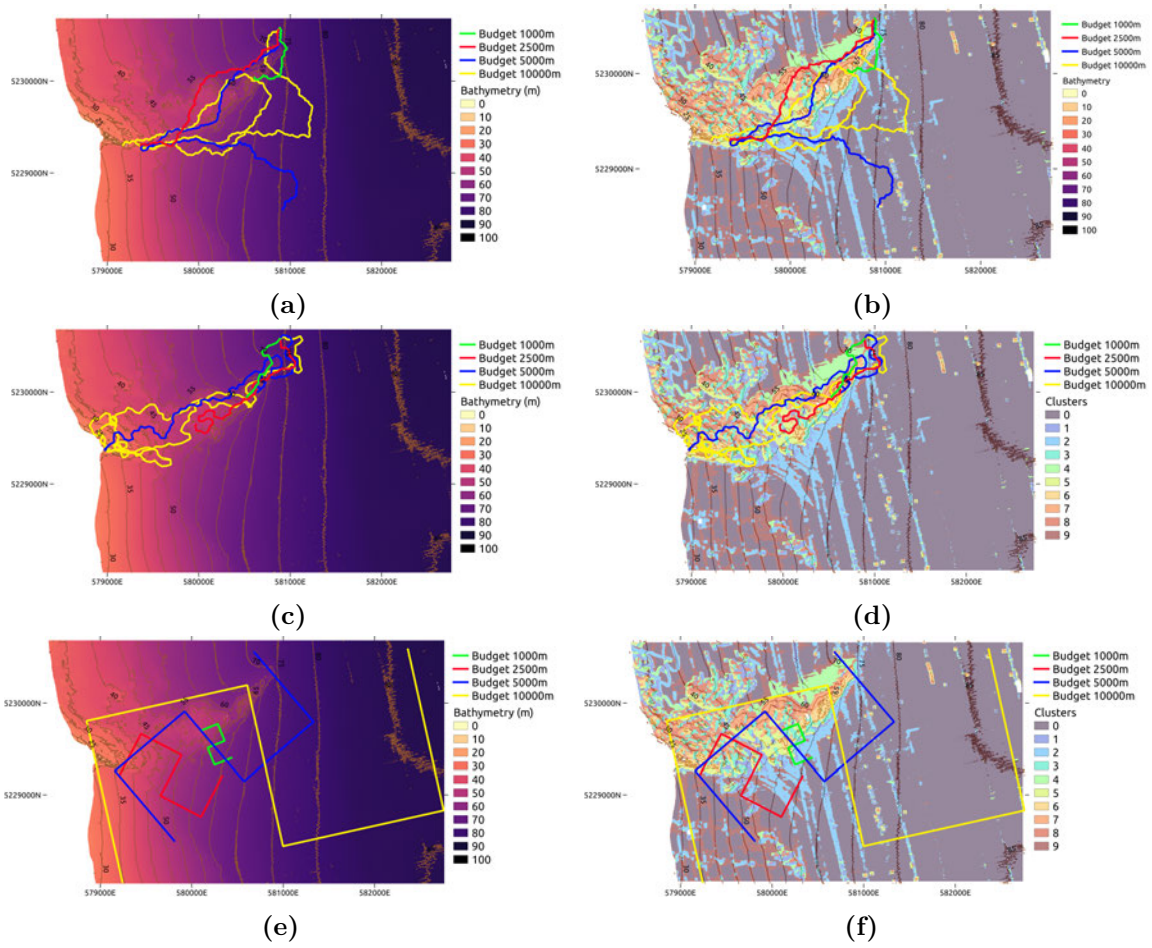
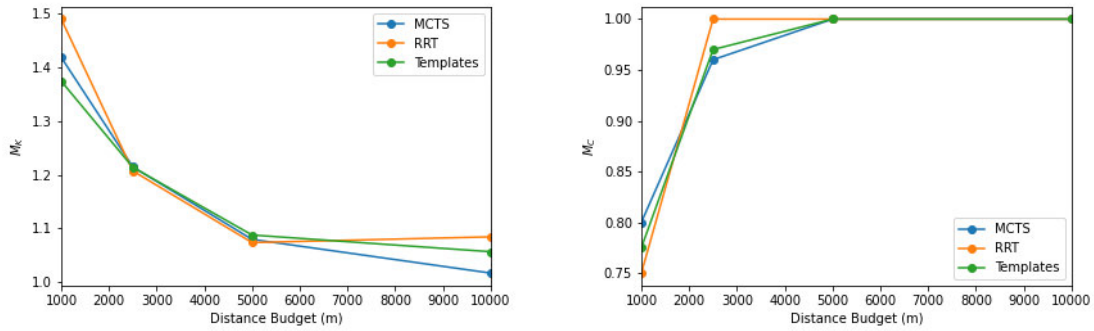


Figure 3.20 – Paths planned with several distance budgets, using *RRT* (a,b), *MCTS* (c,d) and *Templates* (e,f). Coordinates are displayed in UTM zone 55S.

of budgets can identify elbow points, where further exploration is less beneficial. Selecting the budget so that the path can visit all the clusters is a more effective way to determine an efficient budget.



(a) **(b)**

Figure 3.21 – Plots show the M_K metric (a) and the M_C metric (b) for each planning method with several different budgets. With a 1,000m budget, the paths do not sufficiently explore the feature space. The M_C metric shows that after after 5,000m all paths have visited all the clusters. The *MCTS* method shows continual improvement on the M_K metric, while the *RRT* and *Templates* exhibit little performance improvement with a budget of 10,000m.

3.6 Field Deployments

Experimental field trials of the informative path planning were carried out by the Universitat de Girona with the Sparus II AUV (Carreras et al., 2018) exploring a canyon offshore of Blanes, Spain. The objective was to explore the bathymetric feature space with a limited budget. The shorter paths are compared against a longer broad grid. The broad grid was generated automatically from a large bounding box covering the area of interest. It was designed to cover a large area while also exploring the depth range of the map. The original broad grid was trimmed to allow for the survey to be conducted in a single day. A budget of 2500m was set for the shorter paths, allowing two surveys to be conducted in a single day. An encoded feature space was used but with a smaller patch size of 11×11 cells (or 44×44 meters given pixel size of 4 meters), allowing for a feature dimension of 16 to reduce the negative impacts of the high dimensionality present in Section 3.5.

When deploying the AUVs in the field, the plans generated have to ensure safe operation of the AUV. For these field trials, the position of the altitude sensor meant the AUV could not go up steep inclines so the paths were restricted to go predominantly

downhill and only allow a rise of 13.5°. This restricts the type of terrain the AUV can explore. This constraint was integrated into the planners. For the incremental planners (*MCTS* and *RRT*), constraints can be integrated naturally, where if the path expansion violates the constraint, it is disallowed. For the *Templates* method, the constraints have to be checked for the entire path and if any section of the path breaks the constraint that survey template position is disallowed. This is much less effective and leads to a large number of disallowed surveys, whereas the incremental planners can potentially plan around the area violating the constraint.

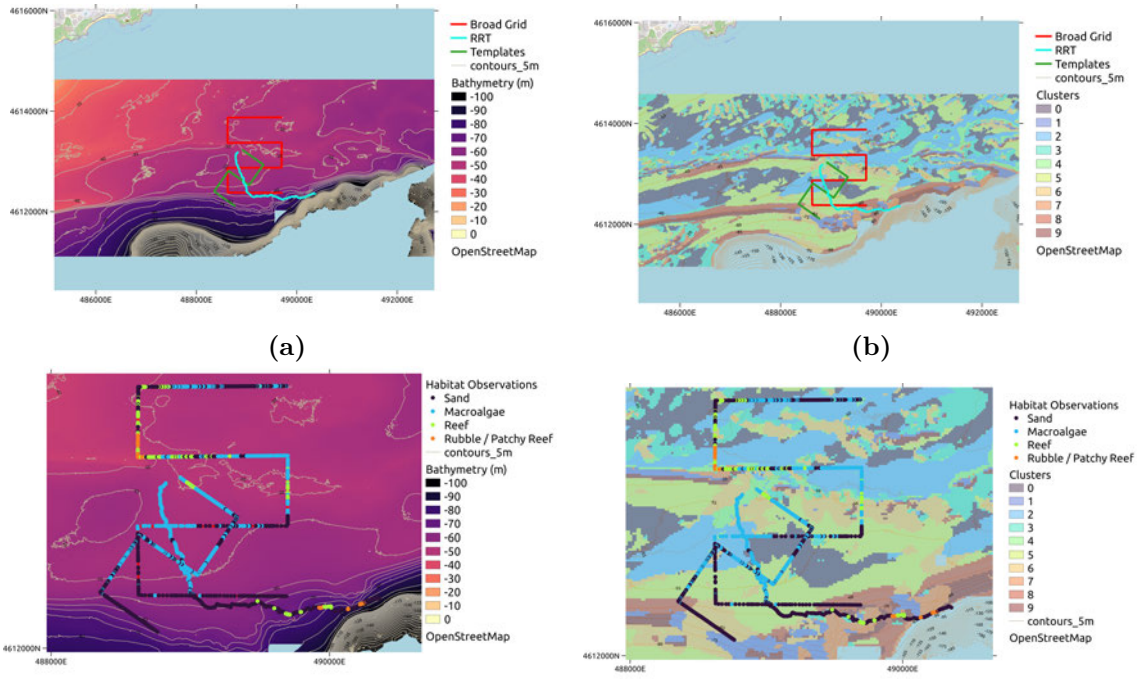


Figure 3.22 – Paths planned and performed by the Sparus II AUV exploring a canyon offshore of Blanes, Spain. The left column shows the paths over the bathymetry, while the right shows the path over the bathymetric clusters. The top row highlights each of the respective paths, while the bottom row shows the habitat observations, obtained from the Sparus II imagery. The RRT and Benchmark paths observed all the the habitats. Coordinates are displayed in UTM zone 31N.

The paths planned by each method are displayed in Figures 3.22a and 3.22b. The respective evaluation metrics are displayed in Table 3.5. The *Broad Grid* displays slightly better feature space exploration metrics (a lower M_K) but also travels signif-

	RRT	Templates	Broad Grid
M_{PD}	**3.24**	1.97	2.11
M_K	2.39	2.50	**2.34**
M_C	**1.0**	0.8	**1.0**
D(m)	**2499**	2500	5792

Table 3.5 – Evaluation metrics for the field deployments near Blanes, Spain.

icantly further, 5792m vs 2500m for the informative trajectories. The *RRT* method has a slightly worse M_K metric, but it still visits all of the clusters. The template method performs relatively poorly compared to the other methods, indicating that the chosen template type is not well suited to this area.

The imagery collected by the Sparus II AUV was labelled into one of four broad habitat classes; sand, macroalgae, reef and rubble. Examples of these images and the corresponding habitat labels are displayed in Figure 3.23. The *RRT* and *Broad Grid* both observe all the habitat categories. This highlights that the shorter informative path is able to explore the feature space with a limited budget. The *Template* method does not observe the rubble habitat, due to either a poor fit between the template and the area, or the rubble habitat not being identifiable from the bathymetry. When considering the entire *Broad Grid* path, it visits more reef areas on the first half, however the second half (2896m) of the survey does not visit any reef areas, highlighting the risk when not using informative planning.

The starting position of *RRT* was selected using the method for finding an informative starting region (outlined in Section 3.4.4), which generally picks successful starting positions on the border between multiple areas of bathymetric terrain. The starting position selected for the *RRT* method is in an informative area, however it is positioned deeper than a gathering of clusters just beyond it. This gathering of clusters is where the *Broad Grid* observed reef habitats. Before the slope constraint was added, the *RRT* path starting from this same starting position would visit the gathering of clusters before descending on a similar trajectory to the current *RRT* path. However, when the slope constraint was added, the *RRT* path could not ascend to this area and was forced to move deeper. Although the *RRT* path still observed the reef habitats,

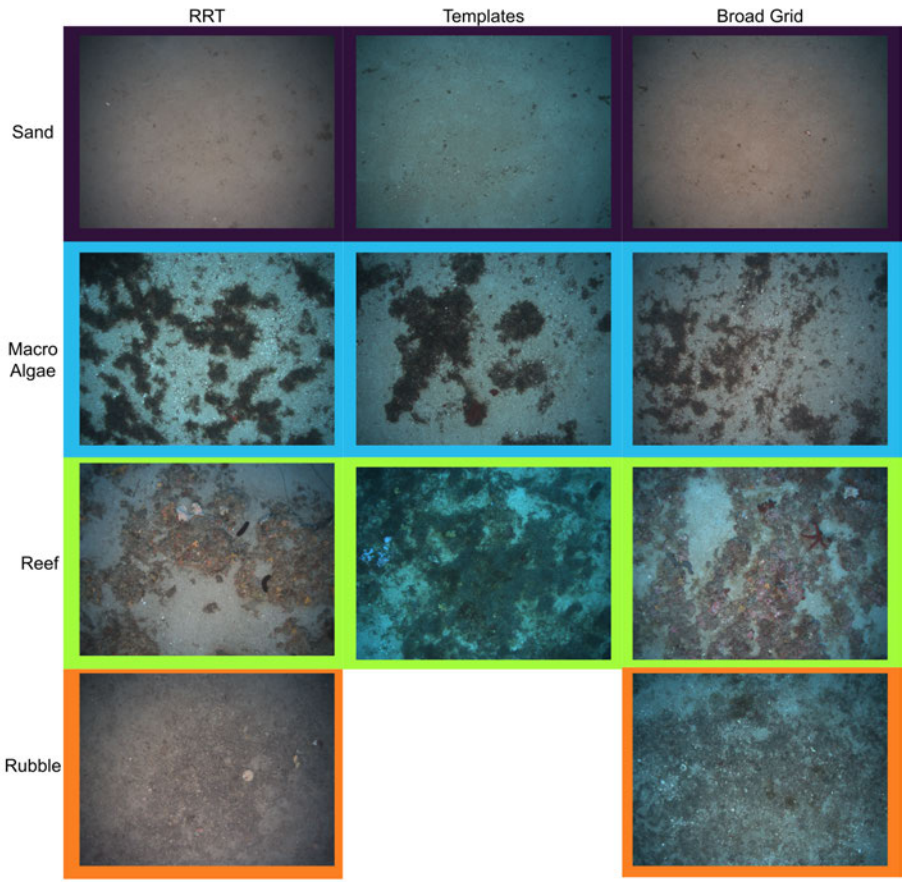


Figure 3.23 – Images captured by the Sparus II AUV during surveys of the underwater canyon near Blanes, Spain. The border colour corresponds to the habitat class of the image, which is displayed spatially in Figure 3.22c. The *RRT* and *Broad Grid* paths observed all the habitat classes, while the *Templates* path did not observe the Rubble class.

a more efficient path could have been possible with a different starting position. This highlights the impact of the starting position and the need to adapt the process to integrate the constraints.

The field trials demonstrate how a shorter informative path can be used to explore the feature space of an area. They highlight the adaptability of these methods to real-world constraints, however a tighter integration of the constraints into the process finding the starting position can further improve these surveys.

3.7 Summary

The large spatial extents of survey areas combined with the small sensor footprint require efficient sampling of the survey area to maximise the information collected with limited resources. Under the assumption that full-cover bathymetry of the survey area is available and has a tangible connection to the benthic habitats present, it can therefore be used as a prior to plan initial surveys. This chapter presents a principled approach that links feature-based active learning to information gathering planning, to design initial benthic surveys that comprehensively explore the feature space. By developing a reward function that prioritises samples that are distinct from those already on the path, the informativeness of the entire trajectory is increased.

This research proposes a set of informative planners that utilise this reward function. These planners either plan freeform surveys (*MCTS*, *RRT*) or place survey templates (*Templates*). All these planners exhibit significant performance improvements over random transects, highlighting the benefit of an informed survey. Furthermore, these paths more thoroughly explore the feature space than the *Cluster-TSP* method, where each of the bathymetric cluster types are visited. When planning freeform surveys, the *RRT* method performs marginally better on most datasets than the *MCTS* method. When the survey template being evaluated is matched well to the informative areas of the survey area, informatively placing survey templates is an effective method for planning.

This chapter demonstrates planning using two different feature extraction methods for the bathymetry, geometric features and encoded features. For both feature representations the informed survey paths comprehensively explore the feature space. However, as the number of dimensions increases, sparsity and distance concentration can reduce interpretability of the results. The relatively large distances between seemingly similar terrain can lead the planners to over-sample in some areas. This can be seen in some of the surveys based on large feature spaces, with the planners preferring to focus on rugose reef sections rather than visiting other areas of the survey area. Therefore the dimensionality of the feature space should be carefully considered

as there is a trade-off between expressibility of the feature space and the ability to effectively plan over it.

The field trials conducted showed a shorter informative survey could still explore the feature space, but indicated further work is necessary to improve the planners to satisfy real-world constraints and objectives. Further development could also focus on making the planner more risk-averse, including ensuring the trajectory is safe when there are localisation errors.

Chapter 4

Adaptive Sampling through Active Learning

4.1 Introduction

This chapter examines planning AUV trajectories when there are already existing in-situ observations of the seafloor. It builds upon Chapter 2 which analysed the process of habitat modelling including the development of probabilistic habitat models whose uncertainty estimates can be used to drive further sampling. It also builds upon Chapter 3, which investigated planning informative AUV trajectories before any previous samples have been collected. It proposed exploring a feature space representation of the remotely-sensed data while traversing the physical space. This chapter brings together the contributions of Chapters 2 and Chapter 3 by utilising the habitat models in conjunction with the AUV trajectory planners to plan informative trajectories that most improve the underlying habitat models.

For typical benthic AUV surveys, the AUV follows a set survey path. These survey paths are either planned manually or programatically (Bender et al., 2013b; Foster et al., 2020). Following the survey, the data is post-processed and labelled (or inferred using a machine learning method) to obtain the habitat observations. While

the AUV could potentially react to the imagery online, assigning an accurate habitat observation and retraining the habitat model is complex and requires significant computing resources and therefore remains an avenue of further research. Instead, this research aims to propose a set survey path that collects a set of habitat observations that *together* will most improve the habitat model. Ensuring the AUV collects informative samples along its entire trajectory is a key challenge. For example, if there are two areas with similar bathymetric terrain and both are identified as highly informative by the habitat model, if the AUV visits one of those areas, it no longer needs to visit the other area as it has already sampled from a similar area. If the habitat model was re-trained after visiting the first area, it would no longer highlight the second area as highly informative. To address this, this research uses batch-mode active learning, which involves selecting data points that together will most improve the underlying model. This ensures diversity of samples across the survey. Specifically, a combination of feature-based and model-based active learning techniques are proposed, that ensure diversity of samples across the survey. This combination allows the feature-based active learning to overcome an overconfident habitat model and respond correctly to OOD data (data that is different to data already observed).

To plan an informative survey, an informative planner based on Monte Carlo Tree Search (MCTS) is utilised (developed in Chapter 3), that is guided by on objective function designed to most improve the habitat model. The MCTS based planner is robust and can be easily integrated with different model constraints. The objective function can be based on feature-based or model-based active learning algorithms. Utilising a hybrid approach, that jointly uses feature-based and model-based active learning, allows targeting of areas of high uncertainty while also ensuring the robot explores diverse terrain.

This chapter aims to demonstrate the benefit of using active sampling to plan informative AUV surveys over the course of a *campaign*. In this context, a campaign refers to repeated AUV deployments over a survey area with the goal of characterising the area. For this chapter, a campaign is defined as a set number of AUV dives concentrated on a single survey area. This research frames a campaign as an active

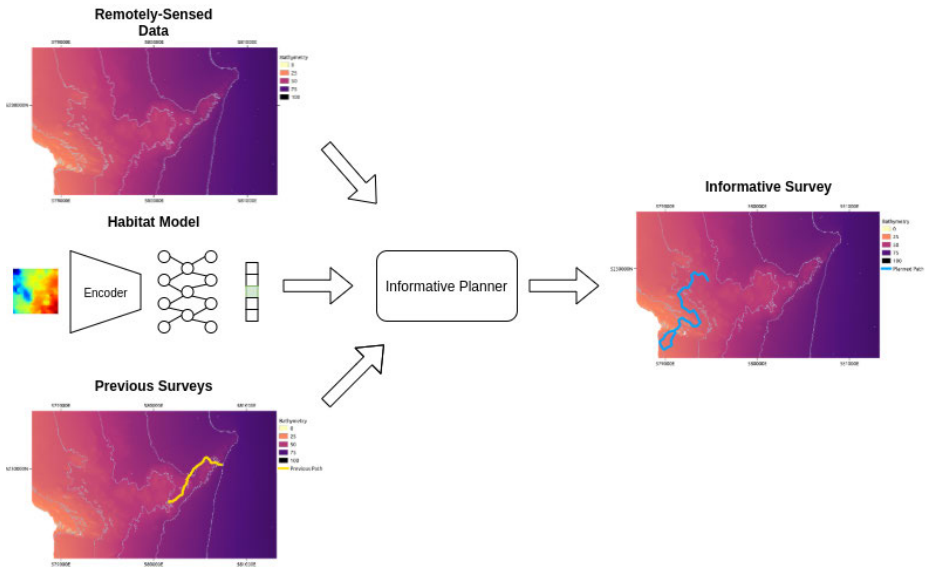


Figure 4.1 – The proposed informative planner, which uses remotely-sensed data, the probabilistic habitat model and the previous surveys to plan the next survey.

learning cycle, as shown in Figure 4.2. Remotely-sensed data is often available prior to the survey or can be collected from the survey vessel's multibeam sonar. Using this remotely-sensed data, an initial survey is planned using the methods developed in Chapter 3 which is used to seed the active learning loop. After the AUV is recovered, the imagery is retrieved and labelled to obtain the habit observations. The georeferenced habitat observations are then used to train the habitat model. The next survey is planned by an informative planner that utilises the habitat model and the previous surveys.

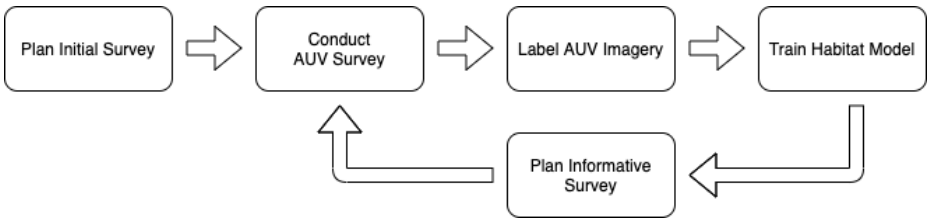


Figure 4.2 – An AUV campaign as an active learning cycle. First, a seed survey is planned using the methods developed in Chapter 3. Then the AUV is deployed to follow this survey, collecting imagery of the seafloor. When the AUV is recovered, the imagery is labelled to obtain the habitat observations. The habitat observations are used to train a habitat model. An informative planner then plans the next survey, using the habitat model and the past surveys. The cycle then repeats, further improving the habitat model for the duration of the campaign.

The primary contributions of this chapter are:

- The conceptualisation of an AUV survey campaign as an active learning cycle, with the goal to most improve the habitat model.
- Fusion of model-based and feature-based information objectives, to achieve both exploration of unique features and exploitation of the decision boundary, whilst ensuring there is diversity amongst the samples collected.
- Analysis of model, feature and hybrid active learning techniques in the context of adaptive benthic AUV surveys.
- Development of a simulated but realistic environment for testing adaptive planners.

- Experimental results that demonstrate the improved performance of informative AUV surveys in most-improving a habitat model.

The remainder of this Chapter is organised as follows: Section 4.2 provides a review of the related literature, Section 4.3 proposes the information objectives used to guide the AUV, Section 4.4 outlines the data used in the experiments, Section 4.5 presents active learning experiments performed on segments of previous AUV surveys, Section 4.6 reviews the informative planner used, Section 4.7 presents the active sampling experiments, Section 4.8 outlines the planned field deployments used to validate this work. Finally, Section 4.9 summarises the contributions of this chapter.

4.2 Related Work

Active learning is the process of training sample selection, where the goal is to improve an underlying model (Settles, 2009). In its most common form, the active learning algorithm selects what data samples from a large dataset pool should be used to train a model. Modelling the relationship between remotely-sensed data and in-situ samples is a form of supervised machine learning, where the cost of acquiring labels (conducting AUV surveys) is very high. Active learning techniques can be utilised to select samples for training and foster informative and efficient AUV surveys.

There are two main approaches to active-learning algorithms; model-based and feature-based (Settles, 2009). Model-based approaches use the underlying model's predictions on the pool dataset to identify samples that are most likely to lead to its improvement. Feature-based approaches select samples for training that represent the distribution of the dataset. There is an exploitation / exploration tradeoff inherit in this problem. Typically, model-based approaches are used for exploitation by further refining the decision boundary, while feature-based approaches are used for exploration of the entire feature space.

In an active learning loop, it is mostly infeasible to retrain a model after selecting each individual data point. Larger models (such as neural networks) need large datasets

and take significant time and computing resources to train, and adding a single point can have a negligible effect on the model. Furthermore, it is also often impractical to acquire labels singularly. Therefore active learning algorithms need to select batches of data samples that *together* improve the model. This is known as Batch-Mode Active Learning (BMAL) (Ren et al., 2020). This is also the case for the problem addressed in this chapter, where an active sampling algorithm has to preplan an informative trajectory. An active sampling algorithm that does not consider the interrelationship between data points will select points that are high reward, but these high reward points will often be similar to other points in that batch (or trajectory) and therefore this batch of data points will have a reduced impact on model improvement.

The idea behind feature-based active learning is that queried samples should be both similar to unlaballed data points and dissimilar to already collected samples (Fujii et al., 1998). To select a set of samples that represent the dataset, Nguyen and Smeulders (2004) first cluster the feature space, then queries from each cluster. This enables the distribution of features in the selected dataset to match the overall dataset. Geifman and El-Yaniv (2017) use the intermediate activations of a pretrained neural network as a feature space, and postulate that the next sample collected should be farthest in feature space from the already collected samples. Sener and Savarese (2018) propose a core-set approach for selecting a set of features that are representative of the overall feature space. They achieve this by solving a robust K-centre problem, that optimally selects K data points based on their position in the feature space. This method was demonstrated to significantly outperform uncertainty-based model driven methods, due to the ability to select data points that together approximate the feature space. To classify benthic images captured by an AUV, Yamada et al. (2022) observes that using training samples that are distributed across the feature space results in higher accuracy than using class-balancing the training samples, highlighting the importance of comprehensively sampling from the feature space. While feature-based active learning approaches guarantee a coverage of the range of features present in the dataset, they do not focus on areas around classification decision boundaries, which can impact the performance of these active learning methods. Furthermore,

feature-based methods rely on distance metrics, which are known to perform poorly in high dimensions, which is common in many machine learning tasks including image classification. For the bathymetry classification task, as the input bathymetry patches are small (e.g. 11×11 pixels) and the autoencoder is effective at compressing them, the features spaces are small and therefore these approaches are applicable.

Model-based active learning involves using the model to identify data points that will most improve the model. Model-based active learning algorithms typically query the model's uncertainty for a given data point, with higher uncertainty points indicating that training with these data points will lead to more model improvement. Alternatively, the expected model change can also be used to identify data points that lead to the most model change (Settles, 2009). For uncertainty-based active learning algorithms, the entropy of the prediction is commonly used to guide further sampling. For classification problems, the categorical entropy is used, with this strategy selecting data points on the decision boundary between the classes. However, when using this strategy with neural networks with the softmax activation function, the network routinely overestimates the confidence of the prediction and is a poor uncertainty estimate. Naively using this uncertainty estimate can be worse than random sampling (Guo et al., 2017). Methods such as *temperature scaling* (Guo et al., 2017) aim to calibrate the softmax output to provide a better estimate of the probability, by calibrating each model on a hold-out set. However, the versatility of these methods has been contested, with Ovadia et al. (2019) identifying that the probability estimates are well calibrated for the hold-out set but do not generalise beyond that. A key issue with using the categorical entropy to guide further sampling is that it measures the confusion between classes and has no mechanism to reflect the model's uncertainty. It has no ability to detect Out Of Distribution (OOD) data.

Probabilistic classifiers were evaluated in Chapter 2, demonstrating that they can output valid uncertainty estimates that can be used to drive further sampling. Although significantly better than deterministic classifiers, they can still be overconfident in their predictions and may fail to identify OOD data.

Houlsby et al. (2011) proposes an active learning algorithm that uses a probabilistic

classifier, which aims to capture the mutual information between the model parameters (weights) and the pooled data points. BALD highlights data points that the model most disagrees upon, which are points that have both high entropy of the model prediction and a low expectation of the entropy of the model prediction over the posterior of the model parameters (Kirsch et al., 2019). However, BALD is only designed to score each individual points and it ignores the similarity between data points. When using it in an active learning scenario, each data point in the pool is scored and the k data points with the highest BALD scores are selected. As the top ranked data points can be similar, the extra utility of the extra data points decreases. BatchBALD extends BALD by estimating the mutual information between multiple points and the model information. This is achieved through computing the joint entropy of the predictions in the batch. The computational complexity of computing the joint entropy is reduced through matrix factorisation and sub-sampling of all possible combinations, although it still remains very computationally intensive. This approach is effective for small batch sizes (less than 40 samples) but its performance begins to degrade as the number of samples in each batch increases. Pinsler et al. (2019) proposes another batch mode, model-based active learning algorithm, that takes inspiration from core-sets (Sener and Savarese, 2018), which acts on the feature space but identifies that often the feature space is too complex and highly-dimensional for these methods to work effectively. To address this, they use the model's predicted posterior to find a sparse subset of the entire dataset, such that the log posterior of the sparse subset best approximates the entire dataset.

Yin et al. (2017) identify that an active learning algorithm needs to both refine the decision boundary and explore the feature space. They propose Exploration-P, that both exploits the decision boundary using maximum entropy, and explores the feature-space by selecting data points that are farthest from collected data points. Ash et al. (2020) propose a hybrid active learning strategy that operates in gradient space. The uncertainty is estimated by the gradient norm of the final layer, while also ensuring there is diversity in the gradients of an intermediate layer in the network. Hybrid approaches are effective as they ensure highly uncertain samples are targeted

while also ensuring there is diversity amongst the samples collected. Furthermore, by relying on feature based active learning, they can compensate for an overly confident classifier.

Some active learning techniques aim to learn the characteristics of the labelled and unlabelled data. Yoo and Kweon (2019) trains a loss prediction module that can is used to identify what unlabelled data points are likely to lead to an incorrect prediction. Prioritising the high loss data points leads to fast model improvement. Sinha et al. (2019) develops Variational Adversarial Active Learning (VAAL) where a VAE is used to compress input data into a latent space, which a discriminator uses as input to predict whether the data is labelled or unlabelled. Modelling the labelled or unlabelled data is an attractive approach as it scales readily to large input data sizes (for example images), whereas feature-based methods do not.

Model-based active learning algorithms need to be guided by an underlying model, which needs to be trained with a seed dataset. 'Cold start' in active learning refers to the ability for the algorithm to select its first samples. Model-based active learning algorithms generally do not have the capacity to 'cold start' and many approaches rely on a randomly-sampled initial seed dataset. However, for the AUV active sampling application, random sampling is inefficient due to the spatial correlation of samples and the difficulty in deploying and recovering the AUV. Chapter 3 addresses the cold start problem by utilising remotely-sensed data to create an informative initial dive, which can create a seed dataset that comprehensively covers the bathymetric feature space. This chapter builds upon Chapter 3 by focusing on planning a sampling trajectory once initial samples have already been collected and therefore can utilise both model and feature-based active learning algorithms.

Information gathering planners can also be used to further improve a classification model, a branch of active perception known as active classification. Patten et al. (2018) proposes Monte Carlo Active Perception (MCAP), an extension of MCTS for the purpose of active classification. It is demonstrated by optimising the path of a robot tasked with classifying objects using a lidar sensor mounted on the robot. MCAP involves performing simulations of further actions and estimating the value of

these potential actions, which allows for better evaluation of an action in the search tree and hence improves decision making for complex search spaces. Using this algorithm for active classification significantly improved upon random sampling. Popovic et al. (2017) develops an active classification framework that is used to plan 3D UAV trajectories. The objective of the informative planner is to classify objects on the ground (such as weeds in an agriculture monitoring application), with a key challenge being the difference in classification accuracy with altitude and the corresponding tradeoff between coverage and accuracy. By characterising the altitude-dependent classification accuracy, they are able to use Bayesian updates to fuse classifications from multiple observations. The proposed planners initially optimise coverage at higher altitudes before focusing on resolving high uncertainty regions at lower altitudes. Bresciani et al. (2021) aim to efficiently classify the extent of Posidinia Oceanis, a species of seagrass, using an AUV. They pre-plan the AUV trajectory using a genetic algorithm, with the algorithm optimising sampling from areas deemed high uncertainty by a GP classifier. Active classification has the capability to identify areas of high classification uncertainty, such as the interface between two classes, where further sampling will most improve the classification model. Bender (2013) utilises prior bathymetric data to plan AUV surveys that are designed to most improve an underlying habitat model, by optimising the placement of an AUV survey template such that it reduces the categorical entropy or uncertainty in the habitat model. This research aims to build on this paper by planning freeform trajectories and by ensuring there is a diverse set of samples contained in the resulting survey.

4.3 Information Objectives

Information objectives are used to guide the AUV towards informative area. These information objectives contain active learning algorithms that reward the AUV for visiting areas that will most improve the model. These rewards are also used for evaluating proposed AUV trajectories. Both feature-based and model-based rewards are implemented. It is possible to also combine rewards to form multi-objective models,

allowing the fusion of model-based and feature-based active learning. These information objectives can utilise the entire AUV trajectory, in order to reward sampling that *together* will improve the model.

The problem can be defined as selecting a subset of locations to sample ($\mathbf{X}_{path}$) from the survey area ($\mathbf{X}_{area}$), such that when the bathymetric features extracted at these locations ($\mathbf{Z}_{path}$) and habitat observations ($\mathbf{Y}_{path}$) are used to train a habitat model $f(z) \rightarrow \hat{y}$, such that the accuracy of this habitat model is maximised:

$$\max_{\mathbf{X}_{path} \subset \mathbf{X}_{area}} \mathbb{E}_{\mathbf{X}_{path}}[f(\mathbf{Z}_{path})] \tag{4.1}$$

4.3.1 Feature Based Reward

Feature-based active learning aims to select data points such that the collected features comprehensively overall feature space of the data. The feature-based reward developed in Chapter 3 is used, which is based on the FF-Active (Geifman and El-Yaniv, 2017) active learning algorithm, which selects the next sample that has the highest feature distance from samples already collected and approximates uniformly sampling across the feature space. It is summarised in the following information objective:

$$R_m = \min_{\forall \mathbf{z}_i \in \mathbf{Z}} D_m(\mathbf{z}_a, \mathbf{z}_i), \tag{4.2}$$

where R_m is the reward, $\mathbf{z}_a$ is the feature currently being evaluated, Z is the set of features being compared against and D_m is the Mahalanobis distance.

Using a feature-based reward have several advantages; they can be used for cold-start active learning as they do not depend on a model, they are not susceptible to overconfident or poorly-performing models and they select samples that together improve the model. A limitation of these approaches is that they are not guaranteed to sample around the decision boundary, which can result in less model improvement.

4.3.2 Model Based Rewards

The model-based rewards use the model's uncertainty estimate to identify what samples will most improve the model. A model is trained using the existing AUV surveys collected in the survey area, with the model inputting the bathymetric features and predicting the benthic habitat. The model is a BNN and therefore there are k Monte Carlo samples of the model, which is used to construct the posterior distribution. These predictions are fed into the reward function which calculates the estimated utility of a sample from that spatial location.

Model-based objectives allow sampling around the decision boundary, which often leads to increased accuracy. Furthermore using the model to drive where to sample is closely related to the objective of the survey - to characterises the benthos efficiently. However, these methods cannot be used to 'cold start' the active learning session, and the seeding data needs to be smartly selected before beginning.

Max Entropy

This objective function selects data points that have the highest entropy in the class predictions (Settles, 2009). This reward function will naturally select data points close to the decision boundaries. Often this leads to an increase in performance due to a refinement of the decision boundary, however it does not promote selecting diverse points. Furthermore, this objective will select points with high aleatoric uncertainty, even though further training with these data points will not significantly improve the model.

The performance of using categorical entropy is reduced in neural networks as they are often overconfident in the prediction (Guo et al., 2017). Utilising BNNs or using calibration procedures can reduce this to some extent.

The predicted entropy $\mathbb{H}$ of a prediction y can be calculated with the following equation:

$$R_{ENT} = \mathbb{H}(\mathbf{y}|\mathbf{z}, \mathcal{D}_{train}) = -\sum \bar{\mathbf{y}} \log(\bar{\mathbf{y}}) \tag{4.3}$$

Where $\bar{\mathbf{y}}$ is the mean output of the k Monte Carlo samples of the classifier.

BALD reward

Bayesian Active Learning by Disagreement (BALD) (Houlsby et al., 2011) aims to estimate the mutual information between the data and the model parameters, by evaluating the following equation:

$$R_{BALD} = \mathbb{I}(y; \omega | z, \mathcal{D}_{\text{train}}) = \mathbb{H}(y | z, \mathcal{D}_{\text{train}}) - \mathbb{E}_{p(\omega|\mathcal{D}_{\text{train}})}[\mathbb{H}(y | z, \omega, \mathcal{D}_{\text{train}})] \tag{4.4}$$

It selects points that are both high entropy and which the model is unsure about, which occurs when the left term is maximised and the right term is minimised. The left term in the above equation is the entropy, which is high when there is high aleatoric uncertainty. The right term is the expectation over each prediction, which is low when the model has differing predictions from each sample. The BALD metric is similar to the epistemic uncertainty measure outlined in Chapter 2, with similar data points being highlighted.

BatchBALD (Kirsch et al., 2019) is a modification of BALD that also approximates the joint entropy of all the points, that allows an entire batch of points to be selected. This joint entropy calculation is too computationally intensive to be run within the planner. Furthermore, BatchBALD has diminishing effectiveness as the batch sizes increase. For this application, the batch sizes can be large. For these reasons, a BatchBALD approach was deemed to be incompatible.

4.3.3 Hybrid Objectives

The limitations of each individual active learning strategy means that relying on a single objective function can lead to sub-optimal performance. Feature-based active learning strategies aim to cover the feature space and do not hone in on decision boundaries, which can lead to slower learning. Model-based strategies are targeted to the mission objective - improving the habitat model, but are not encouraged to

explore the feature space. As they are reliant on a well-trained model, the model-based strategies require a comprehensive seed dataset, while deficiencies in the model can lead to a poorly performing active learning algorithm. A hybrid strategy involves combining model and feature based objectives, with the feature based component responsible for exploration and the model based component driving exploitation (Yin et al., 2017). Furthermore, a hybrid strategy ensures diversity in the batch which is difficult to achieve using only model-based active learning.

A hybrid strategy could be adapted into a reward function in several ways. The simplest way is to add each component reward, while scaling and weighting it. For example if the range of the feature reward is $0 \rightarrow 10$ and the range of the model reward is $0 \rightarrow 1.25$, and the rewards were to be equally weighted, then the feature space should have a scaling factor of 0.1 and the model reward should have a scaling factor of 0.8.

$$R_H = \alpha R_F + \beta R_M \tag{4.5}$$

An alternative strategy is to use the the feature reward to discount the model reward, to prevent double counting. In this strategy the model reward is only accumulated if the feature is sufficiently different to those features already collected.

$$R_H = \begin{cases} R_M, & \text{if } R_F \geq \gamma \\ 0, & \text{otherwise} \end{cases} \tag{4.6}$$

4.3.4 Visualisation of Objectives

The active learning objectives can be visualised in two dimensions to provide a visual interpretation of how each objective selects new data points. This is a snapshot taken midway through one of the dive selection experiments presented in Section 4.5. It uses the labelled dives from the 2008 surveys of the Fortesque region of Tasmania (not the simulated version). At this point there is a model that has been trained on 38% of the available data (1798 data points). The feature space is 33 dimensions but is projected into two dimensions using t-SNE (Van Der Maaten and Hinton, 2008).

This makes the distances between data points no longer meaningful but the overall projection of the data seems to preserve the overall structure effectively. Figure 4.3a shows the visualisation of the feature space, with the data coloured according to its habitat class. The classes are generally grouped together, however there are not clear boundaries between classes which highlights the difficulty of the classification problem.

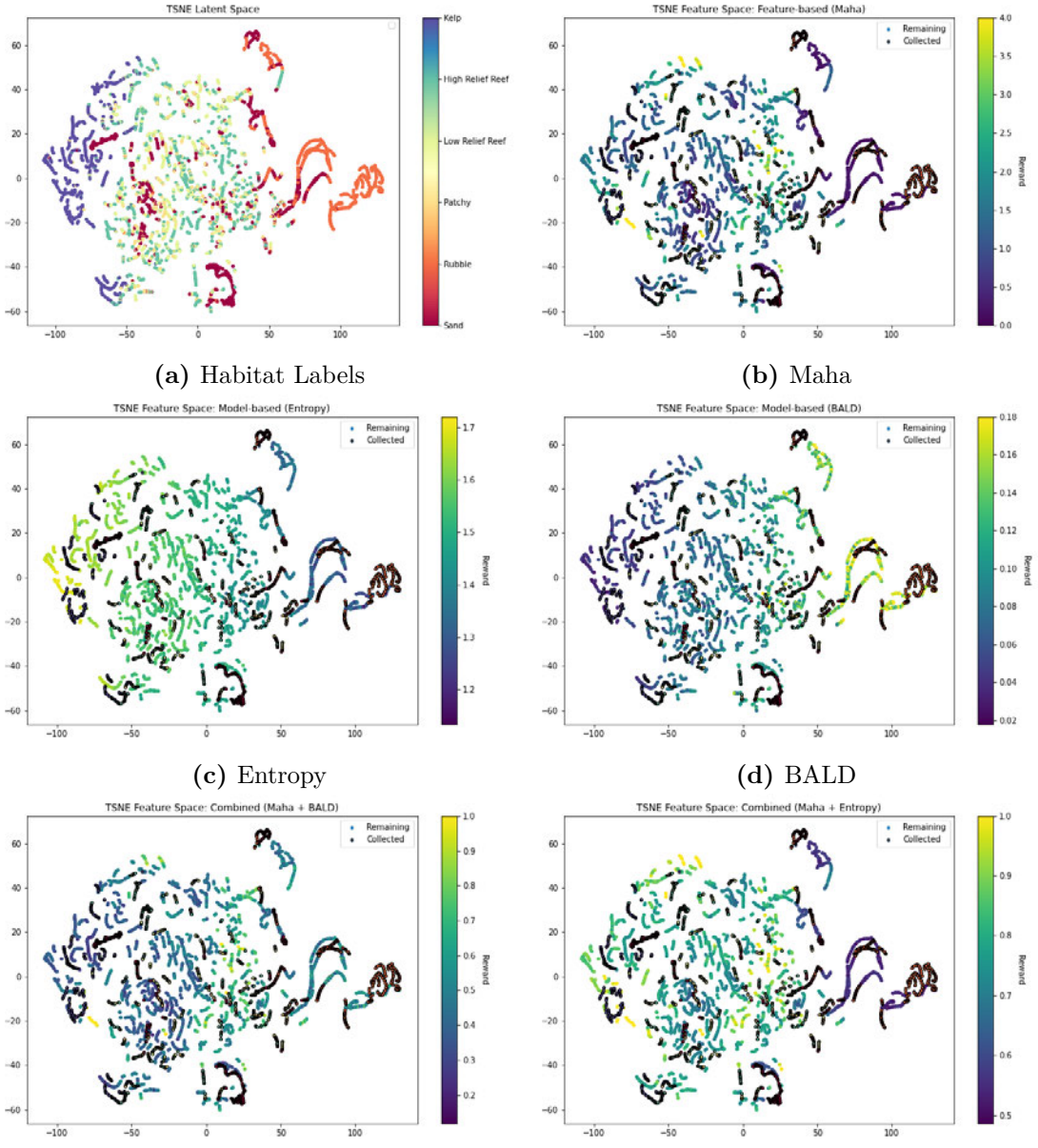
Figure 4.3b shows the reward for the feature-based active learning method, *Maha*. The points already collected are displayed as the habitat classes with a black border (comes across as mostly black), while the remaining points in the dataset are colour coded according to the reward using the viridis colour map (blue is low reward, yellow is high reward). The *Maha* method has a higher reward in areas where the bathymetric terrain is different to those observed before. This is highlighted by the high reward on the outskirts of the feature space however there is also areas of high reward in less dense areas of the centre. The *Maha* reward is designed to constantly search for the next area that is furthest from existing areas, which allows it to cover the feature space effectively.

The model-based objectives are displayed of *Max Entropy* and *BALD* are displayed in Figures 4.3c and 4.3d respectively. The *Max Entropy* objective highlights areas where there is class confusion in the model, which primarily occurs at decision boundaries. This is highest in areas of the feature space where there are a lot of potential classes. The *BALD* objective aims to estimate the mutual information between the model parameters (weights) and the data. It is high where there is class confusion and model uncertainty.

Figures 4.3f and 4.3e show the *Hybrid* objectives, where the model-based objective is combined with the feature-based objective. For the model-based objective, Figure 4.3f using *Max Entropy* and Figure 4.3e using *BALD*. The areas with high reward generally have high model and feature based rewards. By combining objectives, the Hybrid objectives have the capability to overcome an overconfident model. A model can be become overconfident if either the model is incorrectly parameterised or if the training data does not contain enough variation. In these circumstances, a model can

be confident even though the data is out-of-distribution. For these cases, a Hybrid objective can leverage the feature-based method to identify it as highly informative.

Figure 4.3 shows the rewards taken at a snapshot, which means these rewards are static. This does not capture one of the key benefits of the feature-based and hybrid objectives, that the selected batches have high data diversity . When a new data point is acquired, for example in a high reward area, the similar points will have less value as there is now a similar observation. However, it is infeasible to retrain the model for every new data point in modern machine learning models. Furthermore, for this application, the imagery is only labelled after the AUV is recovered and so there is no opportunity to retrain during the AUV survey. For a model-based objective, this will mean the AUV will continue to sample from these high-value areas, as the model has not been updated. This is inefficient use of sampling. The feature-based objective does not use a model, and it continually finds data points that are different than those already observed. This allows the feature-based and hybrid objectives to find points that *together* are informative and prevent oversampling of certain areas.



(a) Habitat Labels **(b)** Maha

(c) Entropy **(d)** BALD

(e) Hybrid (Maha + BALD) **(f)** Hybrid (Maha + Entropy)

Figure 4.3 – Visualising the information objectives. t-SNE (Van Der Maaten and Hinton, 2008) is used to reduce the bathymetric feature space to 2 dimensions. (a) shows the bathymetric feature space with the points coloured according to the habitat labels. (b)(c)(d)(e)(f) each show the feature space with the points colour coded according to the information objective used. This visualisations help to understand how each information objective works.

4.4 Data

The evaluation of active planning techniques benefits from having ground truth datasets to compare against. For the case of broad-scale characterisation with AUVs this would entail a full coverage data set consisting of geo-referenced imagery and bathymetry, however due to the small spatial footprint of imagery, it is infeasible to collect full coverage imagery over the required spatial extents. To address this, this research proposes a method for generating a virtual full coverage survey.

4.4.1 AUV Surveys

The AUV surveys from the 2008 ACFR AUV campaign to Tasmania are used as the baseline. These dives have habitat labels associated with the imagery and there is high-quality bathymetry covering the same area. An overview of these surveys and the process of extracting the labels has been provided in Chapter 2. For this research, the area around the O'Hara and Waterfall reefs are focussed on, which have largely similar areas. From these four dives are used; *O'Hara 07*, *O'Hara 20*, *Waterfall 05* and *Waterfall 06*. The habitat observations are displayed spatially over the bathymetry in Figure 4.5a. These surveys are used for the dive selection experiment in Section 4.5.2 and to create the virtual datasets in 4.4.2.

4.4.2 Virtual Dataset Creation

To create the virtual dataset, bathymetry and labelled georeferenced imagery over the same area are combined. This process is the same as 'cluster assignment' detailed in Chapter 2. The goal of this process is to generate a realistic habitat label for every location in the survey area, enabling the retrieval of a habitat label wherever the AUV was to traverse inside the survey area. In this context, a realistic habitat label means there is a consistent relationship between the bathymetry and the habitat label, that is similar to the real relationship. It is important that these habitat labels

are realistic as the habitat model needs to be able to learn them, while also not being overly simplified.

The process of creating the virtual dataset is shown in Figure 4.4. The inputs to this are the remotely-sensed data and georeferenced habitat labels. Next, an autoencoder is used to extract features from the bathymetry. This autoencoder is different from the one used for habitat modelling. This feature space is clustered using Gaussian Mixture Models with a high number of clusters (e.g. 50), which should be expressive enough to differentiate between types of bathymetric terrain. This outputs a spatial cluster map, where each pixel in the map contains the cluster ID. The spatial location of each of the in-situ habitat observations (from the AUV surveys) is extracted and used to find the cluster ID at that point. For each of the cluster groups, the habitat labels are tallied, resulting in a distribution of habitat labels contained in every cluster. Each cluster is assigned a habitat label based on the modal class within each cluster. If there are no habitat labels contained in the cluster, a similar habitat observation is selected to label the cluster. To do this, the mean of the cluster is compared with all the habitat observations. The habitat observation with the smallest feature distance (most similar) is selected to label the cluster. The final result is a full coverage habitat map that resembles a habitat map created by the neural network based methods.

This method allow the creation of realistic habitat labels for the full extent of the survey area. This is valuable as it allows planning methods to be tested on the entire survey area. A limitation of this approach is that it is not learning the true habitat labels, which are likely to have a weaker relationship between the habitat labels and the remotely-sensed data. The true habitat labels have higher label noise than these simulated labels. Another limitation is that scarce habitat labels can be eliminated from the final habitat map, if they are not the modal class in any cluster. Care was taken to ensure the process for creating the virtual habitat labels and the classification task were different. The autoencoder used for the creation of the habitat labels used a different patch size than the autoencoder used for classification (21×21 versus 11×11), which results in a completely different feature space and makes the classification task harder. Overall, this virtual dataset process is effective

at providing a simulation environment to test the informative planners. This has been utilised in an expanded dive selection experiment in Section 4.5.3 and the active planning experiment in Section 4.7.

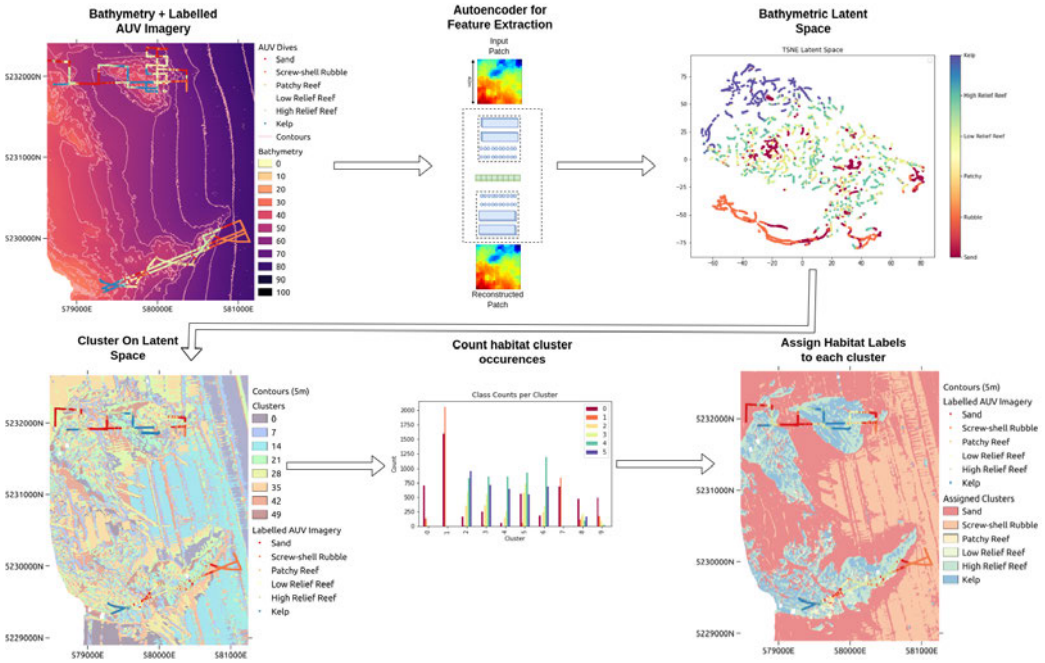


Figure 4.4 – The process for creating a virtual dataset for evaluating active sampling strategies.

4.5 Dive Selection

This dive selection experiment is an example of a pool-based active learning scenario, where the aim is to maximise the model accuracy with the minimum samples. Usually it is used to prioritise labelling efforts, however here the goal is also to prioritise where to send the AUV. This experiment divides existing AUV surveys into many sequential segments and the active learning algorithm needs to select which segments to add to the collected datasets, which are used to train on next. It is designed to be an introduction to the active sampling experiments, but it decouples the information objectives from the informative planners which allows for an evaluation focused on the information objectives.

For active learning algorithms that can consider the informativeness of a batch of data, rather than just individual samples, the dive selection process can utilise this to increase the utility of each batch. This can occur when the planner has to select multiple segments at each planning iteration. To select multiple informative segments, a greedy approach is used. For each selection, the segment that maximises the information reward is selected. The next selection is then conditioned on this segment and will select the next segment so that they together optimise the information objective. The results indicate a significant improvement in performance, however it dramatically increases the runtime due to the large size of the pool datasets.

Two dive selection experiments are run. The first uses existing AUV dives over the area. The second uses the virtual dataset to create a broad grid over the survey area, which is more reflective of the active sampling problem, as the existing AUV dives were already too focused on informative areas and are limited in size.

4.5.1 Dataset Preparation

The dive selection datasets are created by first segmenting the input datasets into many sequential segments. For the 2008 AUV data, the *O'Hara 07*, *O'Hara 20*, *Waterfall 05* and *Waterfall 20* are used as inputs and 200 sequential segments are used. The virtual dataset is a simulated broad grid covering the O'Hara / Waterfall area with 500 sequential segments. For each active learning run, a unique seed dataset is created that is made up of several segments and together is coarsely balanced. The leftover segments are added to the data pool for that run. The validation and test datasets each comprise 10% of the total segments and are filtered to remove points that are within the same patch as a training dataset point. Each of the algorithms being evaluated uses the same datasets. The dataset split is displayed spatially in Figure 4.5b.

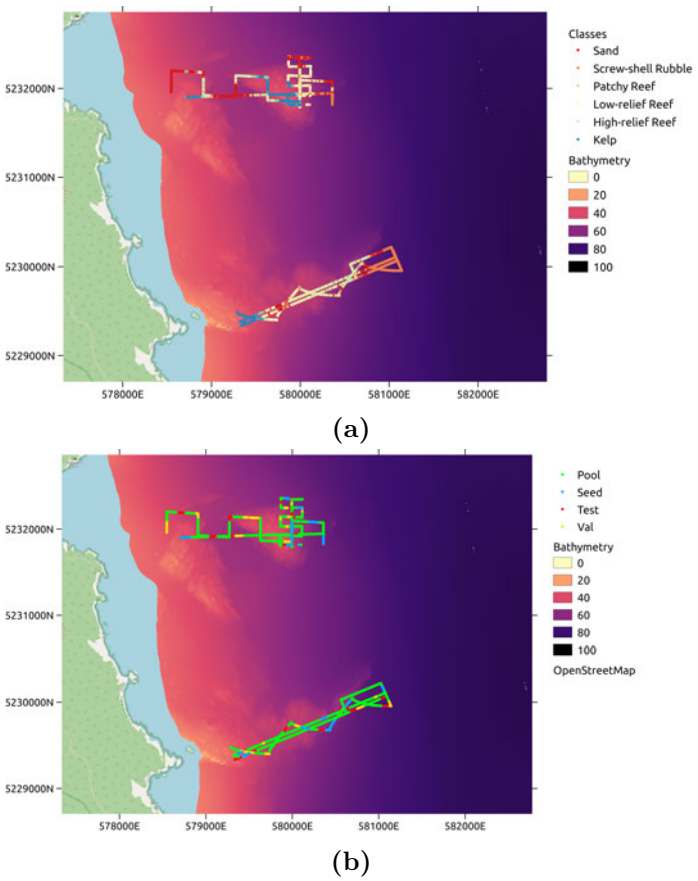


Figure 4.5 – Data collected by the ACFR AUV Sirius in 2008 near Fortesque, Tasmania is used for the dive selection task. (a) shows the labelled imagery covering the O'Hara and Waterfall reefs. (b) displays how the dives are segmented into seed, pool, validation and test datasets. The length of each segment is 95m and there is a 25m gap between each segment. Coordinates are displayed in UTM zone 55S.

4.5.2 Dive selection using existing AUV surveys

At the start of each run, the habitat model is first trained on the seed dataset. Next, the information objective selects what dataset segments to add to its training dataset collection, from the available dataset pool. It does this by evaluating each dataset segment in the pool to identify which datasets have the highest value. The ten highest value dataset segments are added to the training pool at each iteration. This process continues for 10 iterations.

Five runs are conducted for each algorithm, with each run having a unique seed dataset. Each run takes approximately two days to complete for each algorithm.

After the experiment is completed, the habitat model from each iteration is evaluated on the test dataset. Figure 4.6 shows the accuracy at each iteration of the active learning planner, averaged over the five runs.

For this experiment, all the methods follow a similar growth trajectory. In the earlier iterations, there is a slight advantage to the *BALD* and *Maha* methods. Towards the final iterations, the *Hybrid* strategy performs best, although the difference is small. It is difficult to show much improvement using active learning in this experiment as it created segments within an already informative survey. These surveys were manually planned to best characterise this area. Therefore it is not representative of the active survey planning problem. In this scenario, a random action is often a good selection as it is drawn from an informative survey. In general, a random action would often be poor, due to the different distributions of habitats and the relatively small portion of the area composed of reef habitats.

4.5.3 Active selection using a virtual dataset

This experiment repeats the dive selection process above, but with a virtual dataset to allow for simulated habitat classes to be retrieved for the entire survey area. The survey area for this experiment is the area covering the Waterfall and O'Hara reefs. A broad grid is placed over the entire survey area, running from west to east, with

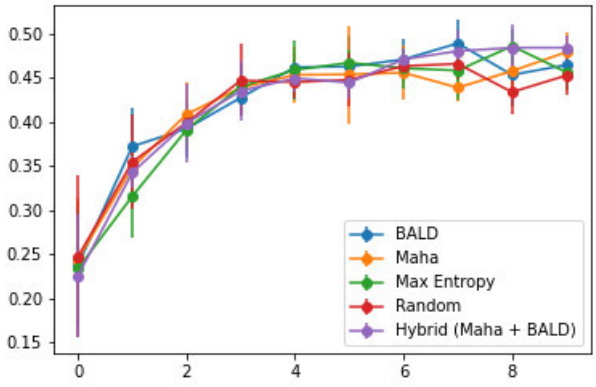


Figure 4.6 – Accuracy at each iteration of the dive selection cycle, where the task was to select what segments of the AUV surveys should be used for training. Previous AUV surveys from the Fortesque region of Tasmania were used.

a path separation of 200m. It is clipped near the edge of the remotely-sensed data. The northern-most line is 3267m long, while the southern-most is 2144m. The broad grid and the simulated habitat labels collected along it are presented in Figure 4.5a.

The process of creating the seed, pool, validation and testing datasets is identical to the method using past AUV surveys. For this dataset, the broad grid is broken up into 500 segments, each averaging 100m in length. The combination of seed, pool, validation and testing datasets for the first run (each run has different pool and seed datasets) is displayed in Figure 4.5b. As shown in Figure 4.7c, there is a 25m gap between each segment.

With the larger data quantity compared to the real AUV surveys, a larger seed dataset can be used, allowing the model-based methods to perform better in early iterations. The *BALD* planner performs marginally better than the *Max Entropy* objective.

The *Hybrid* approach exhibits a significant performance increase over other methods. The model is driving the selection of informative segments, while the feature component is ensuring diversity among segments. The improved performance of this combined approach over model-only based methods (*BALD*, *Max Entropy*) highlight the benefit of ensuring data diversity within the batches.

The *Random* strategy performs worse using the virtual dataset than using the past AUV surveys, as the likelihood of picking an uninformative segment is increased. In this experiment, many segments contain primarily sand or rubble, which are generally easy to classify, making these segments overall uninformative. A random planner is more likely to select these segments than other more informative segments. This better reflects the composition of the survey area than the previous experiment and highlights the need for an informative planner.

The dive selection experiments highlight the benefit of informative surveys. In particular, a *Hybrid* approach that leverages both model and feature based active learning is effective at visiting highly informative areas while also ensuring there is diversity amongst the samples.

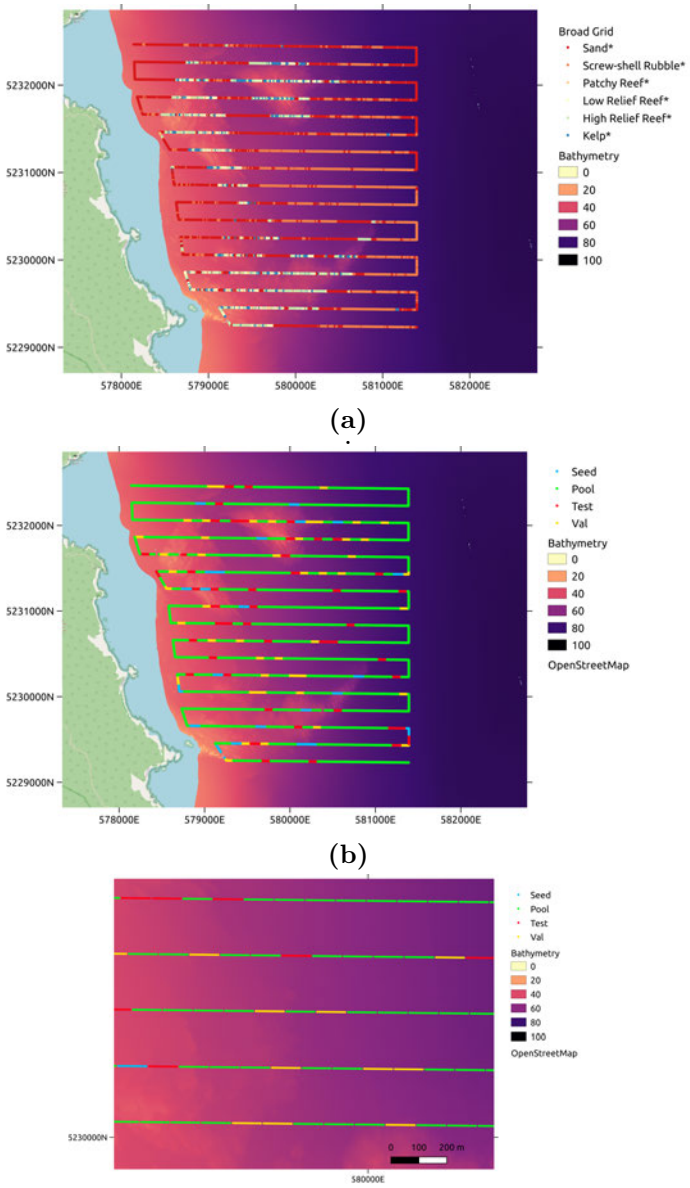


Figure 4.7 – A virtual dataset is used to simulate a broad grid over the survey area. (a) displays the virtual classes, (b) shows the seed, pool, validation and testing datasets for the first run and (c) zooms in on the individual segments. Each segment is 100m long with a 25m gap between each segment. Coordinates are displayed in UTM zone 55S.

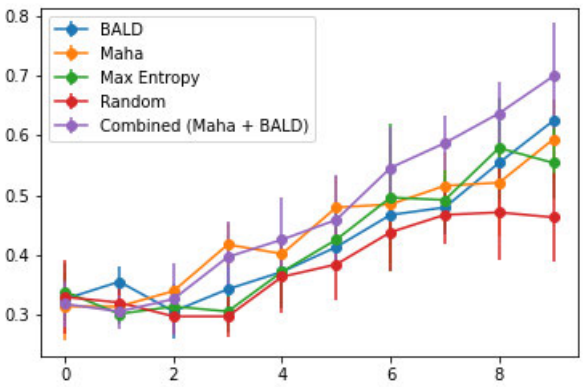


Figure 4.8 – Accuracy at each iteration of the dive selection cycle when using the simulated dataset, where there is more potential datasets covering a representative area of the survey area. The Hybrid objective exhibits a significant performance improvement over the other

4.6 Information Gathering Planner - MCTS

Several information gathering planners were evaluated in Chapter 3. Both the MCTS and RRT based planners used the information metric to plan freeform paths and were shown to be effective at exploring the feature space. Both these planners can be adapted to optimise the any of the information objectives detailed in Section 4.3. For this chapter, the MCTS planner is selected as it is robust and demonstrated better performance on the model-based objectives, as it was able to focus on hotspots more accurately.

4.7 Active Planning Cycle

This experiment simulates a multi-deployment AUV survey where the objective is to create an accurate habitat map of the survey area. The cycle is started by collecting an initial seed survey. At each point on the AUV's path, the georeferenced habitat label is extracted. The habitat model is then trained using all the data collected thus far. The training process uses the location of each habitat label to extract the bathymetry around that point, which is then run through the feature extraction process and used as the input to the probabilistic habitat model, which learns to estimate the habitat class. The information gathering planner then plans the next survey, using the habitat model and previous paths. In this experiment, this experiment is repeated 5 times. This cycle is displayed in Figure 4.9.

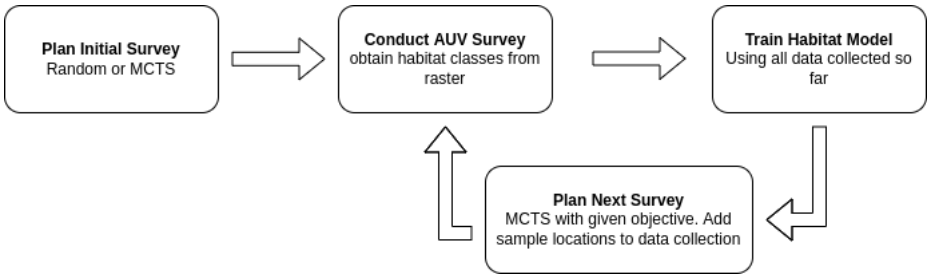


Figure 4.9 – The active planning cycle used for these experiments. First, an initial survey is conducted, which is either a random or informative survey. From here the cycle starts. First the virtual AUV survey is conducted, which involves collecting the habitat observations from the label raster from all the sampling locations collected thus far. Next the habitat model is trained, using the collected sampling locations and corresponding habitat observations. Then the planner is used to plan the next trajectory, using the informative objectives. This can involve using the current habitat model, past sampling locations or both. This cycle is repeated for the remainder of the campaign, which for these experiments is 5 cycles.

Two different seed surveys are used, to highlight the validity of the approach when using both poor and informative seeds. The first survey uses a survey collected with the *Random Walk* method which does not effectively explore the feature space and the initial habitat model has poor accuracy. The second survey is an informative survey, collected using the feature-based objective, which does not rely upon a habitat model and can therefore be used to cold-start the active learning process (see Chapter 3). This survey visits all the habitat classes and the resulting habitat model achieves good accuracy. Each seed survey is 2500m long. Each seed survey can be visualised in Figure 4.10.

4.7.1 Benchmark: Random Walk

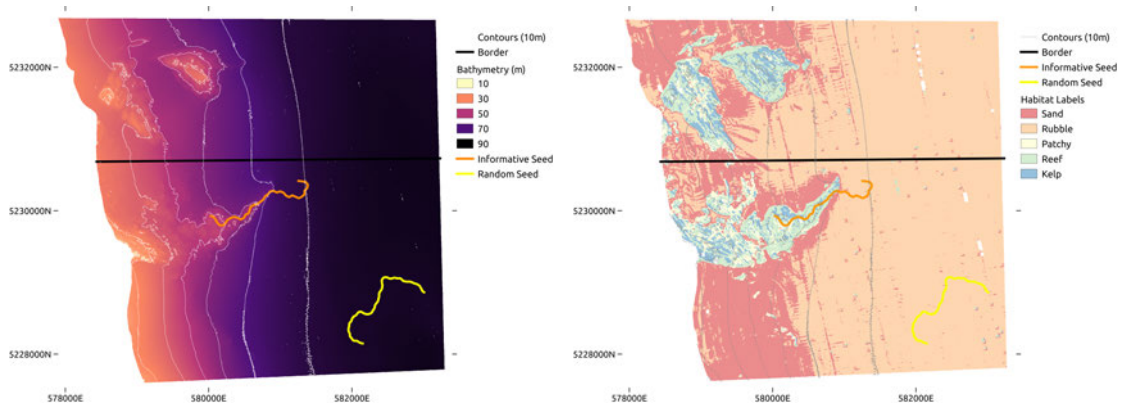
The information gathering planner is benchmarked against a *Random Walk*. For this application, the *Random Walk* selects a set of connected waypoints by meandering around the survey area, without utilising the prior bathymetry or the information objectives.

First a starting position is randomly selected within the survey area. The next position is found by expanding in a random direction (within a given angle of the current path) from the current position. This process is repeated until the length of the path reaches the distance budget for the path.

4.7.2 Data

The O'Hara and Waterfall regions are used to create the virtual dataset. Due to the scarcity of the 'low relief reef' class in the 6 class dataset, both the 'low relief reef' and 'high relief reef' classes have been combined into a single class. As identified in Chapter 2, these habitat labels are part of a hierarchical labelling scheme, where the 'reef' habitat (from the 4 class dataset) is a superset of the 'low relief' and 'high relief' habitats.

For these experiments, the planners can operate in the O'Hara region while the Waterfall region is used for validation. This region split between training and validation has been used in Chapter 2, ensuring there is no overlap between training and validation. The O'Hara and Waterfall should have a relatively similar relationship between bathymetry and habitat. These areas are shown in Figure 4.10.



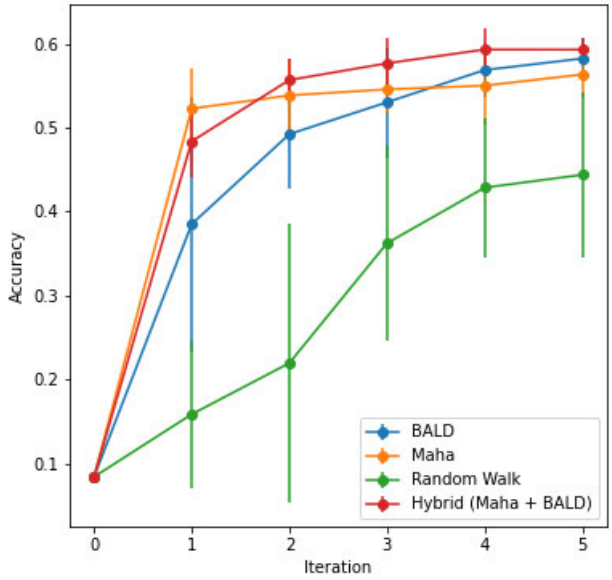
(a) (b)

Figure 4.10 – The seed surveys around the O'Hara / Waterfall reefs near Fortesque, Tasmania. The yellow line is the *Random Walk* seed and the orange line is the informative initial survey (using the strategies from Chapter 3). Left is the bathymetry, right is the virtual habitat labels. The area around O'Hara reef, south of the red line, is the planning area. The area around Waterfall reef, north of the black line, is used for validation. Coordinates are displayed in UTM zone 55S.

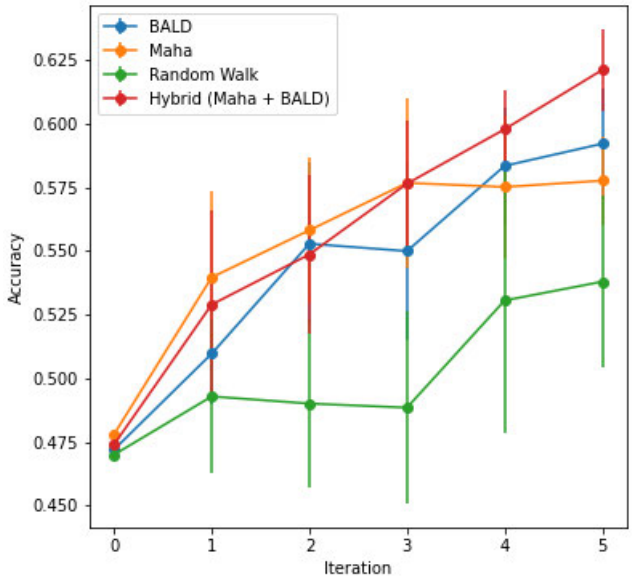
4.7.3 Results

Figure 4.11 displays the results from the active learning experiment, with Figure 4.11a showing the results from using a random walk seed and Figure 4.11b showing the results from an informative seed path, collected with the *Maha* method which can be cold-started. The planned paths for a single run are also visualised in Figure 4.12. Iteration 0 is the starting iteration, with all the models trained with just the seed dataset. At Iteration 1, each planner has planned a single 2000m AUV trajectory and the observations from this trajectory have been added to the total collected observations (including the seed dataset). After the first planning iteration (Iteration 1), the *Maha* method performs well as the feature-based method is designed

to explore the feature space, which allows it to efficiently complement the seed dataset. As the cycles progress, the *Maha* method performs relatively poorly, as the feature space is sufficiently explored and the feature-based method begins to drive towards outlier points. This does not align with the distribution of the habitat classes and as such poor accuracy is obtained. The *BALD* method is slow to learn but reaches a high accuracy by the final iteration. This slow pace of learning is caused by both inadequacies in the model and a lack of diversity of sampling points. Initially, there is limited training data and as such the model is insufficiently trained. As the *BALD* method is model-based, the poor underlying model is not able to generate realistic uncertainties which results in suboptimal trajectories. This was identified in Chapter 2. Secondly, the *BALD* method does not consider the relationship between samples which can make it sample similar high uncertainty areas along the same trajectory, even though observing these areas just once would have been sufficient. *Random* sampling is an ineffective strategy for informative sampling, due to the large spatial extents of the survey area and relatively small high information areas. The *Hybrid* approach leverages both the feature-based and model-based approaches and aims to combine these approaches to allow for both exploration of the feature space and exploitation of the model uncertainty. Initially the *Hybrid* approach learns quickly, though not as quickly as the purely feature-based method. As the underlying model improves, the *Hybrid* approach consistently outperforms the other approaches, as it is both exploring the feature-space, focusing on the decision boundary and ensuring there is diversity amongst the samples collected.



(a)



(b)

Figure 4.11 – Accuracy at each iteration of the active planning cycle, averaged over five runs with the bars showing the standard deviation. (a) shows the accuracy when the planner is initialised with a random seed, while (b) shows the accuracy when initialised with an informative seed. There is more room for improvement when using a random seed. Initially, the feature-based objective (*Maha*) performs the best as it is critical to explore the feature space at this stage. As the iterations progress, the *Hybrid* objective outperforms all the other methods, as it both explores the feature space and visits areas of high model uncertainty, while also ensuring there is diversity amongst the samples collected.

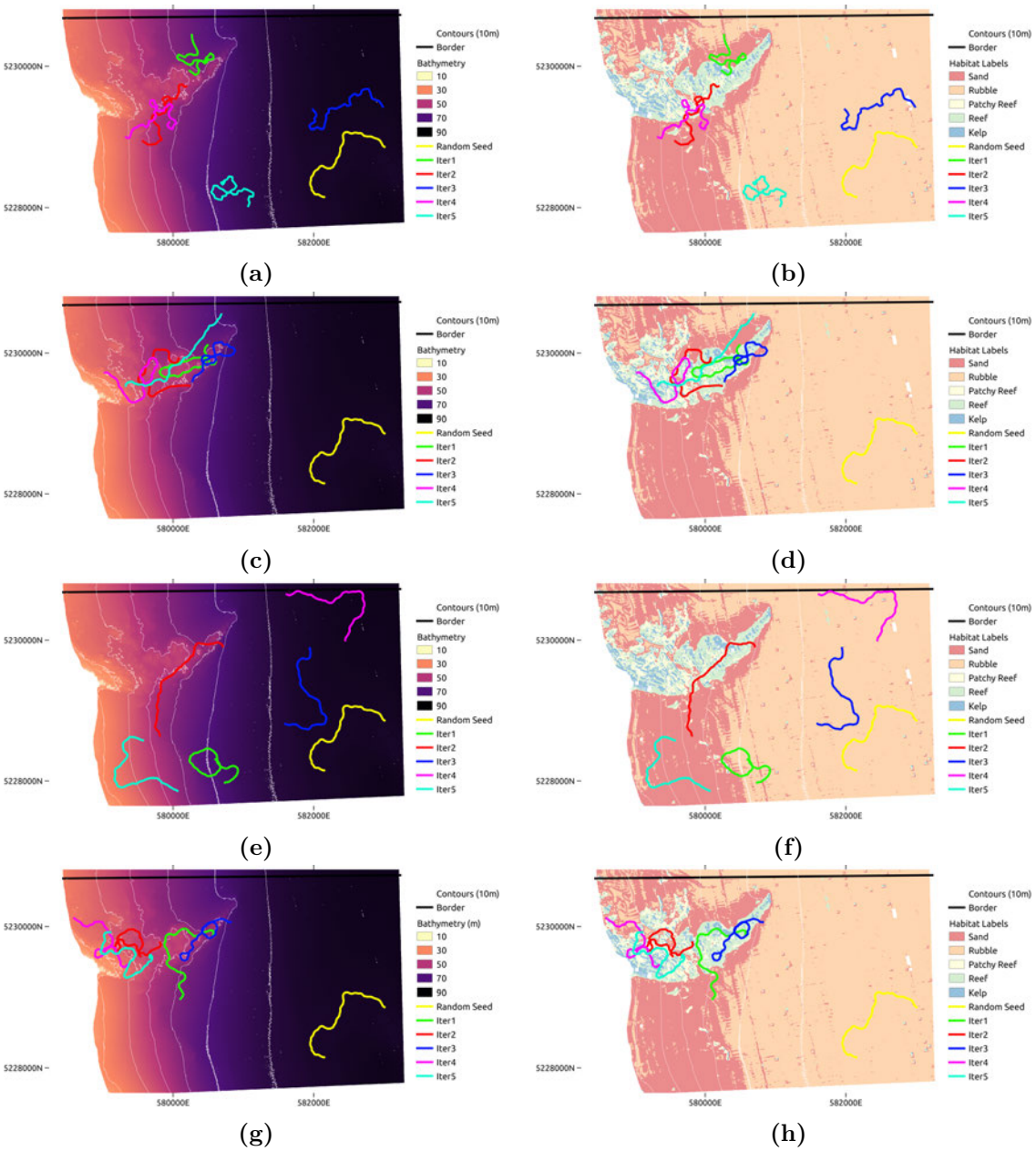


Figure 4.12 – Paths planned over each of the five iterations of the active planning cycle. Each row shows the paths planned for each objective. The left column shows the paths planned over the bathymetry, while the right column shows the same paths planned over the habitat classes. (a)(b) shows paths plannned using the *BALD* objective, (c)(d) using the *Maha* method, (e)(f) using *Random Walk* and (g)(h) using the *Hybrid* objective. The colours of each path correspond to the iteration. Coordinates are displayed in UTM zone 55S.

4.8 Proposed Field Deployments

The active planning procedure proposed in this chapter are evaluated using a set of field deployments around the Jibbon Point area. This area was selected due to the availability of prior AUV surveys collected by AUV Sirius. The prior AUV surveys allow the habitat model to be seeded, without first conducting an initial survey (i.e. in Chapter 3), which reduces the cost of completing this evaluation. The objective of the surveys is to improve the habitat model using smart and efficient sampling.

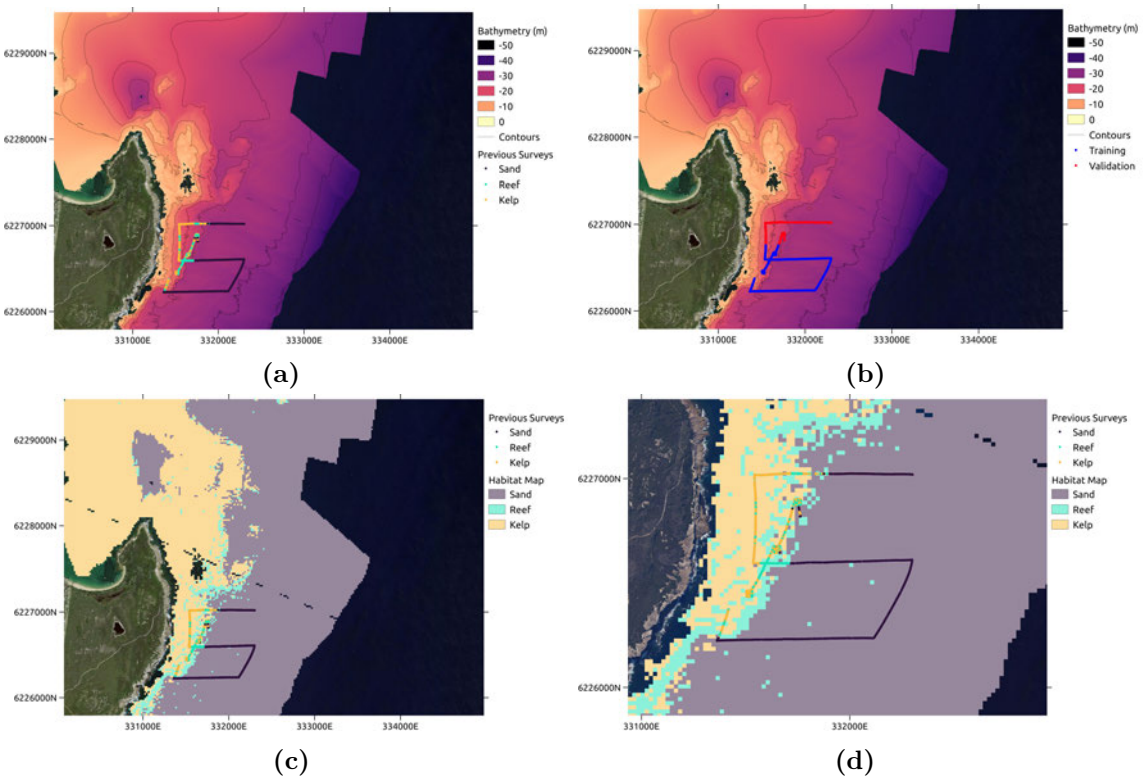


Figure 4.13 – The proposed survey area located around Jibbon Point, NSW, Australia. Previous AUV surveys conducted in 2012 provide the initial seed dataset to seed the habitat model. These previous surveys are overlaid onto the bathymetry in (a). These surveys were split into North and South components, with the North being used for validation and the South being used for training, which is displayed in (b). The habitat maps created with this data is displayed in (c) and (d). The habitat maps are accurate in the areas around where the previous surveys were conducted, but become inaccurate when these models are inferred on a broader area. Coordinates are displayed in UTM zone 56S.

The prior AUV surveys were collected by ACFR AUV Sirius in 2012. The first survey

Jibbon01 consists of several dense grids connected by straight transects. The second survey *Jibbon02* is a broad grid covering a similar area. Three broad habitat categories were observed; sand, reef and kelp. These classes are displayed in Figure 4.14 and plotted spatially in Figures 4.13c and 4.13d. To divide the samples into training and validation components, the dives are split into Northern and Southern sections. This was necessary as *Jibbon01* was primarily dense grids and did not cover the required spatial extent. The Northern sections of each dataset are used for validation, while the Southern is used for training. This is plotted spatially in 4.13b. There is a 25m gap between the Northern and Southern paths.

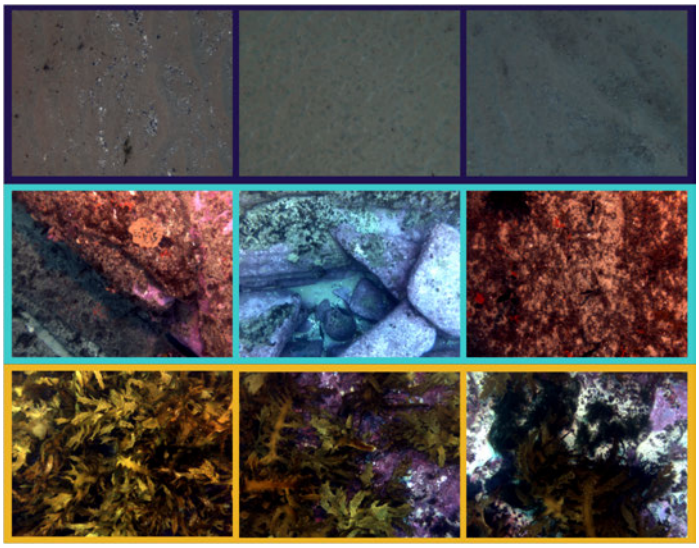


Figure 4.14 – Example images from each habitat class, as captured by ACFR AUV Sirius in 2012. Top row, sand. Middle row, reef. Bottom row, kelp. The colours correspond to the spatial plots in Figure 4.13

The prior AUV surveys are used to train a habitat model using a Bayesian neural network. The model achieves a validation accuracy of 80%, which is a very high accuracy for this task, indicating the habitat model is accurate for the area around the prior AUV paths.

The habitat model can be used to infer over the larger Port Hacking / Jibbon Point area. However, the paths collected in 2012 are not representative of this larger area and do not sufficiently cover the feature space. This makes the larger area mostly out-of-distribution. The habitat map 4.13c appears accurate near where the AUV

surveys have been conducted, but inaccurate in other areas. For the training and validation paths, everything shallower than ∼20m was kelp, which the habitat model learnt and inferred that to the larger area, however, this was incorrect. Similarly, everything deeper than ∼30m was classified as sand, which also is inaccurate for the larger area.

This can be further analysed by plotting the entropy and also the model uncertainty over the survey area. The entropy map highlights areas of high class confusion including the decision boundaries. The reef/kelp interface in the decision boundary still has high entropy, despite extensive sampling which identifies it as difficult to resolve. The uncertainty of the model is higher when each prediction of the model is different. This is limited in the areas containing the reef/kelp interface but it does highlight the perceived sand/kelp interface that is not represented in the training dataset. This highlights that further sampling in areas where the uncertainty is high should improve the habitat model, as it should resolve the inaccurate classifications around the sand/kelp interface.

Problematically, the shallower areas are not highlighted by the model uncertainty despite being misclassified (Figure 4.13c and Figure 4.15c). In the training dataset, all the shallower areas are kelp so when even shallower areas are presented, the model extrapolates and firmly predicts these areas as kelp. As the other categories are apparently unlikely, sampling from the model does not predict another class and therefore the model has low uncertainty. This is a weakness in relying upon a model to drive further sampling, as found in Chapter 2. Although the model has failed to identify some of the out-of-distribution areas as uncertain, these do get highlighted by the feature-based approaches. Figure 4.15d plots the relative feature distance from the features on the training path to each point on the map. The areas with higher feature distances are unique to features found in the training dataset. A feature-based active learning algorithm such as FF-Active (Geifman and El-Yaniv, 2017) will prioritise those areas for further sampling.

Several surveys are designed to sparsely sample the areas surrounding the larger Jibbon Point area. The objective of each deployment is to improve the habitat model.

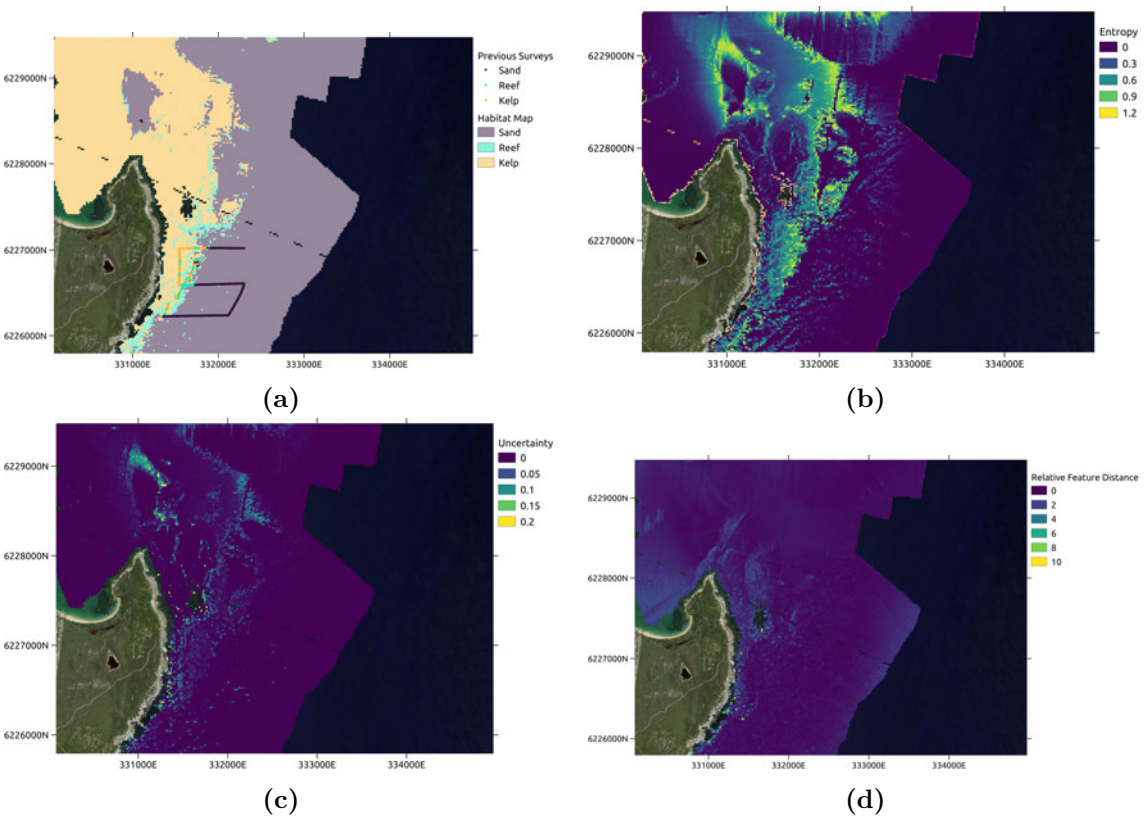
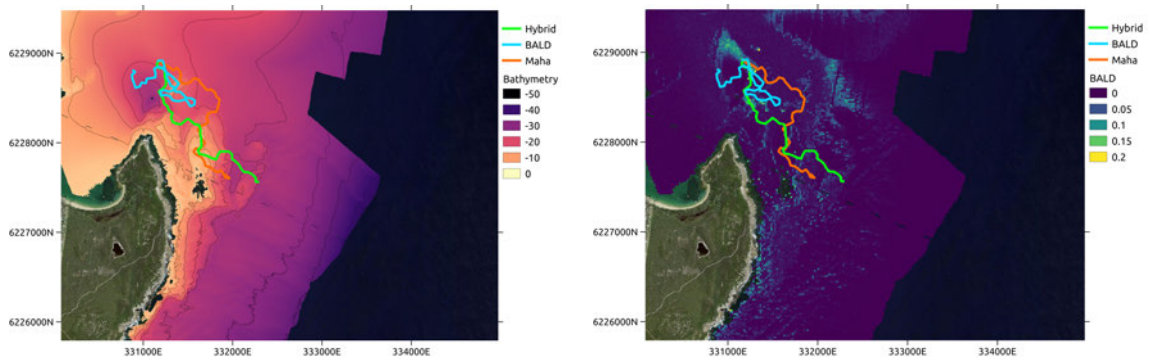


Figure 4.15 – Reward maps of the survey area off Jibbon Point, NSW, Australia. (a) habitat predictions for reference, (b) entropy, (c) BALD prediction and (d) relative feature distance. There is a high reward on the predicted sand-kelp interface by the model-based methods, as this is not represented in the training dataset. The model is overly confident and predicts low reward on the shallow, unobserved areas (mid-left), where it predicts kelp despite it being sand. Coordinates are displayed in UTM zone 56S.

These deployments are designed to evaluate the different active sampling strategies, including a model-driven strategy (*BALD*), a feature-driven strategy (*Maha*), a hybrid strategy (*Hybrid*) and a random strategy (*Random Walk*). All the paths start at the same location, which is automatically selected as the position which has the largest BALD values in the surrounding locations. The *Maha* and *Random Walk* strategies would not normally be allowed to access this information, but having the same starting position for all the strategies allows for easier comparison between the approaches as the starting position has a significant influence on the success of the survey. Due to the high cost in conducting AUV surveys, a single iteration of active planning is proposed to evaluate this research. The planned surveys are displayed in Figure 4.16.



(a) **(b)**

Figure 4.16 – Planned deployments in the survey area off Jibbon Point, NSW, Australia, where the objective is to improve the habitat model. All the methods share the same starting point, but take significantly different trajectories. Coordinates are displayed in UTM zone 56S.

Before the surveys are conducted, the informativeness of the survey can be predicted. The cumulative BALD reward can be used to estimate the informativeness of a survey through the lens of a model-based approach. The cumulative BALD reward is plotted in Figure 4.17a. This shows the model driven survey using a BALD reward accumulates the most reward. However, the model-only driven planner does not consider the relationship between samples and therefore the samples can be similar and have less information when considered together. Figure 4.17b shows the accumulated *Maha* reward, which identifies that the *BALD* only strategy does not explore diverse terrain and mostly focuses on the same area. If the BALD reward is only accumulated when

the point is sufficiently unique to points previously visited, it captures how samples collected along a survey path are *together* informative, which is plotted in Figure 4.17c. The cumulative reward shows that a model-only driven survey appears to visit high uncertainty areas but these areas are not unique so the informativeness of the survey is limited. By both optimising a feature and model-based reward, the survey is able to visit high uncertainty areas that are unique and therefore collect a set of samples that are together informative.

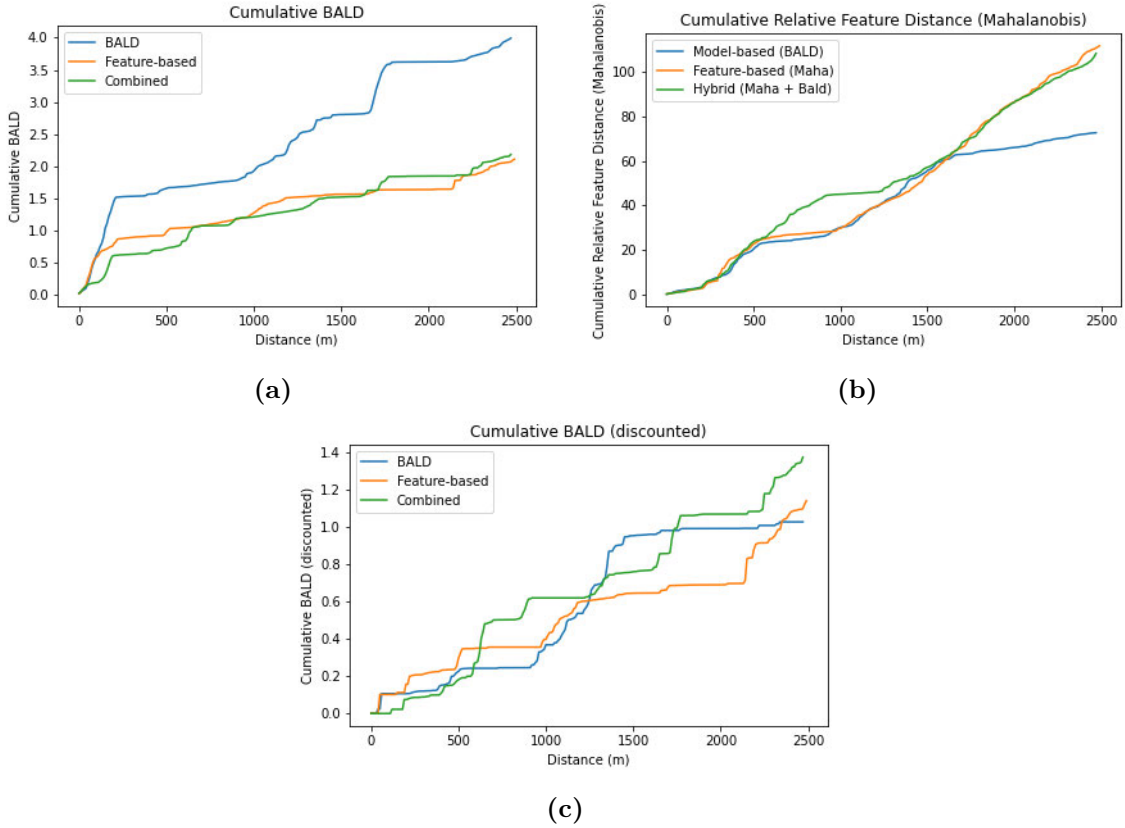


Figure 4.17 – Plots of the cumulative reward for the planned deployments. (a) shows the cumulative BALD reward, where the *BALD* strategy accumulates significantly more reward. (b) shows the cumulative *Maha* reward, which identifies that the *BALD* strategy does not explore diverse terrain. (c) plots the cumulative BALD reward, which only accumulates if the features are sufficiently different. Here, the Hybrid strategy outperforms the other methods.

As of writing these proposed surveys could not be run. They were attempted however the AUV was not working correctly and only collected a small portion of the proposed paths.

4.9 Summary

The objective of this chapter was to investigate planning AUV surveys such that accurate habitat models can be developed, while minimising the AUV deployment costs. While Chapter 3 investigated planning informative surveys before any surveys had been completed, this chapter explored planning surveys to complement the existing surveys, which can together be used to generate an accurate habitat model. The approach to this was to link active learning methods to adaptive planning, which enabled prioritised sampling in areas that will most improve the habitat model. Both model-based and feature-based active learning methods were evaluated. Model-based techniques use the probabilistic habitat model's uncertainty estimate to sample in areas the model is most unsure about or that will most improve the model. Feature-based methods aim to completely explore the feature space (input to the habitat model), by selecting areas to explore that are different to those already observed. A hybrid strategy combines the model-based and feature-based approaches. It is beneficial as the model component directs it to areas that will most improve the model, while the feature component ensures there is diversity amongst the batch and can compensate for an overconfident model. These active learning-based strategies can be integrated into the informative planners proposed in Chapter 3.

A simulated dataset is used for evaluation as a large area covered with remotely-sensed data with corresponding habitat labels is needed. Using these datasets, there is a significant performance boost from using the hybrid strategy over other strategies. Compared to a *Random Walk*, there is a dramatic improvement. This demonstrates the benefit of using informative sampling for improving a benthic habitat model. The experiments showed that different strategies performed well at different stages of the active learning cycle. Initially, the feature-based active learning strategy performed the best as exploring the feature space was necessary given the current coverage of the feature space. As the number of samples increased, it became more critical to focus on the decision boundaries and areas that the model was highly uncertain of. During this phase, the *Hybrid* strategy performed the best, as it used the model component to focus on areas that will most improve the model, while the feature

space component ensured diversity amongst the batches. Using these insights, future informative planners could adjust the information objective based on the number of samples collected, current model accuracy or exploration of the feature space.

Chapter 5

Conclusions

This thesis investigates informative path planning for benthic AUV surveys with the overarching goal to efficiently collect habitat observations of the seafloor such that habitat models can be made. Following the introduction in Chapter 1, Chapter 2 developed probabilistic classification models for habitat modelling, that can be utilised to drive further sampling. Chapter 3 investigated the problem of planning an initial informative AUV survey. It leveraged the remotely-sensed data to plan an survey that efficiently characterised the bathymetric terrain. Finally, Chapter 4 explored linking informative sampling with active learning to efficiently create habitat models with a limited sampling budget. Together, these chapters form the overarching contribution of this thesis, which is the development of active learning-based sample prioritisation procedures for AUV surveys. The core contributions presented in these chapters are summarised below, as well as identifying potential avenues of future research.

5.1 Contributions

5.1.1 Bathymetric Feature Extraction

The seafloor structure can be indicative of the benthic habitat, and the geometric features such as depth, slope, rugosity and aspect are commonly used to learn this relationship. In Chapter 2, autoencoders are used to compress a bathymetric patch into a latent feature that is representative of the seafloor structure. This feature space can be used to improve the performance of unsupervised clustering and also improve the accuracy of habitat models when compared to geometric features. Furthermore, this feature extraction method is used for planning informative surveys in Chapters 3 and 4.

5.1.2 Bayesian Neural Networks for Habitat Modelling

Identifying areas that are undersampled is one of the key requirements for the habitat model in this thesis. Chapter 2 proposes using BNNs for habitat modelling. By performing Monte-Carlo sampling of the network, the posterior distribution can be estimated and the variance of this distribution can be deconstructed into epistemic and aleatoric components. The epistemic component highlights areas that the model is uncertain about and that can be resolved from further sampling, providing an avenue for model driven sampling. An analysis of the uncertainty estimates indicate they are realistic, however they can be overconfident in some cases.

5.1.3 Feature space exploration of remotely-sensed data for informative sampling

Chapter 3 approaches the problem of planning informative AUV deployments of a new survey area. To plan an informative survey, the previously obtained remotely-sensed data can be utilised, which for bathymetry is indicative of the benthic habitat. Using

a feature space representation of the remotely-sensed data, this research proposes using a feature-based active learning strategy to reward sampling features that are different to samples already collected. This strategy approximates uniform sampling of the feature space.

5.1.4 Informative path planners

Informative path planners were developed to leverage information objectives, in order to plan informative AUV trajectories. Benthic AUV surveys are typically planned manually by experts who utilise the remotely-sensed data. Alternatively, surveys can be planned automatically. Existing research has primarily focussed on optimally placing survey templates. This research develops novel information gathering planners, based on RRT and MCTS that explore the feature space by planning freeform trajectories in physical space. Uniquely, they plan trajectories in which the collected samples are together informative. These planners can be used to sample sparsely but effectively from a given survey area.

5.1.5 AUV campaign as active learning cycle

Chapter 4 formulates an AUV campaign as a sequence of deployments with the objective to best characterise the survey area and create the most accurate habitat model. With this formulation, active learning is linked to informative path planning, to enable the planning of trajectories that will most improve an underlying model.

5.1.6 Combining feature and model-based active learning techniques for informative path planning

Model-based active learning methods are effective at directing sampling towards areas that will most improve the model. However, with these techniques it is difficult to ensure diversity of samples in each batch which can result in oversampling of certain

areas, at the expense of sampling in more informative areas. Furthermore, models can be overconfident resulting in misdirected sampling. This research proposes using hybrid objectives, as it allows the habitat model to sample from high value areas, but also ensures diversity amongst the samples collected and explores OOD areas that the model uncertainty misses. This technique outperformed random, model and feature-based active learning techniques in simulated active sampling experiments.

5.2 Future Work

5.2.1 Integration of further scientific objectives into survey planners

Chapter 3 introduced several informative AUV trajectory planner. In the process of planning the fieldwork, it was identified that the planner could take the AUV to steep areas that it could not traverse. To remedy that, constraints were added that prevented the AUV from going up to steep a slope.

It has become evident in planning further fieldwork for these planners that it is important to be able to integrate different scientific objectives. For example, when planning deployments in the North sea (that did not eventuate due to an AUV technical error), the scientists planning the mission wanted the AUV tracks to crossover the previous surveys. This was solved for these surveys in a suboptimal manner, by planning surveys starting from the existing AUV surveys. However, an integrated solution to this problem is needed, where scientists or other partners can request the AUV visit points of interest.

In Chapter 3, the approach to exploring the feature space was to approximate uniform sampling of the feature space of the survey area. However, this approach could be modified to suit different objectives. For example, modifying the distance function could allow scientists to better target the sampling towards specific objectives, such as targeting a specific depth range.

5.2.2 Inclusion of more scientific observations in the feature space

This thesis uses only bathymetry as the only remotely-sensed data source, as it is widely available and be able to easily observe the seafloor terrain, which is an indicator of the benthic habitat. However, the bathymetry cannot be used to perfectly predict the benthic habitat and there are many other factors that influence the benthic habitat. Future work aims to incorporate further observations into the feature vector, to improve classification accuracy and exploration. Observations of water currents, temperature and waves can be collected at scale and can be readily used for this application. The sampling efforts in this thesis have primarily focused on localised areas where these observations would remain relatively consistent. Expansion of these sampling strategies to larger spatial areas would need to account for these extra factors. These observations change over time, so strategies will need to be developed that summarise these changes effectively and allow their integration into the feature vector.

5.2.3 Fine-scale Optimisation of Survey Trajectory

The trajectories that are generated from RRT and MCTS are relatively coarse, with a distance between each point of $\sim 20m$, which is necessary to be able to plan over larger areas. It is possible that the AUV can miss interesting features on a local level. A post planner optimisation of the path could be used to further improve the trajectories, by slightly perturbing the path to identify if that improves the expected reward of the path. This could also be used to refine the trajectories to be smoother, which would allow easier following from the AUV.

5.2.4 Tradeoff between information and cost

Chapter 3 studied how to explore a feature space representation of the survey area, while Chapter 4 examined how to best improve a habitat model over the course of

an AUV campaign. When exploring the feature space, it was recommended that the feature space was sufficiently explored when all the bathymetric clusters were visited, or by planning over different budgets and identifying elbow points where there are diminishing returns. A similar method can be applied to active learning experiments, where deployments could be halted when the habitat model's accuracy plateaus or reaches some required accuracy. However, it would be beneficial to compare the expected information to the cost of acquisition. For example, it would cost $\$x$ to complete another survey, is the expected information gain worth it? Although this would be difficult to quantify, it would be interesting framing AUV campaigns in this manner.

5.2.5 Online imagery interpretation and replanning

In this thesis, the paths were planned offline, prior to the AUV deployment. This was necessary given the complexity of online classification of benthic habitats, particularly for areas where there is no existing data. This is also beneficial for AUV operations, where preplanning the paths can be crucial for validation and operation planning. However, if accurate habitat classifiers were built and the AUV had the capability to run online classification, there would be the opportunity for online image interpretation and replanning. This would open up a whole new avenue of research, where decisions would need to be made about how the robot should react to the imagery and when to trigger replanning or retraining.

List of References

Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. On the surprising behavior of distance metrics in high dimensional space. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1973:420 434, 2001. ISSN 16113349.

Allen Coral Atlas. Allen Coral Atlas: Imagery, maps and monitoring of the world's tropical coral reefs. 2020.

Akash Arora, P. Michael Furlong, Robert Fitch, Salah Sukkarieh, and Terrence Fong. Multi-modal active perception for information gathering in science missions. *Autonomous Robots*, 43(7):1827 1853, 2019. ISSN 15737527.

Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep Batch Active Learning by Diverse, Uncertain Gradient Lower Bounds. In *International Conference on Learning Representations*, 2020.

Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. Beyond RGB: Very high resolution urban remote sensing with multimodal deep networks. *ISPRS Journal of Photogrammetry and Remote Sensing*, 140:20 32, 2018. ISSN 09242716.

Benjamin Ayton, Brian Williams, and Richard Camilli. Measurement maximizing adaptive sampling with risk bounding functions. *33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*, pages 7511 7519, 2019. ISSN 2159-5399.

Elaine K. Baker and Peter T. Harris. *Habitat mapping and marine management*. Elsevier Inc., 2020. ISBN 9780128149607.

Asher Bender. *Autonomous Exploration of Large-Scale Natural Environments*. PhD thesis, University of Sydney, 2013.

Asher Bender, Stefan B. Williams, and Oscar Pizarro. Classification with probabilistic targets. *IEEE International Conference on Intelligent Robots and Systems*, pages 1780 1786, 2012a. ISSN 21530858.

Asher Bender, Stefan B. Williams, and Oscar Pizarro. Classification with probabilistic targets. *IEEE International Conference on Intelligent Robots and Systems*, pages 1780 1786, 2012b. ISSN 21530858.

Asher Bender, Stefan B. Williams, and Oscar Pizarro. Autonomous exploration of large-scale benthic environments. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 390 396, 2013a. ISSN 10504729.

Asher Bender, Stefan B. Williams, and Oscar Pizarro. Autonomous exploration of large-scale benthic environments. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 390 396, 2013b. ISSN 10504729.

M. S. Bewley, B. Douillard, N. Nourani-Vatani, A. Friedman, O. Pizarro, and S. B. Williams. Automated species detection: An experimental approach to kelp detection from sea-floor AUV images. *Australasian Conference on Robotics and Automation, ACRA*, 2012. ISSN 14482053.

Eli Bingham, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D. Goodman. Pyro: Deep Universal Probabilistic Programming. *Journal of Machine Learning Research*, 20(1):973 978, 2019.

David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*, 112 (518):859 877, 2017. ISSN 1537274X.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight Uncertainty in Neural Networks. 37, 2015.

Matteo Bresciani, Francesco Ruscio, Simone Tani, Giovanni Peralta, Andrea Timperi, Eric Guerrero-Font, Francisco Bonin-Font, Andrea Caiti, and Riccardo Costanzi. Path planning for underwater information gathering based on genetic algorithms and data stochastic models. *Journal of Marine Science and Engineering*, 9(11), 2021. ISSN 20771312.

Leo Brieman. Random Forests. *Machine Learning*, 45:5 32, 2001. ISSN 16113349.

Glenn Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1 8, 1950.

Craig J. Brown, Stephen J. Smith, Peter Lawton, and John T. Anderson. Benthic habitat mapping: A review of progress towards improved understanding of the spatial ecology of the seafloor using acoustic techniques. *Estuarine, Coastal and Shelf Science*, 92(3):502 520, 2011a. ISSN 02727714.

Craig J. Brown, Stephen J. Smith, Peter Lawton, and John T. Anderson. Benthic habitat mapping: A review of progress towards improved understanding of the spatial ecology of the seafloor using acoustic techniques. *Estuarine, Coastal and Shelf Science*, 92(3):502 520, 2011b. ISSN 02727714.

Craig J. Brown, Jonathan Beaudoin, Mike Brissette, and Vicki Gazzola. Multispectral multibeam echo sounder backscatter as a tool for improved seafloor characterization. *Geosciences (Switzerland)*, 9(3), 2019. ISSN 20763263.

Cameron B. Browne, Edward Powley, Daniel Whitehouse, Simon M. Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of Monte Carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1 43, 2012. ISSN 1943068X.

Jay Calvert, James Asa Strong, Matthew Service, Chris McGonigle, and Rory Quinn. An evaluation of supervised and unsupervised classification techniques for marine benthic habitat mapping using multibeam echosounder data. *ICES Journal of Marine Science*, 72(5):1498 1513, 6 2015. ISSN 1095-9289.

Alberto Candela, Kevin Edelson, Michelle M. Gierach, David R. Thompson, Gail Woodward, and David Wettergreen. Using Remote Sensing and in situ Measurements for Efficient Mapping and Optimal Sampling of Coral Reefs. *Frontiers in Marine Science*, 8(September):1 17, 2021. ISSN 22967745.

Marc Carreras, Juan David Hernandez, Eduard Vidal, Narcis Palomeras, David Ribas, and Pere Ridao. Sparus II AUV - A Hovering Vehicle for Seabed Inspection. *IEEE Journal of Oceanic Engineering*, 43(2):344 355, 2018. ISSN 03649059.

Weizhe Chen and Lantao Liu. Pareto Monte Carlo Tree Search for Multi-Objective Informative Planning. In *Robotics: Science and Systems 2019*, Freiburg im Breisgau, 2019.

Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-View 3D Object Detection Network for Autonomous Driving. 2016.

Corrina Cortes and Vladimir Vapnil. Support Vector Networks. *Machine Learning*, 20:273 297, 1995. ISSN 08859000.

Jnaneshwar Das, Julio Harvey, Frederic Py, Harshvardhan Vathsangam, Rishi Graham, Kanna Rajan, and Gaurav S. Sukhatme. Hierarchical probabilistic regression for AUV-based adaptive sampling of marine phenomena. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 5571 5578, 2013. ISSN 10504729.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. ImageNet: A large-scale hierarchical image database. *Cvpr*, pages 248 255, 2009. ISSN 1063-6919.

Heidi M Dierssen and Albert Theberge. Bathymetry : History of Seafloor Mapping. In *Encyclopedia of Natural Resources*, number September. Taylor & Francis, 2014.

Martin Engelcke, Dushyant Rao, Dominic Zeng Wang, Chi Hay Tong, and Ingmar Posner. Vote3Deep: Fast Object Detection in 3D Point Clouds Using Efficient Convolutional Neural Networks. In *IEEE International Conference on Robotics and Automation*, 2017.

State Government of NSW Environment and Department of Planning Industry and. NSW Marine LiDAR Topo-Bathy 2018 Geotif. Technical report, NSW Government, 2018. URL `https://data.nsw.gov.au/data/dataset/marine-lidar-topo-bathy-2018`.

Scott D. Foster, Geoffrey R. Hosack, Nicole A. Hill, Neville S. Barrett, and Vanessa L. Lucieer. Choosing between strategies for designing surveys: Autonomous underwater vehicles. *Methods in Ecology and Evolution*, 5(3): 287 297, 2014. ISSN 2041210X.

Scott D. Foster, Geoffrey R. Hosack, Jacquomo Monk, Emma Lawrence, Neville S. Barrett, Alan Williams, and Rachel Przeslawski. Spatially balanced designs for transect-based surveys. *Methods in Ecology and Evolution*, 11(1):95 105, 2020. ISSN 2041210X.

Ariell Friedman, Oscar Pizarro, Stefan B. Williams, and Matthew Johnson-Roberson. Multi-Scale Measures of Rugosity, Slope and Aspect from Benthic Stereo Image Reconstructions. *PLoS ONE*, 7(12), 2012. ISSN 19326203.

Atsushi Fujii, Takenobu Tokunaga, Kentaro Inui, and Hozumi Tanaka. Selective Sampling for Example-based Word Sense Disambiguation. *Computational Linguistics*, 24(4):573 597, 1998. ISSN 08912017.

Yarin Gal. Uncertainty in Deep Learning. *PhD Thesis*, (October):174, 2016.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *International Conference on Machine Learning*, volume 48, pages 1050 1059, 2016.

Jacob R. Gardner, Geoff Pleiss, David Bindel, Kilian Q. Weinberger, and Andrew Gordon Wilson. Gpytorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration. *Advances in Neural Information Processing Systems*, 2018-Decem(NeurIPS):7576 7586, 2018. ISSN 10495258.

Yonatan Geifman and Ran El-Yaniv. Deep Active Learning Over the Long Tail. *arXiv*, (m):1 10, 2017. ISSN 23318422.

Wenbo Gong, Sebastian Tschiatschek, Richard Turner, Sebastian Nowozin, José Miguel Hernández-Lobato, and Cheng Zhang. Icebreaker: Element-wise Active Information Acquisition with Bayesian Deep Latent Gaussian Model. In *Advances in Neural Information Processing Systems 32*, pages 14820 14831. Curran Associates, Inc., 2019.

H. Gary Greene, Joseph J. Bizzarro, Victoria M. O'Connell, and Cleo K. Brylinsky. Construction of digital potential marine benthic habitat maps using a coded classification scheme and its application. *Special Paper - Geological Association of Canada*, (47):141 155, 2007. ISSN 00721042.

Xifeng Guo, Xinwang Liu, En Zhu, and Jianping Yin. Deep Clustering with Convolutional Autoencoders. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10635 LNCS:373 382, 2017. ISSN 16113349.

Rozaimi Che Hasan, Daniel Ierodiaconou, and Jacquomo Monk. Evaluation of four supervised learning methods for benthic habitat mapping using backscatter from multi-beam sonar. *Remote Sensing*, 4(11):3427 3443, 2012. ISSN 20724292.

W. K. Hastings. Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*, 57(1):97 109, 1970.

James Hensman, Alexander G. Matthews, and Zoubin Ghahramani. Scalable variational Gaussian process classification. *Journal of Machine Learning Research*, 38:351 360, 2015. ISSN 15337928.

Gregory Hitz, Enric Galceran, Marie Ève Garneau, François Pomerleau, and Roland Siegwart. Adaptive continuous-space informative path planning for online environmental monitoring. *Journal of Field Robotics*, 34(8):1427 1449, 2017. ISSN 15564967.

Geoffrey A. Hollinger and Gaurav S. Sukhatme. Sampling-based robotic information gathering algorithms. *International Journal of Robotics Research*, 33 (9):1271 1287, 2014. ISSN 17413176.

Berthold K.P. Horn. Hill Shading and the Reflectance Map. *Proceedings of the IEEE*, 69(1):14 47, 1981. ISSN 15582256.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian Active Learning for Classification and Preference Learning. pages 1 17, 2011.

Maani Ghaffari Jadidi, Jaime Valls Miro, and Gamini Dissanayake. Sampling-based incremental information gathering with applications to robotic exploration and environmental monitoring. *International Journal of Robotics Research*, 38(6): 658 685, 2019.

Michael V. Jakuba, Christopher R. German, Andrew D. Bowen, Louis L. Whitcomb, Kevin Hand, Andrew Branch, Steve Chien, and Christopher McFarland. Teleoperation and robotics under ice: Implications for planetary exploration. *IEEE Aerospace Conference Proceedings*, 2018-March:1 14, 2018. ISSN 1095323X.

Michael Jordan, Zoubin Ghahramania, Tommi Jaakkola, and Lawrence Saul. An Introduction to Variational Methods for Graphical Models. *Machine Learning*, 37: 183 233, 1999. ISSN 08856125.

Sertac Karaman, Matthew R. Walter, Alejandro Perez, Emilio Frazzoli, and Seth Teller. Anytime motion planning using the RRT. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 1478 1483, 2011. ISSN 10504729.

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, 1996. ISSN 1042296X.

Alex Kendall and Yarin Gal. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? In *Advances in Neural Information Processing Systems 30*, number 5574-5584, 2017.

Sinae Kim, Mahlet G. Tadesse, and Marina Vannucci. Variable selection in clustering via Dirichlet process mixture models. *Biometrika*, 93(4):877 893, 2006. ISSN 00063444.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. pages 1 15, 2014. ISSN 09252312.

Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. *Foundations and Trends in Machine Learning*, 12(4):307 392, 2019. ISSN 19358245.

Diederik P. Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In C. Cortes Garnett, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R., editors, *Advances in Neural Information Processing Systems 28*, pages 2575 2583. Curran Associates, Inc., 2015.

Andreas Kirsch, Joost Van Amersfoort, and Yarin Gal. BatchBALD: Efficient and diverse batch acquisition for deep bayesian active learning. *arXiv*, (NeurIPS), 2019. ISSN 23318422.

Levente Kocsis and Csaba Szepesvári. Bandit Based Monte-Carlo Planning. In *Machine Learning: ECML 2006*, pages 282 293, 2006. ISBN 978-3-540-45375-8.

Suhit Kodgule, Alberto Candela, and David Wettergreen. Non-myopic Planetary Exploration Combining in Situ and Remote Measurements. *IEEE International Conference on Intelligent Robots and Systems*, pages 536 543, 2019. ISSN 21530866.

Vladimir Kostylev, Brian Todd, Gordon Fader, R. Courtney, Gordon Cameron, and Richard Pickrill. Benthic habitat mapping on the Scotian Shelf based on multibeam bathymetry, surficial geology and sea floor photographs. *Marine Ecology Progress Series*, 219:121 137, 2001.

Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9:235 284, 2008. ISSN 15324435.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Advances In Neural Information Processing Systems*, pages 1 9, 2012. ISSN 10495258.

Y Kwon, JH Won, BJ Kim, and MC Paik. Uncertainty quantification using Bayesian neural networks in classification: Application to ischemic stroke lesion segmentation. In *Medical Imaging and Deep Learning*, Amsterdam, The Netherlands, 2018.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, 2017-Decem(Nips):6403 6414, 2017. ISSN 10495258.

Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521 (7553):436 444, 2015. ISSN 14764687.

Rongxing Li, Steven W. Squyres, Raymond E. Arvidson, Brent A. Archinal, Jim Bell, Yang Cheng, Larry Crumpler, David J. Des Marais, Kaichang Di, Todd A. Ely, Matt Golombek, Eric Graat, John Grant, Joe Guinn, Andrew Johnson, Ron Greeley, Randolph L. Kirk, Mark Maimone, Larry H. Matthies, Mike Malin, Tim Parker, Mike Sims, Larry A. Soderblom, Shane Thompson, Jue Wang, Patrick Whelley, and Fengliang Xu. Initial results of rover localization and topographic mapping for the 2003 mars exploration rover mission. *Photogrammetric Engineering and Remote Sensing*, 71(10):1129 1142, 2005. ISSN 00991112.

V. Lucieer and A. Lucieer. Fuzzy clustering for seafloor classification. *Marine Geology*, 264(3-4):230 241, 2009. ISSN 00253227.

Mitchell Lyons, Chris Roelfsema, Emma Kennedy, Eva Kovacs, Rodney Borrego-Acevedo, Kathryn Markey, Meredith Roe, Doddy Yuwono, Daniel Harris, Stuart Phinn, Gregory P. Asner, Jiwei Li, David Knapp, Nicholas Fabina, Kirk Larsen, Dimosthenis Traganos, and Nicholas Murray. Mapping the world's coral reefs using a global multiscale earth observation framework. *Remote Sensing in Ecology and Conservation*, 6(4):557 568, 2020. ISSN 20563485.

Prasanta Chandra Mahalanobis. On the generilised distance in statistics. *Proceedings of the National Institute of Sciences of India*, pages 49 55, 1936.

Ivor Marsh and Colin Brown. Neural network classification of multibeam backscatter and bathymetry data from Stanton Bank (Area IV). *Applied Acoustics*, 70(10):1269 1276, 2009. ISSN 0003682X.

Larry Mayer, Martin Jakobsson, Graham Allen, Boris Dorschel, Robin Falconer, Vicki Ferrini, Geoffroy Lamarche, Helen Snaith, and Pauline Weatherall. The Nippon Foundation-GEBCO seabed 2030 project: The quest to see the world's oceans completely mapped by 2030. *Geosciences (Switzerland)*, 8(2), 2018. ISSN 20763263.

Ewan Morgan. Can Scientists Map the Entire Seafloor by. 2021.

Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using Bayesian Binning. *Proceedings of the National Conference on Artificial Intelligence*, 4:2901 2907, 2015. ISSN 2159-5399.

Radford M. Neal. *Bayesian Learning for Neural Networks*. PhD thesis, University of Toronto, 1996.

Hieu T. Nguyen and Arnold Smeulders. Active learning using pre-clustering. *Proceedings, Twenty-First International Conference on Machine Learning, ICML 2004*, pages 623 630, 2004.

Sabine Ohlendorf, Andreas Müller, Thomas Heege, Sergio Cerdeira-Estrada, and Halina T. Kobryn. Bathymetry mapping and sea floor classification using multispectral satellite data and standardized physics-based data processing. *Remote Sensing of the Ocean, Sea Ice, Coastal Waters, and Large Water Regions 2011*, 8175(October 2011):817503, 2011.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in Neural Information Processing Systems*, 32(NeurIPS), 2019. ISSN 10495258.

Edouard Oyallon. Building a regular decision boundary with deep networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua:1886 1894, 2017. ISSN 1703.01775.

Timothy Patten, Wolfram Martens, and Robert Fitch. Monte Carlo planning for active object classification. *Autonomous Robots*, 42(2):391 421, 2018. ISSN 15737527.

Liam Paull, Sajad Saeedi, Mae Seto, and Howard Li. AUV Navigation and Localization: A Review. *IEEE Journal of Oceanic Engineering*, 39(1):131 149, 2014. ISSN 03649059.

Adam Pazke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in prose. In *NIPS-W*, 2017.

Laurent Perron and Vincent Furnon. OR-Tools. 2019.

Robert Pinsler, Jonathan Gordon, Eric Nalisnick, and José Miguel Hernández-Lobato. Bayesian batch active learning as sparse subset approximation. *Advances in Neural Information Processing Systems*, 32 (NeurIPS), 2019. ISSN 10495258.

Marija Popovic, Teresa Vidal-Calleja, Gregory Hitz, Inkyu Sa, Roland Siegwart, and Juan Nieto. Multiresolution mapping and informative path planning for UAV-based terrain monitoring. *IEEE International Conference on Intelligent Robots and Systems*, 2017-Septe:1382 1388, 2017. ISSN 21530866.

Redmond Ramin Shamshiri, Cornelia Weltzien, Ibrahim A. Hameed, Ian J. Yule, Tony E. Grift, Siva K. Balasundram, Lenka Pitonakova, Desa Ahmad, and Girish Chowdhary. Research and development in agricultural robotics: A perspective of digital farming. *International Journal of Agricultural and Biological Engineering*, 11(4):1 11, 2018. ISSN 1934-6344.

Dushyant Rao, Mark De Deuge, Navid Nourani-Vatani, Stefan B. Williams, and Oscar Pizarro. Multimodal learning and inference from visual and remotely sensed data. *International Journal of Robotics Research*, 36(1):24 43, 2017. ISSN 17413176.

Carl Edward Rasmussen and Christopher Williams. *Gaussian Processes for Machine Learning*, volume 1. MIT Press, Massachusetts, 2006. ISBN 026218253X.

Alex Rattray, Daniel Ierodiaconou, Laurie Laurenson, Shoaib Burq, and Marcus Reston. Hydro-acoustic remote sensing of benthic biological communities on the shallow South East Australian continental shelf. *Estuarine, Coastal and Shelf Science*, 84(2):237 245, 2009. ISSN 02727714.

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po Yao Huang, Zhihui Li, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *arXiv*, 2020. ISSN 23318422.

Douglas Reynolds. Gaussian Mixture Models. *Encyclopedia of biometrics*, 741(2): 1 5, 2009.

Daniel Roper, Catherine A. Harris, Georgios Salavasidis, Miles Pebody, Robert Templeton, Thomas Prampart, Matthew Kingsland, Richard Morrison, Maaten Furlong, Alexander B. Phillips, and Stephen McPhail. Autosub Long Range 6000: A Multiple-Month Endurance AUV for Deep-Ocean Monitoring and Survey. *IEEE Journal of Oceanic Engineering*, 46(4):1179 1191, 2021. ISSN 15581691.

Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(1), 1958.

Julius Rückin, Liren Jin, and Marija Popović. Adaptive Informative Path Planning Using Deep Reinforcement Learning for UAV-based Active Sensing. 2021.

David Rumelhart, Geoffrey Hinton, and Ronald Williams. Learning internal representations by error propagation. In *Parallel Distributed Processing*. MIT Press, Cambridge, MA, 1 edition, 1986.

Stuart Russel and Peter Norvig. *Artificial Intelligence - A Modern Approach*. Pearson Education Limited, Harlow, 3rd edition, 2014.

Gregory Schultz, Joe Keranen, Chet Bassani, Matthew Cook, and Timothy Crandle. Littoral applications of advanced 3D marine electromagnetics from remotely and autonomously operated platforms: Infrastructure and hazard detection and characterization. *OCEANS 2017 - Aberdeen*, 2017-Octob:1 7, 2017.

Ramprasaath Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Praikh, and Dhruv Batra. Visual Explanations from Deep Networks via Gradient-based Localization. In *Conmputer Vision (ICCV)*, Venice, Italy, 2017. IEEE.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, pages 1 13, 2018.

Burr Settles. Active Learning Literature Survey. Technical Report Computer Sciences Technical Report 1648, University of Wisconsin Madison, 2009.

Mark Sheinin and Yoav Schechner. The Next Best Underwater View. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3764 3773, Las Vegas, NV, 2016. IEEE.

Jackson Shields, Oscar Pizarro, and Stefan B. Williams. Towards Adaptive Benthic Habitat Mapping. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 9263 9270, 2020. ISSN 10504729.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484 489, 2016. ISSN 14764687.

Amarjeet Singh, Andreas Krause, Carlos Guestrin, and William J. Kaiser. Efficient Informative Sensing using Multiple Robots. *Journal of Artificial Intelligence Research*, 34:707 755, 2009. ISSN 10769757.

Hanumant Singh, Ali Can, Ryan Eustice, Steve Lerner, Neil Mcphee, Oscar Pjzarro, and Chris Roman. Seabed auv offers new platform for high-resolution imaging. *Eos*, 85(31), 2004. ISSN 00963941.

Samrath Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-Octob:5971 5980, 2019. ISSN 15505499.

Korsuk Sirinukunwattana, Shan E Ahmed Raza, Yee-Wah Tsang, David Snead, Ian Cree, and Nasur Rajpoot. Locality Sensitive Deep Learning for Detection and Classification of Nuclei in Routine Colon Cancer Histology Images. *IEEE Transactions on Medical Imaging*, (February):1 12, 2016.

M Spinoccia. Bathymetry Grids Of South East Tasmania Shelf. 2011.

Daniel Matthew Steinberg. An Unsupervised Approach to Modelling Visual Data. (December), 2012.

David L. Valentine, Christopher M. Reddy, Christopher Farwell, Tessa M. Hill, Oscar Pizarro, Dana R. Yoerger, Richard Camilli, Robert K. Nelson, Emily E. Peacock, Sarah C. Bagby, Brian A. Clarke, Christopher N. Roman, and Morgan Soloway. Asphalt volcanoes as a potential source of methane to late Pleistocene coastal waters. *Nature Geoscience*, 3(5):345 348, 2010. ISSN 17520894.

Laurens Van Der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(1):2579 2605, 2008.

Alberto Viseras and Ricardo Garcia. DeepIG: Multi-robot information gathering with deep reinforcement learning. *IEEE Robotics and Automation Letters*, 4(3): 3059 3066, 2019. ISSN 23773766.

David Williams. Underwater Target Classification in Synthetic Aperture Sonar Imagery Using Deep Convolutional Neural Networks. *International Conference on Pattern Recognition (ICPR)*, pages 2497 2502, 2016.

Stefan B. Williams, Oscar Pizarro, Ian Mahon, and Matthew Johnson-Roberson. Simultaneous Localisation and Mapping and Dense Stereoscopic Seafloor Reconstruction Using an AUV. *Springer Tracts in Advanced Robotics*, 54: 407 416, 2009. ISSN 16107438.

Stefan B. Williams, Oscar R. Pizarro, Michael V. Jakuba, Craig R. Johnson, Neville S. Barrett, Russell C. Babcock, Gary A. Kendrick, Peter D. Steinberg, Andrew J. Heyward, Peter J. Doherty, Ian Mahon, Matthew Johnson-Roberson, Daniel Steinberg, and Ariell Friedman. Monitoring of benthic reference sites: Using an autonomous underwater vehicle. *IEEE Robotics and Automation Magazine*, 19(1):73 84, 2012. ISSN 10709932.

Margaret F.J. Wilson, Brian O'Connell, Colin Brown, Janine C. Guinan, and Anthony J. Grehan. Multiscale terrain analysis of multibeam bathymetry data for habitat mapping on the continental slope. *Marine Geodesy*, 30(1-2):3 35, 2007. ISSN 01490419.

Troy Wilson and Stefan B. Williams. Adaptive path planning for depth-constrained bathymetric mapping with an autonomous surface vessel. *Journal of Field Robotics*, 35(3):345 358, 2018. ISSN 15564967.

Lloyd Windrim, Arman Melkumyan, Richard J. Murphy, Anna Chlingaryan, and Rishi Ramakrishnan. Pretraining for Hyperspectral Convolutional Neural Network Classification. *IEEE Transactions on Geoscience and Remote Sensing*, 56(5):2798 2810, 2018. ISSN 01962892.

Anne Cathrin Wölfl, Helen Snaith, Sam Amirebrahimi, Colin W. Devey, Boris Dorschel, Vicki Ferrini, Veerle A.I. Huvenne, Martin Jakobsson, Jennifer Jencks, Gordon Johnston, Geoffroy Lamarche, Larry Mayer, David Millar, Terje Haga Pedersen, Kim Picard, Anja Reitz, Thierry Schmitt, Martin Visbeck, Pauline Weatherall, and Rochelle Wigley. Seafloor mapping - The challenge of a truly global ocean bathymetry. *Frontiers in Marine Science*, 6(JUN):1 16, 2019. ISSN 22967745.

Dawn Wright and William D. Heyman. Introduction to the Special Issue: Marine and Coastal GIS for Geomorphology, Habitat Mapping, and Marine Reserves. *Marine Geodesy*, 31(4):1 8, 2008. ISSN 01490419.

Russell B. Wynn, Veerle A.I. Huvenne, Timothy P. Le Bas, Bramley J. Murton, Douglas P. Connelly, Brian J. Bett, Henry A. Ruhl, Kirsty J. Morris, Jeffrey Peakall, Daniel R. Parsons, Esther J. Sumner, Stephen E. Darby, Robert M. Dorrell, and James E. Hunt. Autonomous Underwater Vehicles (AUVs): Their

past, present and future contributions to the advancement of marine geoscience. *Marine Geology*, 352:451 468, 2014. ISSN 00253227.

Takaki Yamada, Adam Prügel-Bennett1, Stefan Williams, Oscar Pizarro, and Blair Thornton. GeoCLR: Georeference Contrastive Learning for Efficient Seafloor Image Interpretation. *Field Robotics*, 2(1):1134 1155, 3 2022. ISSN 27713989.

Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang. Linear spatial pyramid matching using sparse coding for image classification. In *2009 IEEE Conference on computer vision and pattern recognition*, pages 1 8. IEEE, 2009. ISBN 0033-8419 (Print)\r0033-8419 (Linking).

Changchang Yin, Buyue Qian, Shilei Cao, Xiaoyu Li, Jishang Wei, Qinghua Zheng, and Ian Davidson. Deep similarity-based batch mode active learning with exploration-exploitation. *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2017-Novem:575 584, 2017. ISSN 15504786.

Donggeun Yoo and In So Kweon. Learning Loss for Active Learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 93 102, 2019.