UNIVERSITY OF CALIFORNIA,
IRVINE

Variational Methods for Optimal Experimental Design

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Computer Science

by

Noble William Kennamer

Dissertation Committee:
Professor Alexander Ihler, Chair
Chancellor's Professor Padhraic Smyth
Professor David Kirkby

2022

DEDICATION

To my wife Jacqueline DeMarco,
for your love and support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I am deeply grateful to all of the amazing collaborators and mentors I had the opportunity to work with during the course of my Ph.D. As an undergraduate student at UCI I began working with professor Padhraic Smyth and I still reflect on many of the early lessons I learned from Padhraic then, in particular his advice to not get so caught up with computational experiments that I neglect forming a deep understanding of the mathematical roots of the problems I work on. I'm also grateful to Padhraic for introducing me to my graduate advisor, professor Alex Ihler. I could not have asked for a better mentor than Alex, who never stopped challenging me to be the best researcher I could be while remaining patient and giving me the freedom and support to pursue the problems that I found most interesting. I am deeply grateful for his willingness to answer any question I came to him with (even willing to answer the same question over and over again until I got it); whether it was directly related to our work or on a more tangential subject, he always encouraged my curiosity. If I was fortunate to have Alex as my mentor, than I was doubly fortunate to have a second great mentor in professor David Kirkby. I worked with David for the entirety of my Ph.D. and am deeply grateful for his warmth and generosity. He always made me feel welcome and valued as a computer scientist in the world of cosmology, and he is an incredible collaborator, not only always willing to share his knowledge but also to learn from and incorporate knowledge from other fields. Alex, David and Padhraic embody the best qualities of a scientist: they posses an incredible work ethic and have become deeply knowledgeable in their areas of expertise, while remaining curious and open minded, treating those around them with kindness. Since knowing them I've tried to live up to the example they set; hopefully one day I'll succeed.

Early in my graduate studies I participated in the Cosmostatistics Initiative's (COIN) resident program, in which I got to spend a week living and working with an incredible group of scientists and turned into a long-term collaboration. Joining this community of scientists is easily one of the best decisions I made during my graduate studies. The COIN members

form a diverse group of scientists with expertise in many different fields and come from all over the globe. I value the professional and personal relationships I've formed with the COIN members and am proud of the work we produced together. The COIN leaders, in particular Dr. Emille Ishida, Dr. Alberto Krone-Martins and Dr. Rafael de Souza, created a community that empowers all of its members to be heard and in which their contributions are valued. It's a space to share knowledge and exchange ideas, even the crazy ones (perhaps especially the crazy ones), and to collaborate with researchers who truly care both about the work and the people they work with. During some of the hardest times of my Ph.D. I found support in the COIN community and would like to personally thank Emille, Alberto and Rafael for everything they've done for me. I look forward to all our future collaborations.

I would also like to thank all of my fellow students that I got to know and work with over the course of my Ph.D. I greatly valued my relationships with my fellow lab mates from Alex's, professor Rina Dechter's and David's groups: Qi Lou, Nick Gallo, Tiancheng Xu, Filjor Broka, Bobak Pezeshki, Litian Liang, Yaosheng Xu, Bela Abolfathi, Dylan Green, Abby Bault and Javier Sanchez. I also had to opportunity to receive two Fellowships: the National Science Foundation fellowship for Machine Learning Applied to the Physical Sciences (MAPS) and the Hasso Plattner Institute (HPI) fellowship, as well as an ARCS Foundation scholar award, which both provided financial support to enable me to complete my thesis and also gave me access to a community of other UCI students from many different departments, and researchers and students at the Hasso Plattner Institute. My Ph.D. was deeply enriched by the relationships I formed in MAPS, ARCS and HPI, in which I had the opportunity to learn about all of the amazing research being done across UCI and at HPI, and to get constructive feedback on my own work. I would like to thank everyone involved with MAPS, ARCS and HPI. I would also like to thank Steven Walton, a Ph.D. student at the University of Oregon and one of my closest friends. His keen insight and creative ideas made my work better and made me think differently about artificial intelligence, machine

learning, statistics, and technology and science in general. Its difficult to express how much I value all of our discussions and collaborations.

Finally I would like to thank all of my close family and friends who supported me during my Ph.D., especially my mother Gail Kennamer, my mother- and father- in-law Lance and Caroline DeMarco, Dr. Michele Rousseau, who first introduced me to computer science and encouraged me to pursue a Ph.D., and most of all my wife Jacqueline DeMarco. Without her love and support there is no way I could have finished this thesis.

The text of 6.1-6.8 is based on material as it appears in Ishida et al. [2019], "Optimizing Spectroscopic follow-up strategies for supernova photometric classification with active learning" originally published in Monthly Notices of the Royal Astronomical Society Volume 483, Issue1, Pages 2-18, 2019. Used with permission from Oxford University Press (OUP). The co-authors listed in this publication are Emille Ishida, Robert Beck, Santiago Gonzalez-Gaitan, Rafael de Souza, Alberto Krone-Martins, Jim Barrett, Noble Kennamer, Ricardo Vilalta, Bruno Quint, Andre Vitorelli, Ashish Mahabal and Emmanuel Gangler.

The text of 6.9-6.10 is based on material as it appears in Kennamer et al. [2020], "Active learning with RESSPECT: Resource allocation for extragalactic astronomical transients" originally published in 2020 IEEE Symposium Series on Computer Science, pages 3115-3124, 2020. Used with permission from the Institute of Electrical and Electronic Engineers (IEEE). The co-athors listed in this publication are Noble Kennamer, Emille Ishida, Santiago Gonzalez-Gaitan, Rafael de Souza, Alexander Ihler, Kara Ponder, Ricardo Vilalta, Anais Moller, David Jones, Mi Dai, Alberto Krone-Martins, Bruno Quint, Sreevarsha Sreejith, Alex Malz and Lluis Galbany. The co-author Alexander Ihler listed in this publication directed and supervised research which forms the basis for the thesis.

The text of chapter 7 in this dissertation is based on the material in Kennamer et al. [2018], "ContextNet: Deep learning for star galaxy classification" originally published in Proceedings

of the 35th International Conference on Machine Learning, volume 80 of PMLR pages 2582-2590, 2018. Used with permission from Proceedings of Machine Learning Research (PMLR). The co-authors listed in the publication are Noble Kennamer, Alexander Ihler, David Kirkby and Francisco Javier Sanchez-Lopez. The co-author Alexander Ihler listed in this publication directed and supervised research which forms the basis for the thesis.

I would like to thank the following funding sources the National Science Foundation Machine Learning Applied to the Physical Sciences fellowship (grant number 1633631), the Hasso Plattner Institute fellowship, the ARCS Scholar award, NSF grant IIS1254071, DARPA under the World Modelers program (W911NF18C0015) and the US Deoartment of Energy award DE-SC0009920,

VITA

Noble William Kennamer

EDUCATION

Doctor of Philosophy in Computer Science **2022**
University of California, Irvine *Irvine, California*

Master of Science in Computer Science **2017**
University of California, Irvine *Irvine, California*

Bachelor of Science in Physics and Mathematics **2015**
University of California, Irvine -*Summa Cum Laude* *Irvine, California*

Research EXPERIENCE

Graduate Research Assistant **2015 – 2022**
University of California, Irvine *Irvine, California*

Conference Presentations

Active Learning with RESSPECT:Resource Allocation for Extra-galactic Astronomical Transients
December 2020 IEEE SSCI *virtual*

ContextNet: Deep Learning for Star Galaxy Classification
July 2018 ICML *Stockholm, Sweeden*

Awards

Hasso Plattner Institute Fellowship 2020-2022
ARCS Scholar Award 2020
NSF, Machine Learning Applied to the Physical Sciences Fellowship 2017-2020

PUBLICATIONS

N. Kennamer, A. Ihler, "Amortized Optimization for Variational Bayesian Optimal Experimental Design," *In submission to AI & Statistics*, 2023.

N. Kennamer, S. Walton, A. Ihler, "Design Amortization for Bayesian Optimal Experimental Design," *In submission to AAAI*, 2023. https://arxiv.org/abs/2210.03283.

N. Kennamer, E. E. O. Ishida, S. González-Gaitán, R. S. de Souza, A. Ihler, K. Ponder, R. Vilalta, A. Möller, D. O. Jones, M. Dai, A. Krone-Martins, B. Quint, S. Sreejith, A. I. Malz, L. Galbany, "Active learning with RESSPECT: Resource allocation for extragalactic astronomical transients" *IEEE Symposium Series on Computational Intelligence* , 2020. 10.1109/SSCI47803.2020.9308300.

N. Kennamer, D. Kirkby, A. Ihler, F. J. Sanchez-Lopez, "ContextNet: Deep Learning for Star Galaxy Classication," *Proceedings of the 35th International Conference on Machine Learning*, 2018. https://proceedings.mlr.press/v80/kennamer18a.html https://proceedings.mlr.press/v80/kennamer18a.html.

E. E. O. Ishida, R. Beck, S. González-Gaitán, R. S. de Souza, A. Krone-Martins, J. W. Barrett, **N. Kennamer**, R. Vilalta, J. M. Burgess, B. Quint, A. Z. Vitorelli, A. Z. Mahabal, E. Gangler, "Optimizing Spectroscopic Follow-up Strategies for Supernova Photometric Classification with Active Learning" *Monthly Notices of the Royal Astronomical Society* , 2018. 10.1093/mnras/sty3015.

DESI Collaboration, "The DESI Experiment Part I: Science,Targeting, and Survey Design" *arXiv* , 2016. 10.48550/ARXIV.1611.00036.

DESI Collaboration, "The DESI Experiment Part II: Instrument Design" *arXiv* , 2016. 10.48550/ARXIV.1611.00037.

ABSTRACT OF THE DISSERTATION

Variational Methods for Optimal Experimental Design

By

Noble William Kennamer

Doctor of Philosophy in Computer Science

University of California, Irvine, 2022

Professor Alexander Ihler, Chair

In this work we study variational methods for Bayesian optimal experimental design (BOED). Experimentation is a cornerstone of science and is central to any major engineering effort. Often experiments require the use of substantial resources, from expensive equipment to limited researcher time; in addition, experiments can be dangerous or may be required to be completed in a given period of time. For these reasons, we prefer to conduct our experiments as efficiently as possible, acquiring as much information as we can given the resources available to us. Optimal experimental design (OED) is a sub-field of statistics focused on developing methods for accomplishing this goal. The OED problem is formulated by defining a utility function over designs and optimizing this function over the set of all feasible designs. We focus on the *Expected Information Gain* (EIG), a widely used utility function with sound theoretical support. However, in practice the EIG is intractable to compute, and approximation strategies are required. We investigate the use of variational methods for this purpose and show substantial improvement over competing approximation techniques. A specific form of OED common in the field of machine learning (ML) is *active learning* (AL). In the active learning framework, we would like to obtain a labeled dataset in order to train a supervised model. However, for all the reasons stated, labeling data points can be costly and again we should make efficient use of our labeling resources. We present a novel application of active learning to optimize spectroscopic follow up for large scale

astronomical surveys. Finally, much of this work requires learning functions over sets which we know must satisfy certain properties (e.g., permutation invariance). We conclude the thesis by presenting a novel neural network architecture for predicting the astronomical class of individual objects in the same exposure using a neural architecture specifically designed to accommodate known inductive biases present in the data.

Chapter 1

Introduction

> To consult the statistician after an experiment is finished is often merely to ask him to conduct a post mortem examination. He can perhaps say what the experiment died of.
>
> Sir Ronald Fisher

This thesis primarily covers two broad themes in machine learning and statistics: optimal experimental design (OED) and learning functions over set-based data. Both topics play a fundamental role in three applied projects that motivate and inform the work, as illustrated in Figure 1.1, however OED is covered in much more in depth as it is the more central topic of this thesis.

1.1 Optimal Experimental Design

Experimenters are consistently faced with the challenge of conducting their experiments in resource- and time- limited settings. This can be for a number of reasons, including the limited capacity of expensive scientific devices or because the phenomena of interest can

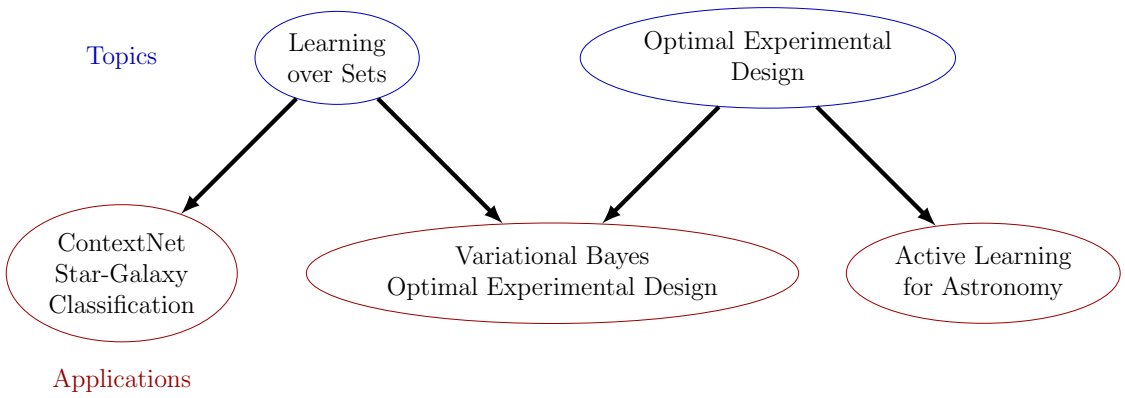

Figure 1.1: Research Overview

only be observed for a short period of time. Because of these limitations and constraints, experimenters must design their experiments to be maximally informative. For controlled experiments this can mean determining the best possible settings (e.g., temperature, chemical concentrations, time, etc.) to apply to each experimental unit, in order to produce observations that are more informative than any other settings we could have chosen. For observational studies, where we lack the ability to control the conditions of the experimental units, researchers still must to carefully pick which set of units should be observed and evaluated so as to gain as much information as possible from our observations.

Optimal experimental design is a sub-field of statistics that aims to address these challenges. The aim of OED is to produce methods and algorithms that can be used by experimenters to design their experiments to be as informative as possible. For this reason, advances in OED can yield advances across many areas of science and engineering generally, thanks to the central role of experimentation in science. In fact, the history of design of experiment (DOE) illustrates the impacts and success these concepts have already had, contributing to major advancements in science and engineering and thus across human society.

1.1.1 A Brief History of Design of Experiment

Experimental design was first pioneered by Danish statistician Kirstine Smith, who invented G-optimal designs for polynomial regression models in her dissertation published in 1918 [SMITH, 1918]. Subsequent development of the field of design of experiments is typically viewed as occurring in four different, distinct eras, starting with the work of Ronald Fisher in the 1920s and 1930s at Rothamstead experimental station, a research farm in Hertfordshire, England that remains in operation to this day. Many of the core principles of DOE came out of this early work, such as the importance of *randomization* to blunt the effect of unobserved confounding factors; the importance of *replication* to reduce statistical error; *blocking*, or the inclusion of factors in our experiment that are not of primary interest but contribute observational variation, so that their inclusion in design and analysis can be used to reduce error variance; and varying more than one factor at a time (*factorial designs*), which can greatly enhance our statistical efficiency and reduce the total number of observations required in our experiment. The vast majority of this work was developed for agricultural studies, but history has shown the ideas to be general enough to inform all areas of science and engineering.

The second major era (1950-1970) was pioneered by George E. P. Box and his colleagues, and was primarily directed towards applications in chemical manufacturing. In this work the goal was to control some industrial process so as to maximize the yield of the substance being produced. This required first determining what factors were actually influential in the overall yield of the process (often called *screening*) and then determining the optimal settings of these factors. Given the great expense of running these industrial processes and the many factors that could potentially contribute to the yield of the process, Box and his colleagues were inspired to invent major advancements in design of experiment. These include *response surface* methodology, where one formulates the output of a process as a function (often a quadratic function) and then uses sequential experimentation to screen for the relevant

factors and determine their optimal settings. The many potential factors and the quadratic effects in the models describing the process motivated the use of *fractional factorial designs*, which can greatly reduce the required number of experimental runs.

The third era of DOE (1970-1990) began in the 1950s, when statistical consultant W. Edwards Deming traveled to Japan to assist in its reconstruction following World War II. In the 1950s and 1960s, Japanese manufacturing was less advanced than other parts of the world, and many of their products were considered of lesser quality. Much of Deming's work on the use of statistical tools and design of experiments in quality control was embraced by Japanese manufacturers, especially their automobile industry, and led to massive improvements in the quality and consistency of the goods produced in Japan, turning the country into one of the centers of global innovation. Japanese engineer and statistician, Genichi Taguchi, was another major contributor during this time. His application of *orthogonal array designs* [Taguchi, 1987], which are similar to fractional factorial designs, contributed to much of this progress. The work done in Japan during this era was ultimately so successful that many of the approaches were ultimately adopted by companies across the world.

The 1990s is considered the start of the "modern era", and from this time design of experiment has proliferated through much of science and engineering, with applications stretching from culture experiments in microbiology [Montgomery, 2017, Goos and Jones, 2011] to wind tunnel experiments for NASA [DeLoach, 1997] and everything in between. The advent of powerful and relatively inexpensive computation has profoundly changed the way we are able to apply design of experiments today. In the past, researchers were often limited to choosing from a set of predetermined designs drawn from a table in the back of a book, while forcing them to try to make their experiment fit into one of these "standard cases". But experiments are diverse, with all sorts of differing goals and constraints, and often will not fit nicely into one of these precomputed cases. Now, however, experimenters are not required to force a square peg into a round hole – instead, they can use techniques from optimal experimental

design along with readily available computation to find bespoke designs that are tailored to their needs and unique circumstances. This requires specifying a *utility function*, which is used to compare the quality of different designs, and a *feasible set* of designs, which define all designs that could be executed in the lab or factory. Then, the researcher solves a mathematical optimization problem to find the best (or a very good) design. However, this is easier said then done – the desired utility function is often intractable to evaluate, and the subsequent optimization problem is then even harder. This thesis provides progress on both grounds, as described in the next section.

Our preceding discussion is meant to provide only a brief history of the field of design of experiment, and to give motivation for one of the main subjects of this thesis. It should not be considered comprehensive – for example, omitting notable contributions on clinical trials starting in the 1950s and experiments on artillery systems during World war II. For a more in-depth history of DOE, see [Montgomery, 2017].

1.1.2 Bayesian Optimal Experimental Design

The preceding section focused on approaches from classical statistics, which have been and remain widely used. However, the contributions of this thesis are primarily built on the Bayesian approach to statistics, which is being increasingly adopted of late. While both statistical philosophies have their merits, the Bayesian approach appears particularly advantageous to experimental design. Bayesian statistics focuses on quantifying uncertainty by interpreting probability in terms of subjective degrees of belief. Central to this approach are two mathematical objects: the prior and posterior distributions. The prior distribution encodes our beliefs about the world prior to conducting our experiment (or receiving new information), while the posterior distribution encapsulates our beliefs *after* receiving new information. In a sense, this dichotomy between the two different objects is superficial, since our posterior distributions of today then become our prior distributions for tomorrow. This

viewpoint of an ongoing refinement of belief provides an elegant framework for sequential experimentation, which naturally enables us to design experiments that build on our past knowledge for greater efficiency. However, both for sociological and computational reasons, Bayesian approaches were not widely adopted during the early development of DOE, and are still less widely used than classical methods today. This does, however, seem to be changing. In this thesis we make contributions for Bayesian approaches to optimal experimental design that are more computationally efficient and therefore more accessible to experimenters. This thesis does not attempt to comprehensively cover the differences between classical and Bayesian approaches; for further discussion on this subject see, e.g., Bernardo and Smith [2009], DIACONIS and SKYRMS [2018].

In Chapter 2 we give an illustrative example of OED used in a setting from microbiology. This example shows how OED can be formulated within classical statistics as well as Bayesian statistics. In Chapter 3 we focus on the Bayesian approach, defining a widely used and theoretically well supported utility function for comparing different possible designs called the *expected information gain* (EIG). This leads to a clear framework within which Bayesian optimal experimental designs can be conducted. While conceptually straightforward, the expected information gain is notoriously difficult to compute, leading to significant practical challenges. We discuss several approximation strategies, with a strong focus on variational methods, which we consider the most promising direction. In Chapter 4 we propose a novel deep learning architecture for approximating the expected information gain. Our method allows for amortization over designs, making it possible to train a single variational model that can be applied to approximate the expected information gain for designs across the feasible space. In addition we propose an inexpensive method for training our variational model. Moreover, our model is differentiable with respect to the design variables, which can facilitate the use of first order techniques for efficiently optimizing the design. This methodology represents the first major application of the thesis, denoted "Variational Bayesian Optimal

Experimental Design" in Figure 1.1.

1.2 Active Learning for Astronomy

Active learning is a specific application of optimal experimental design applied to supervised machine learning and draws from both statistics and machine learning to derive its techniques. In supervised machine learning, our goal is to learn a function that can be used to make predictions. This could be a discrete classification problem, like deciding if an image depicts a cancerous or healthy cell, a regression problem like attempting to predict the amount of rain from infrared satellite images, or structured prediction problems like predicting the connections between objects in a network. In any supervised learning problem, we require a training dataset: a set of pairs (x_i, y_i), where x_i denote measured features of the ith unit, and y_i is its label. We then define a parameterized function, $f_\phi(x_i)$, where ϕ are the trainable parameters, and a loss function, $L(f_\phi(x_i), y_i)$. We use the training data to learn the parameter values ϕ that minimize our loss function. For example, in regression the loss function could be the mean squared error between the true values and our predictions, $\sum_i^N (y_i - f_\phi(x_i))^2$, where N is the total number of objects in the training dataset. However, the goal of supervised learning is not simply to minimize the loss function over the training data, but rather to provide accurate predictions on data *not* seen during the training process, often called the test data. This concept is referred to as *generalization*. It is also common good practice to use an intermediate dataset called the validation set, which is used not to train the parameters of the model, but for model selection (deciding on the best form for the function f). Only after selecting a final model do we measure our performance on the test set.

As we see, the training set plays a central role in any supervised learning problem, where the amount and representativeness of the data in the training set directly impact our ability

to generalize well. Ideally, we would like a large training dataset with complete coverage of the types of features and labels we expect to see when applying our model in practice. Unfortunately, obtaining such a dataset can be both challenging and costly. Often, however, we may have plenty of unlabeled data – just the x_i's – and labeling each one (finding the desired value of the target y_i) represents the main bottleneck. For example, we might have a large repository of medical images, but the labeling process requires hiring medical experts to go through them one by one. These experts can be expensive to hire or have limited availability, so we should choose a set of data to label that will be most helpful in training a good model. Active learning frames this problem concretely: given an unlabeled set of data $\mathcal{U}$ and resource constraints, find a subset, $\mathcal{D}_{\text{tr}}$, to label that will lead to the best expected performance of our supervised model. (Versions of active learning can also be formulated for stream-based settings or data-synthesis settings.)

Chapter 6 covers a novel application of active learning to optimizing spectroscopic follow-up strategies for supernova photometric classification. We focus on a significant upcoming challenge posed by the Vera Rubin Observatory's Legacy Survey of Space and Time (LSST). LSST is an upcoming astronomical survey that will operate for ten years and improve our understanding of wide ranging topics in cosmology and astrophysics. The scale of the survey is enormous, with the expectation that it will collect 15 terabytes of data per night, culminating in a final dataset of 200 petabytes.

For our application the cadence of the survey is critically important, imaging the entire southern sky every three nights. This cadence will enable the detection of thousands to tens of thousands of transients per night. Transients comprise a broad category consisting of any astronomical object that changes with time over intervals of minutes to weeks. This change could be due to movement, as in the case of asteroids, or due to changes in the object itself, such as variable stars or supernovae (an exploding start). Type-Ia supernovae are an especially important class of transient objects, giving us the ability to measure the

rate of expansion of the Universe. A type-Ia supernova occurs when a white dwarf, typically in a binary system, gains enough mass (i.e., beyond the Chandrasekhar limit), triggering a violent explosion. Because of the consistency in their formation we can assume the intrinsic brightness and its dynamics are the same wherever they occur in the Universe. Thus, when observing a type-Ia supernova we measure its apparent brightness on Earth and use its known intrinsic brightness to infer the distance of the object from Earth via the inverse square law. Objects with this property are called standard candles. By comparing the distance of the object with its measured redshift we can better understand the rate of expansion of the Universe. In fact, the Supernova Cosmology Project and the High-z Supernova Search Team used exactly these techniques with type-Ia supernova to discover that the expansion of the Universe was accelerating, earning the group leaders the 2011 Nobel Prize in Physics.

As we can see, obtaining a large dataset of type-Ia supernova is incredibly useful for cosmology, and given the scale of LSST we ought to be able to accomplish this goal. But it is a serious challenge to accurately classify all of the different types of transients that LSST will observe. In particular we require a model to distinguish between type-Ia supernova and non-type-Ia supernova (a binary prediction problem). One of the major challenges is related to the type of data that LSST collects. Astronomical data tends to fit into two main categories: photometry and spectroscopy. The main difference between photometry and spectroscopy is the width of the wavelength bands used to filter the incoming light. Photometry uses a relatively small number (a couple to a few dozen) of wide broad band-pass wavelength bands, requiring only short exposure times (a few seconds) and enabling rapid collection of a massive amount data. Spectroscopy, on the other hand, is on the opposite end of the spectrum (pun intended), separating the incoming light into thousands of narrow bands (on the order of a angstrom) and giving much more information on the object we are observing. However this comes at the cost of time, with exposure times on the order of tens of minutes to hours, and requiring more complicated instrumentation. LSST will be collecting photometric data

using six bands. The collection of photometric data is what allows for the scale and quick cadence of LSST.

As noted, our goal is to train a model that can predict from LSST data if an object is a type-Ia supernova or not. To complete this task we will require a high quality training dataset, in which we know the correct label of each object. But, assigning this label can be very difficult purely from photometric data, even for humans. So in order to build our training dataset, we must make spectroscopic observations from which the objects can be more precisely labeled. However, LSST will observe far too many transients for each one to be spectroscopically labeled, resulting in a setting ripe for the application of active learning. In Chapter 6 we cover our investigations on this problem, showing that techniques from active learning can applied successfully. This work is designated as our second application, labeled "Active Learning for Astronomy", in Figure 1.1.

1.3 Learning over Sets

Another frequently occurring theme in our work is the need to learn over set-based data. In many practical settings one needs to build a machine learning model that can act on variable sized inputs, or where there are meaningful relationships between the data points such as permutation invariance, where re-orderings of the input should lead to the same answer. For instance, permutation invariance often occurs in optimal experimental design, where the units within the experiment are exchangeable and so re-orderings of the units should not have any change on the utility of the design. In essence, we would like our methods to produce the same value of expected information gain for all possible orderings of the proposed design. In our work on variational methods for Bayesian optimal experimental design, we used recent developments on constructing permutation invariant neural networks. This forms a critical component of our proposed neural architecture, which allows us to directly encode the desired

invariance into our architecture while also enabling weight-sharing, which is known to lead to benefits in the computational cost of neural network models.

We also describe a standalone project on developing machine learning algorithms over set-based data, which is covered in Chapter 7. In this work, our goal was to produce a classifier for discriminating between stars and galaxies, with a focus on the data that are expected to be produced by LSST. Star/Galaxy classification is one of the early steps in a typical astronomical data-processing pipeline, making it a critical component with significant downstream impacts.

In space, stars appear as point sources of light while galaxies have spatial extent, making this a relatively simple classification problem for space-based observations. But for ground-based telescopes (like LSST), the difficulty is magnified due to the presence of the atmosphere. As light passes through the atmosphere it is spread out, making stars' point sources appear spatially distributed and widening galaxies even further. More technically speaking, the incoming light is being convolved with a point spread function (PSF) due to its interaction with the atmosphere and telescope optics. Moreover, due to atmospheric turbulence, this point spread function is constantly changing in time and in space. This creates a confounding factor that must be accounted for to produce a good classifier.

Each LSST exposure will observe on the order of a thousand objects per sensor, and the point spread function of the atmosphere will change from one exposure to the next. Even more challenging, it will also change spatially, from one point in the focal plane to the next, even within the same exposure. This could result in a star in one exposure appearing larger than a galaxy in another, so that a classifier cannot solely rely on spatial extent but must factor in the influence of the point spread function. We develop a novel neural architecture that is able to achieve strong performance on this challenging problem. At a high level, we crop out small images of each individual object in the focal plane and build a classifier that

takes in this set of images along with the spatial coordinates of each image. The classifier is specifically designed to be able to compare the images (and their positions) within the set and produce a classification for each object. Chapter 7 shows the success of our method while other, more standard, approaches fail.

Chapter 8 concludes the thesis with a discussion of what we believe are the most promising directions for future work. We particularly emphasize our work on variational Bayesian optimal experimental design, as we anticipate a great number of open problems that can be attacked with these techniques, potentially producing major benefits to science generally.

Chapter 2

An Illustrative Example

> One important idea is that science is a means whereby learning is achieved, not by mere theoretical speculation on the one hand, nor by the undirected accumulation of practical facts on the other, but rather by a motivated iteration between theory and practice
>
> George E. P. Box

In this chapter we provide a concrete optimal experimental design example to illustrate the high-level ideas discussed in the previous chapter. Our example is based on an example from Goos and Jones [2011, Chapters 2 and 3] from the field of microbiology. In Section 2.1 we state the problem, then in Sections 2.2 and 2.3 we detail a solution from both the classical and Bayesian perspectives, respectively.

2.1 The Experimental Problem

For this example we imagine ourselves as statistical consultants hired by a biotechnology company, Phylo-Networks, to help design an experiment for the development of a new product. Phylo-Networks has developed a proprietary strain of *Bacillus subtilis* that produces a molecule which is able to inhibit the growth of harmful bacteria. Currently the yield from their cultures are too small for them to create a commercial product. But they believe if they can increase the yield, they can use this molecule to develop a new class of antibiotic drugs that can save many lives, in addition to using it as an agent for food safety, which can both prevent food-borne illnesses and reduce food waste.

The current process starts by culturing the strain of *B. subtilis* in a specific medium at 37°C for 24 hours in a flask that is being gently shaken. The culture is then put into a new flask with a different medium at a temperature between 30°C and 33°C for 48 hours. The contents of the flask are then centrifuged to remove the bacterial cells, leaving a solution that is ready for extraction of the molecule. The extraction process starts with 100 ml of the solution; four different solvents, methanol, ethanol, propanol and butanol can be added to the solution at concentrations between 0 and 10 mg/ml. In addition the pH of the solution can be adjusted between 6 and 9, and it can be left to sit anywhere between 1 and 2 hours. The yield of the extraction is then measured with chromatography and is expressed in mg per 100 ml.

Phylo-Networks believes the extraction process can be greatly improved to increase the yield of molecule per culture extraction. They wish to design an experiment to determine the effects of the 6 factors (methanol, ethanol, propanol, butanol, pH and time) on the extraction process, and have hired us for our statistical expertise. If they can better understand the effects of these factors, they can determine settings to increase the yield of the extraction. Furthermore, based on prior knowledge, they believe the interaction effects and higher order effects (quadratic or greater) do not play a role in the yield within the range of settings

considered. Thus we determine to use the following main-effects model to analyze our results, and it is for this model that we seek an optimal design:

$$y_i = \theta_0 + \theta_1 d_{i,1} + \theta_2 d_{i,2} + \theta_3 d_{i,3} + \theta_4 d_{i,4} + \theta_5 d_{i,5} + \theta_6 d_{i,6} + \epsilon_i, \tag{2.1}$$

where y_i is the yield from experimental run i, θ_j for $j = 0 \ldots 6$ is the effect of factor j (including an intercept term), and $d_{i,j}$ is the setting of the jth factor for the ith experimental run. We refer to the quantities $d_{i,j}$ as our design variables. Finally, ϵ_i is a random error that we assume to be normally distributed with mean zero and with variance σ, which we assume has known value $\sigma = 1$. Due to budget limitations, we will only be able to conduct 16 runs. Our goal is then to find settings for all $d_{i,j}$ that will make our experiment maximally informative. We denote the settings of all the runs as a 16×7 matrix, D, often called the design matrix. (The first column of D is all ones.) The yields from all individual runs are denoted by the 16 dimensional column vector Y, all main effects will be denoted by the 7 dimensional column vector θ, and ϵ is a 16 dimensional column vector representing the random error from each run. Thus we may compactly represent our main-effects model as:

$$Y = D \cdot \theta + \epsilon. \tag{2.2}$$

Note that throughout this thesis we refer to the individual experimental runs as experimental units, and the ith row of D denotes the settings for the ith experimental unit.

2.2 A Classical Approach

In the preceding section, we stated that our goal was to find the settings of the design matrix that will lead to our experiment being maximally informative. But what exactly is meant by "maximally informative"? In experimental design, there are a number of utility functions

that we may use to define precisely what is meant, and in this section we define one of the most common utility functions from classical statistics, called *D-optimality* [Goos and Jones, 2011].

Imagine that we have conducted our experiment with design settings D, and observed yields Y. We could then use ordinary least squares to estimate the effects of each factor:

$$\hat{\theta} = (D^T \cdot D)^{-1} \cdot D \cdot Y. \tag{2.3}$$

Furthermore, the covariance matrix of this estimator is,

$$\operatorname{cov}\left(\hat{\theta}\right) = \sigma^2 \left(D^T \cdot D\right)^{-1}, \tag{2.4}$$

in which the diagonal elements correspond to the variances of each individual effect estimate, and the off-diagonal elements to the covariances between each different pair of effects. Intuitively, we would like our variance and covariance of the estimators to be as small as possible, allowing us to tightly constrain the most likely regions of our estimators (i.e., produce the smallest possible confidence regions). We can visualize the shape and extent of a confidence region on our 7-dimensional estimator as an ellipse defined by the positive definite covariance matrix expressed in Equation (2.4), with a scaling factor that depends on the degree of confidence, and so it is reasonable to try to minimize the volume of this ellipse. Using basic linear algebra, we can compute the volume of this ellipse by computing the determinant of the matrix $\operatorname{cov}(\hat{\theta})$. Thus, we would like to minimize the determinant of the matrix $(D^T \cdot D)^{-1}$, or equivalently to maximize the determinant of $(D^T \cdot D)$. (Note that we can ignore the constant scaling factor as it does not affect our resulting optimization problem.) Maximizing the determinant of $(D^T \cdot D)$ is known as the D-optimality criterion.

In order to maximize this quantity, we can use the *coordinate exchange* algorithm, which is

frequently used to optimize experimental designs Meyer and Nachtsheim [1995b], although we could also use a variety of other algorithms from mathematical optimization. We assume that we have coded our design variables between the ranges of -1 to 1. We then choose a starting design, D, by randomly sampling design variables uniformly within this range (excluding the first column which is set to 1). We then loop through each element of the matrix D (again excluding the first column), and calculate the D-optimality criterion for three different values: the original design matrix D, the design matrix where the current element is set to 1 with all other elements left the same, and the design matrix with the current element set to -1, again with all other elements left the same. We then update our design matrix to the setting with the highest D-optimality score of the three. After looping through every element of the matrix, we check if any elements of the design were updated, and if so we iterate through again; if not the algorithm terminates. We also terminate the algorithm when a maximum number of iterations is reached, although in practice the procedure typically terminates after only a few iterations.

The coordinate exchange algorithm finds a design matrix that is a local optimum within the neighborhood of the starting design. Since this local optimum may not be a global optimum, it is common to re-run the algorithm from multiple starting designs and select the best. For the purposes of our example, we run it from 100 different starting designs, after which we have identified the following design, which we discuss in more detail in Section 2.4:

Intercept	Methanol	Ethanol	Propanol	Butanol	pH	Time
1.00	-1.00	-1.00	-1.00	-1.00	-1.00	1.00
1.00	-1.00	1.00	-1.00	1.00	1.00	-1.00
1.00	1.00	-1.00	-1.00	1.00	1.00	1.00
1.00	-1.00	-1.00	1.00	-1.00	1.00	-1.00
1.00	1.00	-1.00	1.00	1.00	-1.00	-1.00
1.00	1.00	1.00	-1.00	-1.00	-1.00	-1.00
1.00	-1.00	1.00	1.00	1.00	-1.00	1.00
1.00	1.00	1.00	1.00	-1.00	1.00	1.00
1.00	-1.00	1.00	1.00	-1.00	1.00	-1.00
1.00	1.00	-1.00	1.00	1.00	-1.00	-1.00
1.00	1.00	-1.00	-1.00	-1.00	1.00	-1.00
1.00	1.00	1.00	-1.00	1.00	1.00	1.00
1.00	1.00	1.00	1.00	-1.00	-1.00	1.00
1.00	-1.00	1.00	-1.00	1.00	-1.00	-1.00
1.00	-1.00	-1.00	1.00	1.00	1.00	1.00
1.00	-1.00	-1.00	-1.00	-1.00	-1.00	1.00

Recall that the ranges of the chemical concentration factors are 0 to 10 mg/ml, while the pH can range from 6 to 9 and the time from 1 to 2 hours. All design variables are linearly transformed to map these intervals to the range $[-1, 1]$.

2.3 A Bayesian Approach

We now consider how we would solve our experimental design problem from a Bayesian perspective. First, we should define a prior distribution over our model parameters, θ, which encodes our current knowledge about the parameter values before conducting the experiment.

We select a weakly informative prior centered at zero, as we do not know whether each factor may increase or decrease the yield amount. Specifically, we select a multivariate normal distribution with zero mean and diagonal covariance with 5 on the main diagonal, $p(\theta) \sim \mathcal{N}(0, 5I)$, where I is the 7×7 identity matrix and 0 denotes the 7 dimensional vector of all zeros. We have already assumed that our random error is normally distributed with mean 0 and standard deviation 1, so that our likelihood is $y_i \sim \mathcal{N}(0, \sigma)$. Recall that we assume $\sigma = 1$ is known. We can then apply Bayes rule to compute the posterior distribution over our parameters, representing our state of knowledge after conducting the experiment and observing the results:

$$p(\theta|Y, D) = \frac{p(Y|\theta, D)p(\theta)}{p(Y|D)}. \tag{2.5}$$

Normally the posterior distribution is intractable to compute, the central challenge being the marginal likelihood $p(Y|D)$. In this case, however, we can actually solve for the posterior distribution analytically thanks to the conjugate nature of the the normal likelihood with normal prior. Carrying out this calculation is more or less an exercise in patience and completing the square, but after doing so we find the posterior distribution:

$$p(\theta|Y, D) = \mathcal{N}\Big(\sigma^{-2}\big(\sigma^{-2}D^TD + \frac{1}{5}I\big)^{-1}D^TY, \big(\sigma^{-2}D^TD + \frac{1}{5}I\big)^{-1/2}\Big). \tag{2.6}$$

Assuming we had conducted our experiments on design D and observed yields Y, we could measure the amount of information the experiment gave, by subtracting the entropy of the posterior distribution from the entropy of the prior distribution. This difference corresponds to a quantity called the information gain (IG):

$$IG(Y, D) = H[p(\theta)] - H[p(\theta|Y, D)], \tag{2.7}$$

where $H(\cdot)$ is the (differential) entropy function, a common measure of the uncertainty for a distribution. The entropy for a probability mass function, $p(x)$, is defined to be:

$$-\sum_x p(x)\log p(x) \tag{2.8}$$

and the differential entropy of probability density function, $p(x)$, is defined to be:

$$-\int p(x)\log p(x)\,dx \tag{2.9}$$

. However, our purpose is to choose the most informative design *before* conducting the experiment and observing Y. We cannot evaluate the information gain directly without the value of Y; instead our score should depend only on the design, D. We can achieve this by taking the expectation of the information gain with respect to $p(Y|D)$, giving the *expected information gain* (EIG):

$$EIG(D) = \mathbb{E}_{p(y|d)}\big[\, H[p(\theta)] - H[p(\theta|Y,D)]\,\big]. \tag{2.10}$$

The EIG is a widely used utility function for Bayesian optimal experimental design, and has been shown to be an optimal choice; see Bernardo and Smith [2009, Chapter 2].

Unfortunately, the expected information gain is typically intractable to compute, and much of this thesis will focus on proposing approximation strategies that are both accurate and computationally efficient. However, for our current example, it turns out that the EIG can be computed analytically. The differential entropy of a multivariate normal distribution with dimension N, mean μ and covariance Σ is given by:

$$H\Big[\mathcal{N}(\theta|\mu,\Sigma)\Big] = \frac{1}{2}\log\big(\,(2\pi e)^N \det(\Sigma)\,\big). \tag{2.11}$$

Here we can see that the entropy calculation only depends on the covariance matrix of the

normal distribution, and as we can see from Eq. (2.6), the posterior distribution's covariance depends only on the proposed design D and the prior covariance, both of which are known before conducting the experiment. Thus we can analytically calculate the differential entropy of the posterior distribution for any proposed design. For analogous reasons the differential entropy of the prior can also be easily calculated. Thus for our current example, we can analytically calculate our utility function, the expected information gain:

$$EIG(D) = \frac{1}{2}\log\Big[\ \frac{\det\big(5I\big)}{\det\Big(\big(\sigma^{-2}D^TD + \frac{1}{5}I\big)^{-1}\Big)}\ \Big]. \tag{2.12}$$

. We again apply the coordinate exchange algorithm with 100 random starts, but now using the expected information gain as our utility function, and achieve the following design:

Intercept	Methanol	Ethanol	Propanol	Butanol	pH	Time
1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00
1.00	1.00	-1.00	-1.00	1.00	-1.00	-1.00
1.00	-1.00	1.00	1.00	1.00	-1.00	-1.00
1.00	1.00	1.00	-1.00	-1.00	1.00	-1.00
1.00	-1.00	-1.00	-1.00	1.00	1.00	1.00
1.00	1.00	1.00	1.00	1.00	-1.00	1.00
1.00	1.00	-1.00	-1.00	-1.00	-1.00	1.00
1.00	-1.00	-1.00	1.00	1.00	1.00	-1.00
1.00	-1.00	-1.00	1.00	-1.00	1.00	1.00
1.00	1.00	-1.00	1.00	1.00	-1.00	1.00
1.00	-1.00	1.00	-1.00	-1.00	-1.00	1.00
1.00	-1.00	1.00	-1.00	1.00	1.00	1.00
1.00	1.00	1.00	-1.00	1.00	1.00	-1.00
1.00	1.00	-1.00	1.00	-1.00	1.00	-1.00
1.00	-1.00	1.00	1.00	-1.00	-1.00	-1.00
1.00	1.00	1.00	1.00	-1.00	1.00	1.00

2.4 Analysis

Observe that both designs found form orthogonal matrices, meaning that $D^T D$ is a diagonal matrix. In our classical approach this would mean that there is no covariance between our effect estimates; in the Bayesian approach it means that (assuming our prior covariance is diagonal), our posterior covariance remains diagonal. In fact, both designs fall within the Plackett-Burman class of designs, which are optimal for this problem, which we could have chosen from a pre-computed table of designs. In light of this point it would be reasonable to ask, why go through all of the trouble of defining the D-optimality criterion or expected

information gain and solving a mathematical optimization problem, when the designs we produced could have been found from a table with minimal effort? The answer, of course, is that we were lucky – our experimental constraints matched the conditions for which a Plackett-Burman design (or other standard design) could be generated. For example, if the number of experimental runs available to us was not a multiple of 4, then a Plackett-Burman design would not have existed; or, if we needed to consider higher order effects such as quadratic terms or interaction terms, then a Plackett-Burman design may not have existed for the number of runs in our budget. Had we possessed more complex prior knowledge, resulting in a prior distribution with non-diagonal covariance matrix, then Plackett-Burman designs would no longer be optimal. Employing more complicated distributional assumptions or a more complicated process model (e.g., not linear in the effects) would again mean that no standard design would be available. This example illustrates that the utility functions we selected can be used to generate standard designs from statistical tables when our experiment matches the conditions for them to apply. But taking an optimization perspective allows us to be far more general, and can help us select the best (or better) designs even when (as is common) our experimental conditions fail to conform to any typical, standard case.

We were also fortunate in this example that evaluating our utility functions could be performed analytically, and that we had a robust optimization algorithm available for finding an optimal design. In practice, this is rarely the case and evaluating these utility functions is computationally intractable. The next three chapters focus on developing approximation algorithms to evaluate and optimize the expected information gain. We show that our proposed methods greatly advance the state of the art, producing more accurate approximations while improving efficiency (in both samples required and total computational time).

Chapter 3

Variational Inference for Bayesian Optimal Experimental Design

> I think that it is a relatively good approximation to truth—which is much too complicated to allow anything but approximations—that mathematical ideas originate in empirics.
>
> John von Neumann

In this chapter we introduce variational techniques for approximating and optimizing the expected information gain. We start by formally stating the Bayesian OED problem, which is general enough to encompass a wide variety of controlled experiments and even some observational studies. This framework has been used by scientists and engineers in many fields to help design their studies to be as informative as possible. Next, we give a general discussion of variational inference, an approximate inference technique which has helped make Bayesian methods more practical and usable by researchers. In particular, we show how variational techniques can be used to approximate the expected information gain. We

use these approximation strategies in the subsequent two chapters to achieve state-of-the-art results in terms of both accuracy and efficiency.

3.1 Problem Setup

3.1.1 Bayesian Model

In our work we adopt a Bayesian approach to experimental design, and so we start by defining our Bayesian model, which could cover a broad number of circumstances. We denote the design variables (which are under the experimenter's control) as d; in the example from Chapter 2 this would represent the concentrations for the different solvents, pH, and time for all experimental units. As another example, suppose we plan to conduct an observational study in the field of ecology to better understand how environmental conditions (e.g., elevation, tree coverage, or temperature) affect the breeding habitats of a certain species in the region, information which we have for our region, which we divide up into plots of 1km^2. The design d could then represent which plots we observe, how often, and at what time. We could potentially update d on a daily basis to account for the information we have already received over the duration of the study. Note that in these examples and in any other use case of optimal experimental design we will also have constraints on what designs are feasible and may have additional objectives we wish to optimize for. In chapter 5 we will show how the methods we propose benefit the use of a wide variety of optimization algorithms facilitating their use in a wide variety of practical use cases.

Next, we denote our experimental observations as y. These are the quantities we directly observe after setting our design variables d and running the experiment. From the example in Chapter 2, y corresponds to the yields we measure for all our experimental units. In our ecology example, y would represent the number of individuals of the species counted in the

plots we decide to observe.

Finally we denote our latent variables by θ, corresponding to the quantities we cannot directly observe but are interested in studying. Learning more about these variables is typically the reason for conducting our experiment. We assume that we are given some functional form that relates our latent variables θ and design variables d to the observations of the experiment. In the microbiology example, we assumed a linear model with Gaussian error; the latent variables comprised the factor effects that mediate how the setting of the design variables influences the yield y. In the ecology example, we would have some other functional relationship from theory, or could use a more generic statistical model to relate the environmental conditions of each plot of land to the presence or frequency of the species of interest being on the plot. We quantify this relationship via a likelihood function $p(\theta|y, d)$ which consists of the assumed functional form, often called the process model, and a random component. In our microbiology example, the process model was a linear model and the random component was the Gaussian noise. For the ecology study, the process model would consist of whatever functional form we assume, and the random component would likely be a Bernoulli or Poisson distribution (for event or count measurements, respectively) parameterized by the process model. In addition to the likelihood, we also have prior knowledge on our latent variables which is encoded in our prior distribution $p(\theta)$. Then, via Bayes rule, we can write the posterior distribution, which represents our state of knowledge in the latent variables after conducting the experiment:

$$p(\theta|y, d) = \frac{p(y|\theta, d)p(\theta)}{p(y|d)} \tag{3.1}$$

These relationships are shown pictorially in figure 3.1. Note that the design setting, d, is selected by the experimenter, and thus non-random. In this sense, it is trivially independent of the true parameter value θ.

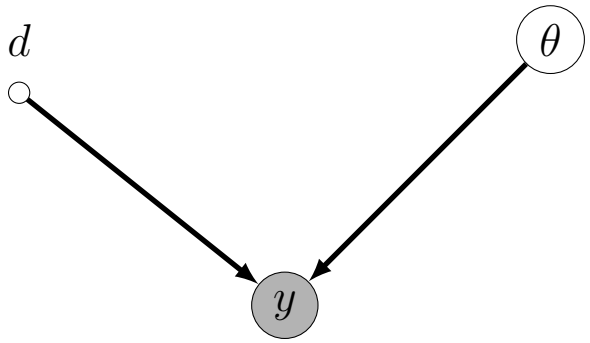

Figure 3.1: Experimental Setup. Our observations y (shaded) are the result of a random process that depends on both the latent random variables θ and our (to be selected) experimental design d. Note that the design setting, d, is selected by the experimenter, and thus non-random. In this sense, it is trivially independent of the true parameter value θ.

Of course, it is possible we may not possess an analytic form of the likelihood, but instead resort to running simulations that implicitly describe the relationship between the three quantities d, θ, and y. This is a fairly common situation in many fields such as population genetics, astrophysics, and systems biology, in which more is known about the random process leading to a result than about the distribution of the result itself. Inference techniques in this context are often called *likelihood-free* or *approximate Bayesian computation* (ABC) methods. At least one of the variational forms we discuss can still optimize experimental design in this more difficult setting, although we do not directly investigate such models in this thesis.

3.1.2 The Expected Information Gain

In this work we focus on evaluating the expected information gain, a commonly used utility function in Bayesian optimal experiment design [Chaloner and Verdinelli, 1995, Ryan et al., 2016]. Assuming we have conducted our experiment at design variables d and observed outcome y, we can quantify the amount of information we received from our experiment via

the information gain,

$$IG(y, d) = H\big[\, p(\theta) \,\big] - H\big[\, p(\theta|y, d) \,\big] \tag{3.2}$$

where $H[\cdot]$ is the (differential) entropy function. Further the conditional (differential) entropy of the far right is often not possible to compute in closed form due to the intractability of the posterior distribution in most cases. However, the information gain cannot be evaluated before conducting the experiment, since it requires knowing the outcomes y. To remove the dependence on y, we take the expectation of the information gain with respect to the outcomes, $p(y|d)$, giving us the expected information gain:

$$EIG(d) = \mathbb{E}_{p(y|d)}\big[\, H[p(\theta)] - H[p(\theta|y, d)] \,\big] \tag{3.3}$$

The EIG has sound theoretical justifications, and is proven to be optimal in certain settings [Sebastiani and Wynn, 2000, Bernardo and Smith, 2009, chapter 2].

The experimenter next seeks the solution to $\text{argmax}_{d\in\mathcal{D}}\, EIG(d)$, where $\mathcal{D}$ is the set of all feasible designs. The set of all feasible designs is defined by the points, d, at which the experiment can be run. In the case of our microbiology example this is the space of all 16×6 matrices with entries falling in interval $[-1, 1]$. In general, different experiments have their own unique constraints, but all experimental design problems boil down to solving a mathematical optimization problem.

Both the evaluation of the expected information gain and the optimization problem are often intractable, so that approximation strategies need to be used. One of the most common approximation strategies used for evaluating the EIG is nested Monte Carlo (NMC) [Rainforth et al., 2018], which can then be used in many different optimization algorithms [Ryan et al., 2016]. We will discuss and build off this estimator shortly showing a broader framework for

approximate inference for the EIG. The intractability of evaluating the expected information gain can be viewed as a consequence of its dependence on the posterior distribution, which itself is often intractable to compute exactly. Even more difficult, we must compute the posterior distribution for every possible y in the expectation, so that the expected information gain is often called "doubly intractable". Intuitively, the optimization problem is intractable for two reasons: (1) the intractability of the its utility function, the EIG; and (2) because the constraints defining the feasible set of designs could result in having to solve a combinatorial optimization problem, or more generally some other type of NP-hard optimization problem. However, as we discuss in the sequel, variational methods can provide efficient and accurate approximation techniques for both evaluation and optimization.

3.2 Variational Inference

As discussed in the preceding section, the expected information gain is often intractable to evaluate, and one of the primary focuses of this thesis is to investigate variational methods for approximating expected information gain. Variational methods are a broad class of techniques that are widely used to make efficient and accurate approximations to intractable inference problems [Wainwright et al., 2008, Blei et al., 2017]. They generally follow the pattern of first proposing a parameterized function meant to approximate an intractable quantity of interest, for example the posterior distribution or marginal likelihood, then deriving a bound to an estimator objective, such as the maximum likelihood, and finally optimizing the parameters of the variational function with respect to the derived bound. After the optimization, we can use the variational form in place of the intractable quantity to perform tasks such sampling or evaluating samples.

We give a concrete example by deriving the so-called *evidence lower bound*, which is the most commonly used variational bound in Bayesian statistics [Blei et al., 2017]. In this

setting, we would like to approximate the posterior distribution $p(\theta|y,d)$. We first define a family of densities $\mathcal{Q}$ over the latent variables, θ. Each point in this set, $q_\phi(\theta) \in \mathcal{Q}$, is indexed by parameters ϕ; for example, if θ is continuous, unbounded and one dimensional, $\mathcal{Q}$ could be the set of all normal distributions and $\phi = \{\mu, \sigma^2\}$, the mean and variance of a specific normal distribution. We would then like to find the q_ϕ that best approximates the true posterior. We use the Kullback-Leibler (KL) divergence [Kullback and Leibler, 1951] to quantify the quality of the approximation. This turns our inference problem into the task of solving the optimization problem,

$$q_\phi^*(\theta) = \underset{q_\phi \in \mathcal{Q}}{\operatorname{argmin}} \, KL(q_\phi(\theta)||p(\theta|y,d)). \tag{3.4}$$

Unfortunately, solving Eq. (3.4) is still intractable in the general case: it requires us to evaluate the posterior distribution. However, by applying the definition of conditional densities to the posterior we obtain,

$$KL\big(\, q_\phi(\theta)||p(\theta|y,d)\,\big) = \mathbb{E}\big[\log q_\phi(\theta)\big] - \mathbb{E}\big[\log p(\theta,y|d)\big] + \log p(y|d), \tag{3.5}$$

and by rearranging terms we can define a tractable lower bound to the marginal likelihood or evidence, called the evidence lower bound (ELBO):

$$\begin{aligned} \log p(y|d) &= KL(q_\phi(\theta)||p(\theta|y,d)) + \mathbb{E}\left[\log p(\theta,y|d)\right] - \mathbb{E}\left[\log q_\phi(\theta)\right] \\ &\geq \mathbb{E}\left[\log p(\theta,y|d)\right] - \mathbb{E}\left[\log q_\phi(\theta)\right] \\ &\stackrel{\text{def}}{=} \text{ELBO}\left(q_\phi(\theta)\right) \end{aligned} \tag{3.6}$$

.

Since $\log p(y|d)$ is fixed, we can see that maximizing the ELBO is equivalent to minimizing the forward KL-divergence, which defined our original training objective. This derivation illustrates the essential steps in developing variational methods: first defining a set of den-

sities meant to approximate an intractable quantity of interest, second deriving a bound to a training objective that characterizes the closeness of our variational approximation to the true quantity, and finally optimizing this bound within the defined set. The ELBO is the most widely used training objective when applying variational methods to Bayesian inference; however,in the next section we discuss bounds proposed in Foster et al. [2019] that are designed for directly approximating the expected information gain. These bounds will be at the center of our investigation.

3.3 Variational Inference for Bayesian Optimal Experimental Design

In this section we study three different estimators for the expected information gain: nested Monte Carlo, the posterior estimator and variational nested Monte Carlo. In addition, both nested Monte Carlo and variational nested Monte Carlo have alternative forms, which change the nature of the bound and which we also study. We start by rewriting the expected information gain in a form that will expose the motivation behind these estimators. First we expand out the entropy terms from equation (3.3) writing everything as integrals with respect to the counting or the Lebesgue measure, with out loss of generality:

$$\begin{aligned} EIG(d) &= -\iint p(y|d)p(\theta)\log p(\theta)\,dy\,d\theta + \iint p(\theta|y,d)p(y|d)\log p(\theta|y,d)\,dy\,d\theta \\ &= -\int p(\theta)\log p(\theta)\,d\theta + \iint \frac{p(y|\theta,d)p(\theta)p(y|d)}{p(y|d)}\log p(\theta|y,d)\,dy\,d\theta \\ &= -\iint p(y|\theta,d)p(\theta)\log p(\theta)\,dy\,d\theta + \iint p(y|\theta,d)p(\theta)\log p(\theta|y,d)\,dy\,d\theta \\ &= \iint p(y|\theta,d)p(\theta)\log\frac{p(\theta|y,d)}{p(\theta)}\,dy\,d\theta \end{aligned} \tag{3.7}$$

Rewriting this as an expectation and applying the law of conditional probability we then get

these convenient forms for the EIG:

$$EIG(d) = \mathbb{E}_{p(\theta,y|d)}\left[\log\left(\frac{p(\theta|y,d)}{p(\theta)}\right)\right] \tag{3.8a}$$

$$= \mathbb{E}_{p(\theta,y|d)}\left[\log\left(\frac{p(y,\theta|d)}{p(\theta)p(y|d)}\right)\right] \tag{3.8b}$$

$$= \mathbb{E}_{p(\theta,y|d)}\left[\log\left(\frac{p(y|\theta,d)}{p(y|d)}\right)\right] \tag{3.8c}$$

Nested Monte Carlo: One common approach to approximating EIG is to use a nested Monte Carlo (NMC) estimator [Myung et al., 2013, Vincent and Rainforth, 2017, Rainforth et al., 2018]:

$$\hat{\mu}_{NMC} = \frac{1}{N}\sum_{n=1}^{N}\log\frac{p(y_n|\theta_{n,0},d)}{\frac{1}{M}\sum_{m=1}^{M}p(y_n|\theta_{n,m},d)} \tag{3.9}$$

$$\text{where} \quad \theta_{n,m} \sim p(\theta) \quad \text{and} \quad y_n \sim p(y|\theta_{n,0},d)$$

The intuition behind the NMC estimator is to use a Monte Carlo sample-based estimator for the outer expectation (the N samples $\theta_{n,0}$ and y_n), and then estimate the denominator in Eq. (3.8c) by generating a set of M "inner" or "nested" samples $\theta_{n,m}$ for each "outer" sample n, and use these inner samples to marginalize over θ and estimate $p(y|d)$. Rainforth et al. [2018] showed that NMC is a consistent estimator converging as $N, M \to \infty$. They also showed that it is asymptotically optimal to set $M \propto \sqrt{N}$, resulting in an overall convergence rate of $\mathcal{O}(T^{-\frac{1}{3}})$, where T is the total number of samples drawn (i.e., $T = NM$ for NMC). However, this is much slower than the $\mathcal{O}(T^{-\frac{1}{2}})$ rate of standard Monte Carlo estimators [Robert and Casella, 1999], in which the total number of samples is simply $T = N$.

The slow convergence of the NMC estimator can be limiting in practical applications of BOED. The inefficiency can be traced to requiring an independent estimate of the marginal likelihood, $p(y_n|d)$, for each y_n. Inspired by this idea, Foster et al. [2019] proposed to em-

ploy techniques from variational inference by defining a functional approximation to either $p(\theta|y,d)$ or $p(y|d)$, and allowing these estimators to amortize across the samples of y_n for more efficient estimation of the EIG. In this thesis we focus on two of the four estimators they proposed: the posterior estimator and variational nested Monte Carlo.

Posterior Estimator: The posterior estimator is an application of the Barber-Agakov bound to BOED, which was originally proposed for estimating the mutual information in noisy communication channels [Barber and Agakov, 2003]. It requires defining a variational approximation $q_\phi(\theta|y,d)$ to the posterior distribution, resulting in a lower bound to the EIG:

$$\begin{aligned} EIG(d) \geq \mathcal{L}_{post}(d) &\triangleq \mathbb{E}_{p(\theta,y|d)}\left[\log\left(\frac{q_\phi(\theta|y,d)}{p(\theta)}\right)\right] \\ &\approx \frac{1}{N}\sum_{n=1}^{N}\log\frac{q_\phi(\theta_n|y_n,d)}{p(\theta_n)} \qquad \text{where} \quad y_n,\theta_n \sim p(y,\theta|d). \end{aligned} \tag{3.10}$$

By maximizing this bound with respect to the variational parameters ϕ, we can learn a variational form to efficiently estimate the EIG. A Monte Carlo estimate of this bound converges with rate $\mathcal{O}(T^{-\frac{1}{2}})$, and if the true posterior distribution is within the class of functions defined by the form q_ϕ, the bound can be made tight (also dependent on the optimization) [Foster et al., 2019]. To prove this as a lower bound one simplly has to consider the difference between the EIG and the posterior estimtor, which results in the expectation of a KL-divergence.

Variational Nested Monte Carlo: The second bound we discuss is variational nested Monte Carlo (VNMC). It is closely related to NMC, but differs by applying a variational approximation $q_\phi(\theta|y,d)$ as an importance sampler to estimate the marginal likelihood term in NMC:

$$EIG(d) \quad \leq \quad \mathcal{U}_{VNMC}(d,M) \quad \triangleq \quad \mathbb{E}\left[\log\frac{p(y|\theta_0,d)}{\frac{1}{M}\sum_{m=1}^{M}\frac{p(y,\theta_m|d)}{q_\phi(\theta_m|y,d)}}\right], \tag{3.11}$$

where the expectation is taken with respect to $y, \theta_{0:M} \sim p(y, \theta_0|d) \prod_{m=1}^{M} q_\phi(\theta_m|y, d)$.

By minimizing this upper bound with respect to the variational parameters ϕ, we can learn an importance distribution that allows for much more efficient computation of the EIG. Note that if $q_\phi(\theta|y, d)$ exactly equals the posterior distribution, the bound is tight and requires only a single nested sample ($M = 1$). Even if the variational form does not equal the posterior, the bound remains consistent as $M \to \infty$. Finally, it is worth noting that by taking $q_\phi(\theta|y, d) = p(\theta)$, the estimator simply reduces to NMC.

It was further shown by Foster et al. [2020] that VNMC can be easily made into a lower bound by including θ_0 (the sample from the prior) when estimating the marginal likelihood, a method we denote as contrastive VNMC (CVNMC):

$$EIG(d) \quad \geq \quad \mathcal{L}_{CoVNMC}(d, M) \quad \triangleq \quad \mathbb{E}\left[\log \frac{p(y|\theta_0, d)}{\frac{1}{M+1}\sum_{m=0}^{M} \frac{p(y,\theta_m|d)}{q_\phi(\theta_m|y,d)}}\right], \tag{3.12}$$

where the expectation is again taken with respect to $y, \theta_{0:M} \sim p(y, \theta_0|d) \prod_{m=1}^{M} q_\phi(\theta_m|y, d)$. We can also employ this same technique to regular NMC to estimate both lower and upper bounds.

Note that the upper bound (3.11) and lower bound (3.12) are particularly useful when evaluating the performance of methods in settings where ground truth is not available. In these cases we can examine the bound pairs produced by NMC and by VNMC to assess which set more tightly constrains the true value.

In the next chapter we propose a variational form that is capable of amortizing over designs to greatly improve computational efficiency. Furthermore, this form encompasses a broad and flexible family of probability densities, leading to superior approximation quality compared to the forms proposed in Foster et al. [2019] and Foster et al. [2020].

Chapter 4

Design Amortization for Bayesian Optimal Experimental Design

In this chapter we build upon successful variational approaches, which optimize a parameterized variational model with respect to bounds on the EIG. Past work focused on learning a new variational model from scratch for each new design considered. Here we present a novel neural architecture that allows experimenters to optimize a single variational model that can estimate the EIG across the design space, including potentially infinitely or continuously many designs. To further improve computational efficiency, we also propose to train the variational model on a significantly cheaper-to-evaluate lower bound, and show empirically that the resulting model provides an excellent guide for more accurate, but expensive to evaluate bounds on the EIG. We demonstrate the effectiveness of our technique on generalized linear models, a class of statistical models that is widely used in the analysis of controlled experiments. Experiments show that our method is able to greatly improve accuracy over existing approximation strategies, and achieve these results with far better sample efficiency.

4.1 Practical Considerations

We apply the same flexible mathematical framework for Bayesian optimal experimental design discussed in Section 3.3 and proposed by Foster et al. [2019]. However, Foster et al. [2019] adopted a "classical" variational distribution setting, in which the variational form q_ϕ is selected to take a standard, parametric form. They found this approach effective, but tested only on very simple design problems, with only one experimental unit at a time. Their variational models only incorporate the design implicitly, requiring a separate optimization for every design to be considered[1]. Unfortunately, as we show in the experiments this approach is not effective on more complex design problems. Instead, we propose a far more flexible, deep learning based distributional form that incorporates the design explicitly, allowing us to amortize training across and apply our trained model to evaluation of all (potentially continuously many) designs in our feasible set.

4.2 Method

We are interested in learning a parameterized function, $q_\phi(\theta|y, d)$, for approximating the posterior distribution. In this section, we describe our proposed deep learning architecture for amortizing over designs, which allows practitioners to train a single model that is capable of supporting estimates of the EIG over potentially infinitely many designs. We also discuss how we can efficiently train this model using the (simpler and cheaper) equation (3.10), then use the resulting approximation in the more accurate bounds provided by VNMC, (3.11)–(3.12), combining fast, efficient training with the accuracy and asymptotic guarantees of VNMC. This advances the work in Foster et al. [2019] by providing a highly flexible variational form that can be used in a wide variety of contexts and an inexpensive procedure to train it.

[1] Although subsequent work [Foster et al., 2020] considered evolving both the design and distribution q simultaneously, even that work remains focused on a single (if evolving) design.

4.2.1 Neural Architecture

Figure 4.1 shows a high level representation of our architecture. Broadly, it consists of two major components. The first is a learnable function for taking in the design variables d and simulated experimental outcomes from the model, y, and producing a design context vector, $c_{y,d}$, that will be used to define a conditional distribution. We focus on the common case where the experimental units lack any meaningful order and our learnable function must therefore be permutation invariant. We can incorporate this inductive bias into our model by making our function follow the general form of set functions proposed in Zaheer et al. [2017]. In the sequel, we denote this component as our *set invariant model*. The second major component is a learnable distribution conditioned on the design context produced by the set invariant model. In this work we use conditional normalizing flows, which consist of a base distribution and sequence of invertible transformations with tractable Jacobian determinants to maintain proper accounting of the probability density through the transformations. Both the base distribution and transformations are learnable and conditioned on the design context.

Set Invariant Model. It is often the case that the individual units being experimented on do not posses an inherent ordering – for example, subjects in a randomized controlled clinical trial, or the petri dishes used to grow our culture from the example in 2. Suppose we would like to find the optimally informative design for an experiment with S experimental units, where d_i and y_i denote the design variables and simulated outcomes of unit i, respectively. In this setting we want our design context to be invariant to permutations in its inputs, e.g., reordering the individuals in the trial should not change our results. Learning permutation invariant functions is an active area of research [e.g., Bloem-Reddy and Teh, 2020]. In this work we follow the general form proposed by Zaheer et al. [2017], where our set invariant

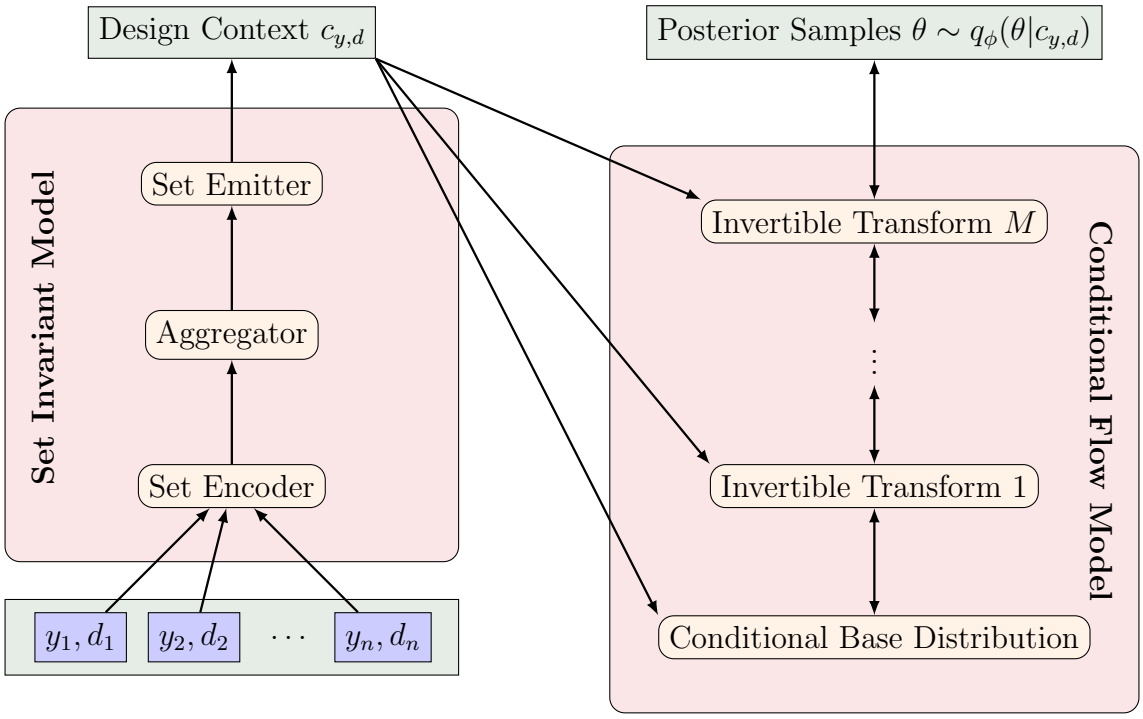

Figure 4.1: A high-level schematic of our architecture for amortizing over designs. The first component (left) takes in the design variables and simulated observations and produces a design context, $c_{y,d}$. In many experiments the individual units being experimented on are exchangeable, thus we use a set invariant architecture. The second (right) is a conditional normalizing flow, conditioned on the design context produced by the first component. Together, they define our variational posteriors $q_\phi(\theta|y, d)$, amortized over designs.

model is defined as,

$$c_{y,d} = \textsc{Emit}_{\phi_{\textsc{Emit}}}\left[\sum_{i=0}^{S} \textsc{Enc}_{\phi_{\textsc{Enc}}}(y_i, d_i)\right]. \tag{4.1}$$

In particular, we define two learnable functions. The set encoder $\textsc{Enc}_{\phi_{\textsc{Enc}}}(y_i, d_i)$ takes as input the design variables and simulated outcomes for each individual experimental unit. Its output is an intermediary representation for each experimental unit, which are aggregated together by summation; the permutation invariance of the sum ensures invariance of the overall function. The aggregated representation is then passed through the set emitter function $\textsc{Emit}_{\phi_{\textsc{Emit}}}(\cdot)$, which creates the final design context used in the conditional normalizing flow. In our experiments we find substantially improved performance using attention layers [Vaswani et al., 2017] in the set encoder, which allows for interactions between the

experimental units before aggregation. In this case we should denote our set encoder as $\textsc{Enc}_{\phi_{\textsc{Enc}}}(y_i, d_i | y_{-i}, d_{-i})$ where y_{-i} and d_{-i} denote all simulated outcomes and design variables except for the ith unit.

Structuring our design encoder function in this way gives two major advantages. First, permutation invariance does not need to be learned by the function since it is already present by construction; this can make learning more efficient and reduces the total number of weights via weight sharing. Second, the function is able to encode designs with a variable number of experimental units, S, as long as the d_i and y_i have the same size for all units.

Note that not all experimental design problems are permutation invariant in the experimental units. For example, in some settings there could be a temporal component in the design variables, in which case we could replace our set invariant function with an order-based model such as a recurrent neural network.

Conditional Normalizing Flow. Normalizing flows define an expressive class of learnable probability distributions, which have been used in generative modeling and probabilistic inference [Papamakarios et al., 2021, Kobyzev et al., 2020]. The main idea of normalizing flow based models is to represent a random variable θ as a transformation $\theta = T(x)$ of a random variable x sampled from a base distribution $p(x)$. The key property is that the transformation T must be invertible and differentiable. This allows us to obtain $p(\theta)$ via a change of variables,

$$p(\theta) = p(x) \; |\det J_T(x)|^{-1} \tag{4.2}$$

where $x = T^{-1}(\theta)$ and $\det J_T(x)$ is the determinant of the Jacobian at x. Both the transformations and base distribution may have learnable parameters. This provides a highly flexible class of distributions that can be both sampled and efficiently evaluated.

In our setting we would like to learn not just a single distribution, but rather a *conditional*

distribution given design variables and experimental outcomes. This conditioning on d is key to allowing us to amortize over all possible designs. To this end, we learn a sequence of K conditional transformations $T_{\phi_i}(\cdot|c_{y,d})$ and a conditional base distribution $p_{\phi_0}(x|c_{y,d})$ which together define our variational approximation to the posterior distribution amortized over designs,

$$q_\phi(\theta|c_{y,d}) = p_{\phi_0}(x_0|c_{y,d}) \prod_{i=1}^{K} |\det J_{T_{\phi_i}}(x_i|c_{y,d})|^{-1}, \tag{4.3}$$

where $\theta = T_{\phi_K} \circ T_{\phi_{K-1}} \circ \dots T_{\phi_1}(x_0)$, with all transformations and base distribution conditioned on the design context (4.1). The full architecture, including the set invariant model and conditional normalizing flow is trained end-to-end.

4.2.2 Variational Posterior Training

The posterior estimator and the (contrastive) VNMC bounds all require learning a variational approximation to the posterior distribution. In both Foster et al. [2019] and Foster et al. [2020] each bound was trained separately, learning its own variational approximation. However in all cases the variational approximation produced by training one of the bounds *can* be used for evaluating any other bound, since they all require only an approximate posterior distribution. Ideally, we would like to train using only the posterior estimator since it is much cheaper – a total cost of only $\mathcal{O}(N)$ – whereas both VNMC bounds have a total cost of $\mathcal{O}(NM)$. However, it is not obvious that training on the (also less accurate) bound should still provide good EIG estimates when used in the VNMC bounds. Our experiments show that it is surprisingly effective across a broad range of models. Intuitively speaking, this is possible because all bounds share the same optimum – the true posterior distribution. Moreover, because the posterior estimator takes its expectation with respect to the model $p(y, \theta|d)$, the variational approximation $q_\phi(\theta|y, d)$ will in general be *wider* than the true posterior distribution, akin to variational inference using the "forward" Kullback-Liebler (KL) divergence, and in contrast to the more commonly used "reverse KL" variational op-

timization methods that result in underdispersed and mode-seeking optima. This property also makes the posterior estimator's q_ϕ an excellent choice for importance sampling (as in VNMC), in which a too-narrow proposal distribution can lead to high variance in the importance weights, causing a small number of samples to dominate the estimator [Owen, 2013, Chapter. 9].

4.3 Related Work

Our approach builds on the framework for BOED developed in Foster et al. [2019], which proposed four variational bounds for estimating EIG. The framework itself is quite flexible, capable of accommodating a wide variety of models (e.g., implicit vs explicit likelihoods), sequential experimentation (of arbitrary batch size) and arbitrary variational forms q. However, their experiments were limited to only single experimental units, and used simple variational forms that cannot amortize over designs (requiring a separate training procedure for each proposed design). In this work we propose a deep learning architecture which can easily be scaled to approximate arbitrarily complex distributions. In addition, our architecture can amortize over designs, allowing us to train a single variational model capable of estimating the EIG for potentially infinitely or continuously many designs. We also show that we can train our model using the cheaper posterior bound, then use its optimized approximate posterior within the VNMC bounds for a more accurate final approximation. We show that, using our proposed variational form, we can achieve highly accurate EIG estimates across a spectrum of complex design problems. While a few other EIG approximations have been proposed (see, e.g., Foster et al. [2019], Ryan et al. [2016]), in light of the experimental results of Foster et al. [2019] we mainly compare our experimental performance relative to NMC.

4.4 Experiments

We perform three types of experiments: *amortization*, *model* and *architecture* experiments. Our *amortization* experiment shows the dramatic increase in efficiency from amortization, and better EIG estimation provided by our more complex variational forms compared to those used in Foster et al. [2019]. *Model* experiments examine how the benchmark method, NMC, breaks down as model complexity grows while our methods remain reliable for accurately estimating the EIG. *Architecture* experiments measure the impact of key componets in our variational approximation and serve as a guide to using our method effectively.

In all experiments we focus on estimating designs for different types of generalized linear models (GLMs) [McCullagh and Nelder, 1989]. GLMs are a very common model class used to analyze controlled experiments and are regularly used in applications of optimal experimental design [Goos and Jones, 2011]. Our GLMs have the general pattern,

$$\begin{aligned} \theta &\sim \mathcal{N}(\mu_p, \Sigma_p) \\ r &= g^{-1}(D\theta) \\ y &\sim \text{Exponential Family Distribution}(r). \end{aligned} \tag{4.4}$$

Here, θ is a $N_p + 1$ dimensional parameter vector, where N_p is the number of predictors ($+1$ for the intercept term). D is a $N_E \times (N_p + 1)$ design matrix, where N_E is the number of experimental units. The inverse link function, g^{-1}, defines the type of GLM. Finally, μ_p and Σ_p are the prior mean and covariance of the parameters. Our experiments cover six GLMs: normal (known observation noise), normal unknown (unknown observation noise), logistic, binomial, categorical and multinomial. For the normal model with known observation noise we take $\sigma = 1$; for the normal model with unknown observation noise[2] we use the prior $\sigma \sim \text{InverseGamma}(a_p, b_p)$ with $a_p = b_p = 3.5$. For the binomial model, we assume 10

[2]In this case, the observation noise is included as the standard deviation in the normal distribution that samples y.

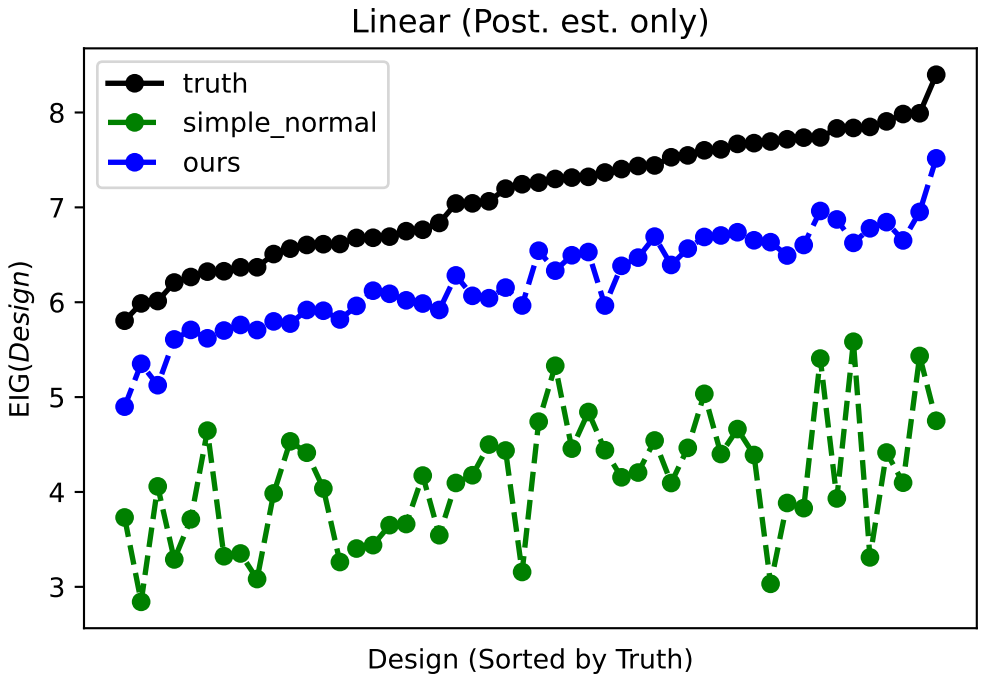

Figure 4.2: Comparing our method and the (non-amortized) variational form used by Foster et al. [2019] on the normal (linear) model with 5 predictors and 5 on the diagonal of the prior covariance (similar to the AB model in Foster et al. [2019]). For clarity we only show the posterior estimator values. Our method is $> 3\times$ faster (293s vs. 920s) and significantly more accurate. See text for further analysis.

random trials; we use 3 classes for the categorical model, and 10 trials and 3 classes for the multinomial model. All experiments in the sequel are run for designs with 5 experimental units. Our implementations made significant use of Pyro [Bingham et al., 2019] to implement the inference procedures and NFlows [Durkan et al., 2020] to construct our conditional normalizing flows. All training was done on a single Nvidia 2080TI and evaluation was done on an Intel I7-9800X with 64 GB of RAM.

4.4.1 Amortization Experiments

While providing an excellent theoretical framework, from a practical point of view the variational forms used in Foster et al. [2019] are too simple to be effective on the GLM models

we consider. Additionally, their work required training a new variational model for every design being approximated, while we propose a method that can amortize over designs. Our closest model to those tested in Foster et al. [2019] is our normal (linear) model, similar to their "AB model"; we can apply their variational form for the AB model in order to perform a comparison. The variational form is:

$$q_\phi(\theta|y,d) = \mathcal{N}(Ay, \Sigma), \tag{4.5}$$

where the variational parameters are $\phi = \{A, \Sigma\}$ with A being a $N_E \times (N_p + 1)$ matrix and Σ a $(N_p + 1) \times (N_p + 1)$ positive definite matrix. As we can see this form incorporates the design d only implicitly, and so cannot amortize over designs. Recall that N_E is the number of experimental units and N_p is the number of predictors (adding 1 for the intercept). We train using the AdamW optimizer for 5000 steps with a learning rate of .001 with $\beta_0 = .9$ and $\beta_1 = .999$.

We set $N_p = 5$ and $\Sigma_p = 5I$, i.e., diagonal with variance 5. We generate 50 random designs with $N_E = 5$ experimental units and compare the quality of EIG approximations given by the posterior estimator as well as total wall clock time. The precise architecture and training procedure we use are described in Section 4.4.4. Figure 4.2 shows the results of this experiment: our method produces a much tighter lower bound that is highly correlated with the true values; selecting the highest estimate would pick the design with highest true EIG in the set. In contrast, the variational form used in Foster et al. [2019] yields a much looser and less correlated bound, which would select the 6th best design if used. Moreover, our method is more than 3× faster (293 seconds compared to 920 seconds), showing the benefit of amortizing over designs. In fact, this speed-up understates how much more computationally efficient our method is, given that it leaves us with a model that can estimate the EIG for arbitrarily many designs without additional training. Training took 291 of our method's 293 seconds; evaluating an additional 50 designs, then, would take virtually the same amount

of time, compared to double the time required for the non-amortized approach. The non-amortized training is prohibitively slow, and moreover, for our other GLM models it is often not clear what variational form from Foster et al. [2019] could be applied; for these reasons, in the rest of the experiments we compare only to standard NMC.

4.4.2 Model Experiments

In our model experiments we vary the GLMs in two ways: the number of predictors (not including the intercept) and the diagonal components of the prior covariance (all off-diagonal terms are zero). The number of predictors N_p is varied from 1 to 5 and the diagonal of the prior covariance varied in $\{1, 5, 25\}$. For the neural network architecture we use attention layers for the set encoder, a residual network for the set emitter [He et al., 2016], a full rank Gaussian distribution for the conditional base of the normalzing flow and four affine coupling layers each parameterized with a residual network. The precise architecture and training procedure we use are described in Section 4.4.4. In all experiments we train the variational approximation using the posterior estimator. During training, new designs are generated randomly from a multivariate normal distribution with identity covariance in dimension $N_p + 1$. For final evaluation we generate 50 new random designs and estimate the posterior bound with $N = 5000$ samples, while the VNMC bounds are estimated with $N = 1000$ and $M = 31$ samples and nested samples, and the NMC bounds are estimated with $N = 30000$ and $M = 173$ samples and nested samples. The number of samples for VNMC and NMC were selected based on the maximum number of samples that fit into memory (64 GB RAM) for the largest model (multinomial with 5 predictors).

Figure 4.3 shows EIG evaluations for 50 randomly generated designs for the 5 estimators: posterior, VNMC upper, VNMC lower (contrastive VNMC), NMC upper and NMC lower (contrastive NMC). Since this figure pertains to the linear model we can calculate the ground truth EIG exactly, shown in solid black. For visual clarity we sort the designs in order of

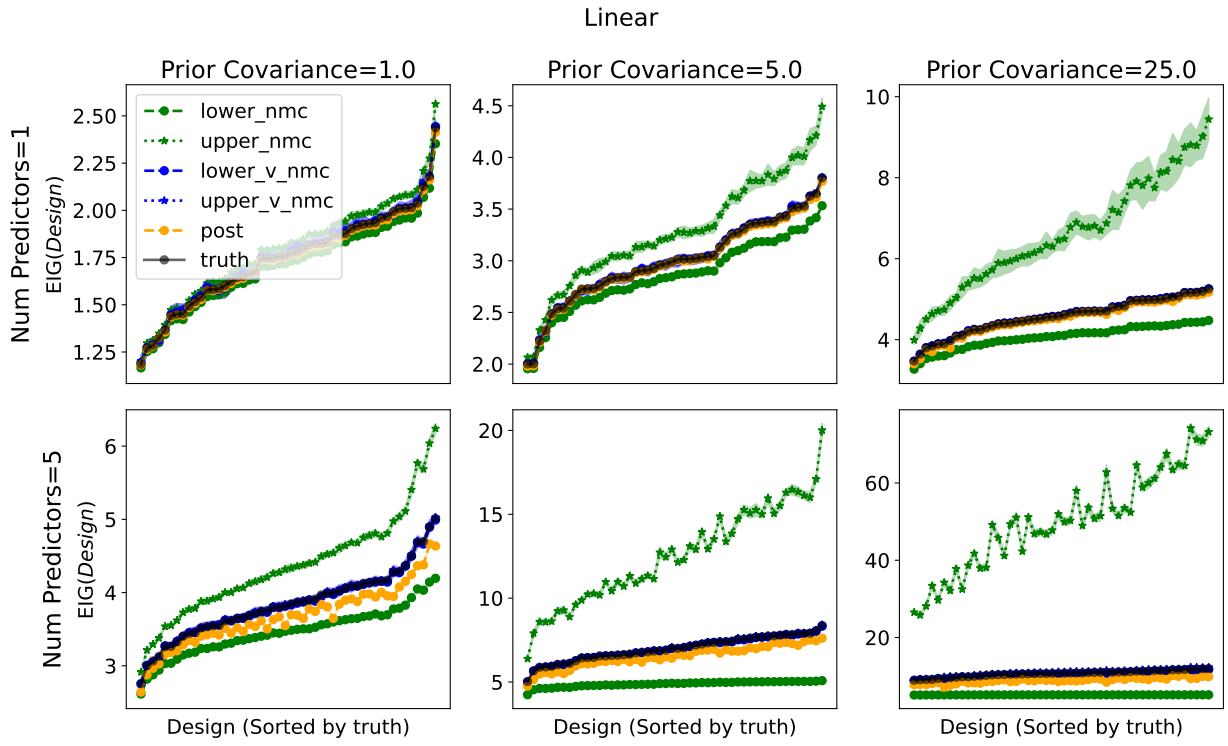

Figure 4.3: Results for estimating EIG in the linear model with known observation noise. The x-axis ranges over the index of 50 randomly selected designs, each with 5 experimental units. Due to the dimensionality, the designs lack a meaningful order; for visual clarity we plot them in the sorted order of the true EIG. The rows vary the number of predictors ($N_p \in \{1, 5\}$) while columns show changes in the diagonal of the prior covariance matrix, $\{1, 5, 25\}$, from informative to uninformative. NMC and VNMC methods can estimate both upper and lower bounds, while the posterior estimator only provides a lower bound. Our proposed method gives much tighter bounds on the truth than the competing NMC (with 167× fewer samples). The shading shows one standard deviation of our estimates over 20 runs.

ground truth EIG value. We see that all estimators perform reasonably well on the easiest form of the model (1 predictor with unit prior covariance). However, even in this case the VNMC bounds (upper and lower) more tightly constrain the ground truth – in fact both are nearly exact. In addition the posterior bound (a lower bound) is consistently above the NMC lower bound and closer to the truth. These trends become magnified as the prior covariance and number of predictors increase. In all cases the VNMC bounds are nearly exact, while the performance of NMC degrades rapidly with problem difficulty. Again, the

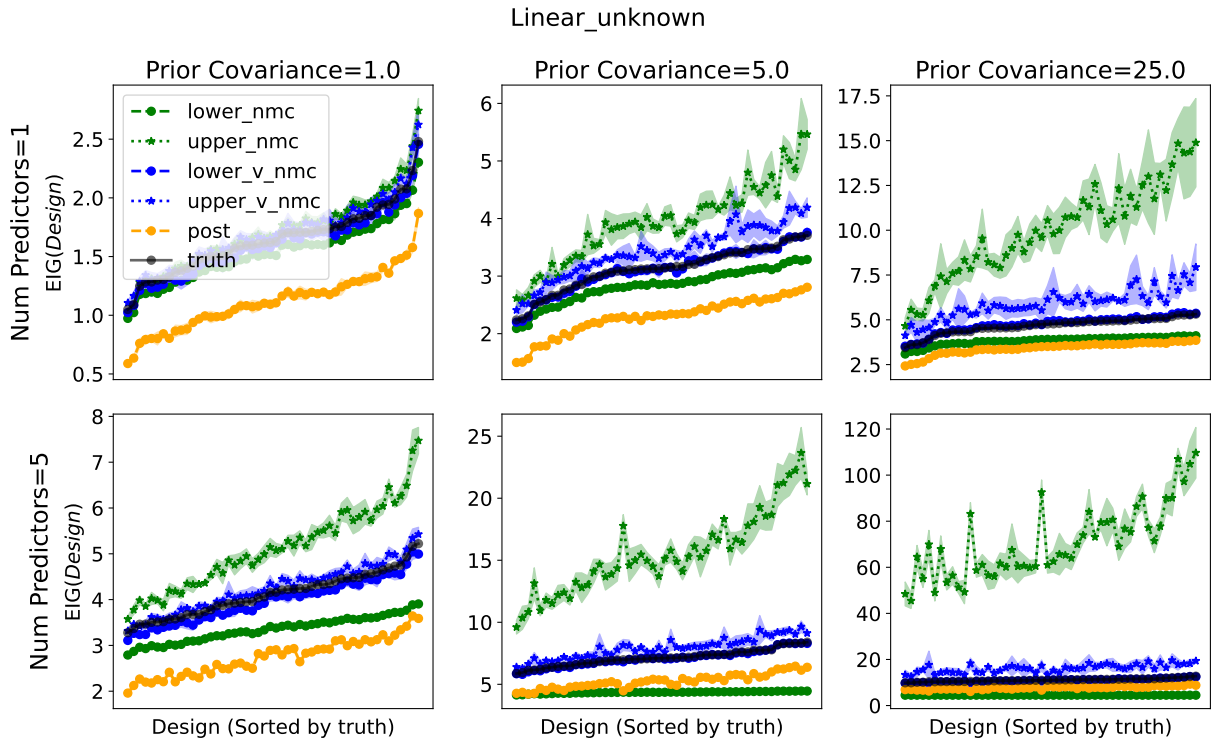

Figure 4.4: Same as Figure 4.3, but for the linear model with unknown observation noise.

posterior estimator remains above and closer to truth than the NMC lower bound.

Figure 4.4 shows exactly the same set of experiments, but for the linear unknown model. In this case we can calculate a high-quality Monte Carlo estimate of the ground truth thanks to conjugacy, and sort the designs to be evaluated in order of this ground truth. The results are largely consistent with those from the linear model with known observation noise: the VNMC bounds constrain the ground truth much more tightly than the NMC bounds. However in this case the posterior estimator is only above the NMC lower bound in the two hardest cases (5 predictors and 5 or 25 diagonal covariance).

Figures 4.5, 4.6, 4.7 and 4.8 show the same set of experiments but for the logistic, binomial, categorical and multinomial models. In none of these cases can we calculate ground truth, so all plots order their designs by the benchmark NMC (upper). Even without ground truth we still clearly see the lower and upper bounds of VNMC are much closer together and

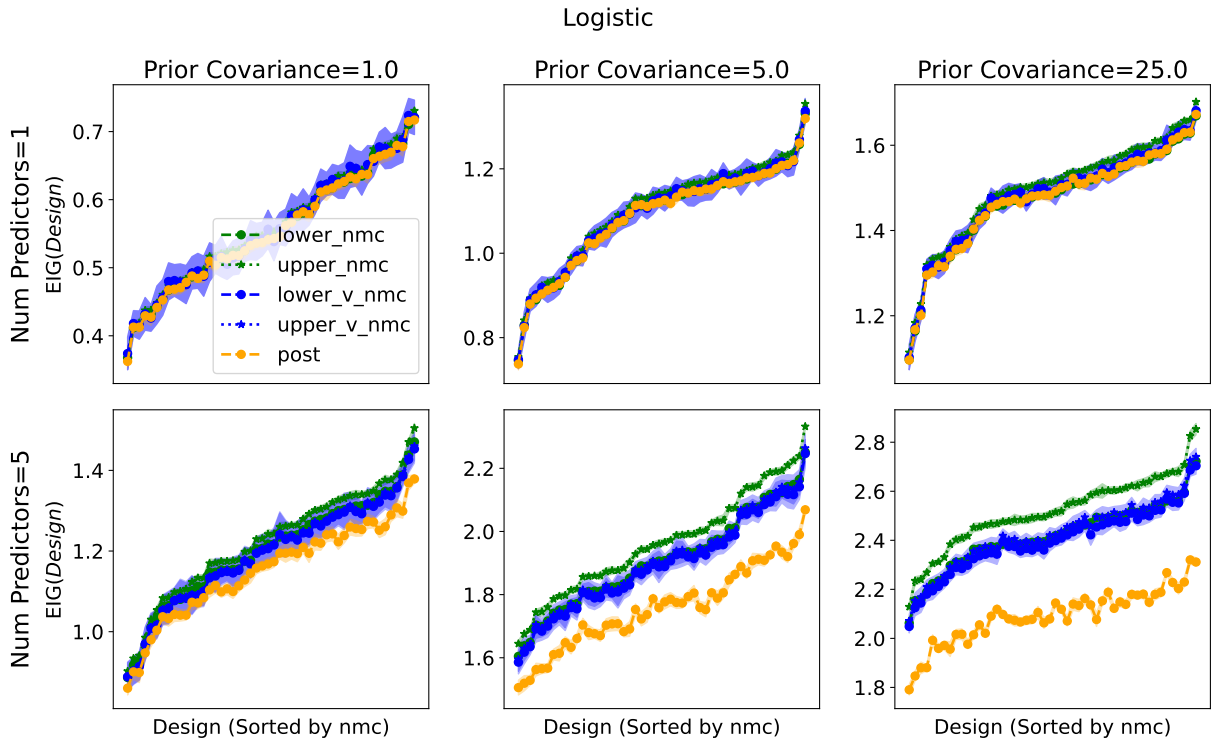

Figure 4.5: Same as figure 4.3, but for the logistic model. For this model we cannot compute ground truth, but we can still infer superior performance for our methods by observing how VNMC produces tighter bounds (both lower and upper) compared to NMC. Designs are ordered on the x-axis via the baseline NMC.

below/above their NMC counterparts in practically all cases. One exception is the logistic model, where as long as the model only contains one predictor variable, the NMC bounds are as tight as the VNMC bounds; however by 5 predictors the VNMC bounds are tighter. In fact the VNMC lower and upper bounds closely agree with each other in all cases, suggesting they are also closely estimating the true EIG. We also see that the VNMC lower and upper bounds are touching across all settings of the binomial model and all but the hardest in the categorical and multinomial models (where they are still much tighter than NMC), again suggesting that our VNMC estimators are nearly exact. The results for the posterior estimator are more mixed – sometimes it is above (and presumably more accurate than) the NMC lower bound, while sometimes it is below (and presumably worse). It is worth noting, however, that since the VNMC estimates shown here use a variational distribution q

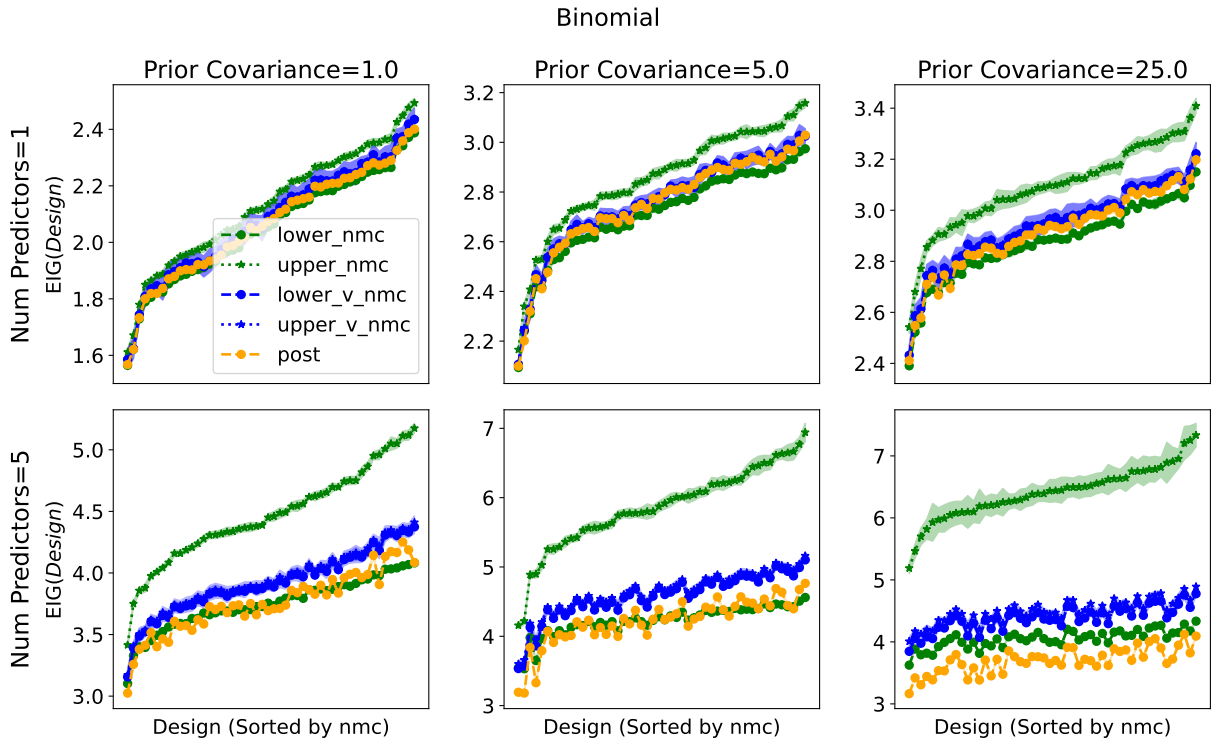

Figure 4.6: Same as figure 4.3, but for the binomial model. For this model we cannot compute ground truth, but we can still infer superior performance for our methods by looking at how VNMC produces much tighter bounds, both lower and upper.

trained on the posterior estimator objective, the posterior estimator can be effective in this role despite its uneven final estimate quality.

Our experiments show that training a *single* variational posterior, amortizing over designs, we can calculate the EIG much more accurately than the competing NMC benchmark, nearly calculating ground truth exactly. Moreover, the experiments show that training on the posterior estimator can provide a variational distribution that remains effective for estimation using the more costly VNMC bounds (see Section 4.2.2). Not only does VNMC provide far more accurate estimates, it does so with many fewer samples – 1000×31, compared to 30000×173 for NMC, i.e., more than two order of magnitudes ($167\times$) fewer samples than NMC.

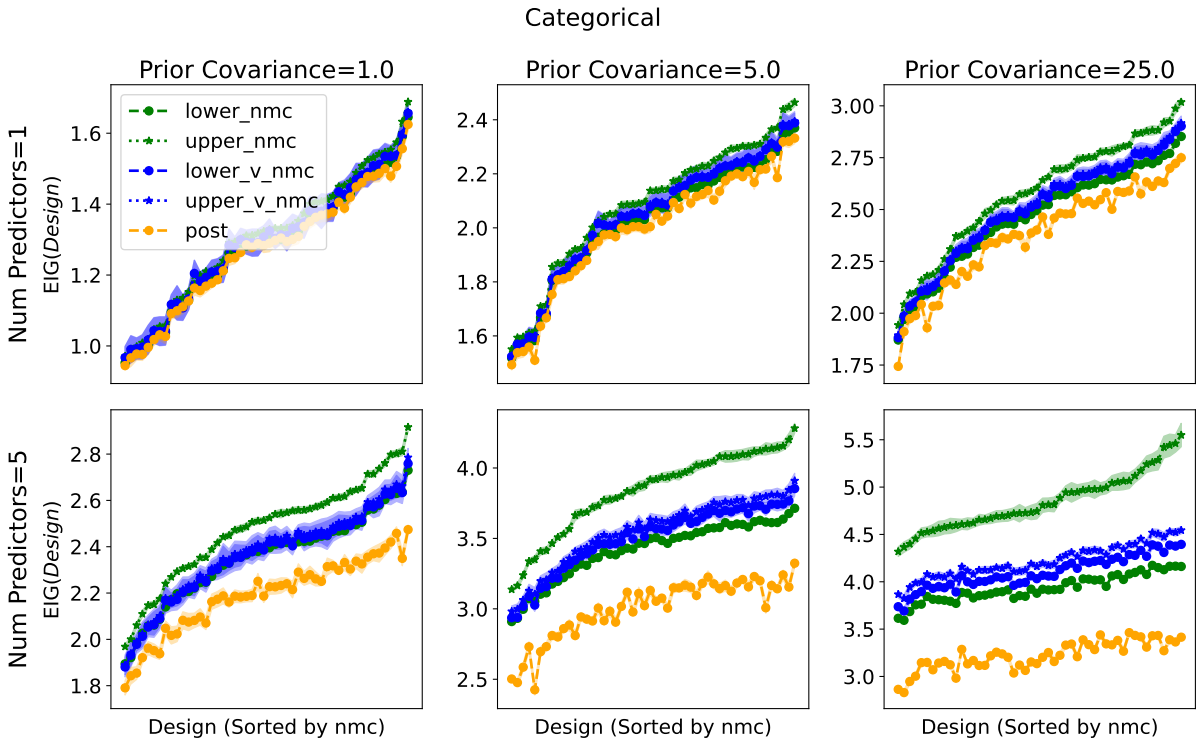

Figure 4.7: Same as figure 4.3, but for the categorical model. For this model we cannot compute ground truth, but we can still infer superior performance for our methods by looking at how VNMC produces much tighter bounds, both lower and upper.

4.4.3 Architecture Experiments

We next investigate the importance of architectural decisions for the neural networks defining $q_\phi(\theta|y, d)$. We compare using attention layers vs. residual layers in the set encoder, and the effect of the transform type and number in the normalizing flow. We compare using 4 vs. 8 transforms, and test affine coupling transforms [Dinh et al., 2017], rational quadratic (RQ) splines [Durkan et al., 2019], and no transform (just the conditional base distribution). We run all combinations for the linear unknown and binomial model with 5 predictors and 25 on the diagonal of the prior covariance. Note that the true posterior for the linear unknown model is t-distributed, while the binomial is not analytically expressible, so we expect the use of normalizing flows to be advantageous over just the normal base distribution. The rest of the architecture components are the same as the Model Experiments and full details can

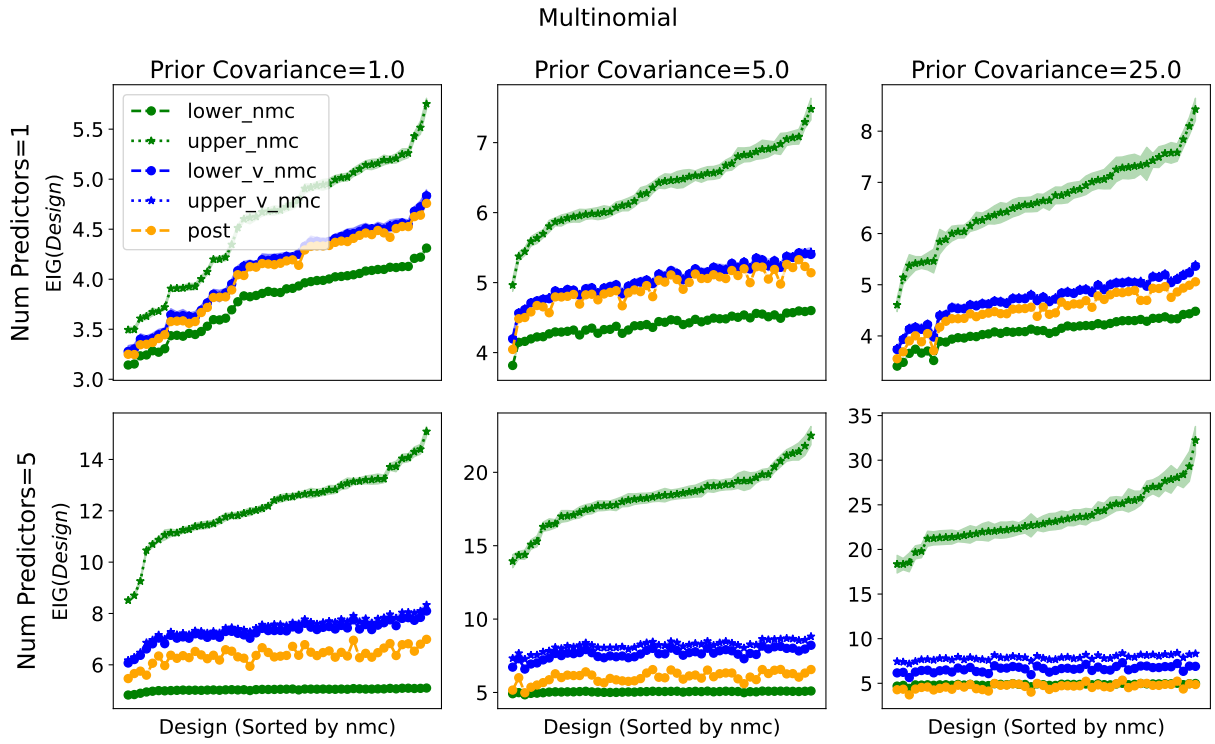

Figure 4.8: Same as figure 4.3, but for the multinomial model. For this model we cannot compute ground truth, but we can still infer superior performance for our methods by looking at how VNMC produces much tighter bounds, both lower and upper.

be found in section 4.4.4.

Figure 4.9 shows the loss curves of the experiments for the linear unknown model. The inset plot on the top right shows the loss curves across all epochs, while the main plot shows a detail of the last 50 optimization steps. Each optimization step is run on a batch of 50 designs, so this plot indicates final performance on 2500 randomly generated designs. Specifically the loss is $-\sum_{i=1}^{N} \log q_\phi(\theta_i|y_i, d)$ where $y_i, \theta_i \sim p(y, \theta|d)$ and $N = 50$ – the cross entropy of the variational posterior. Empirically, we see that using attention layers in the set encoder is the most important architectural decision (lower 5 curves vs. upper); all networks using attention layers achieved superior performance to all networks using ResNets regardless of the other architectural settings. Beyond this, we see that using an affine coupling layer is also important, but see little difference between 4 and 8 transform layers. Surprisingly, the RQ

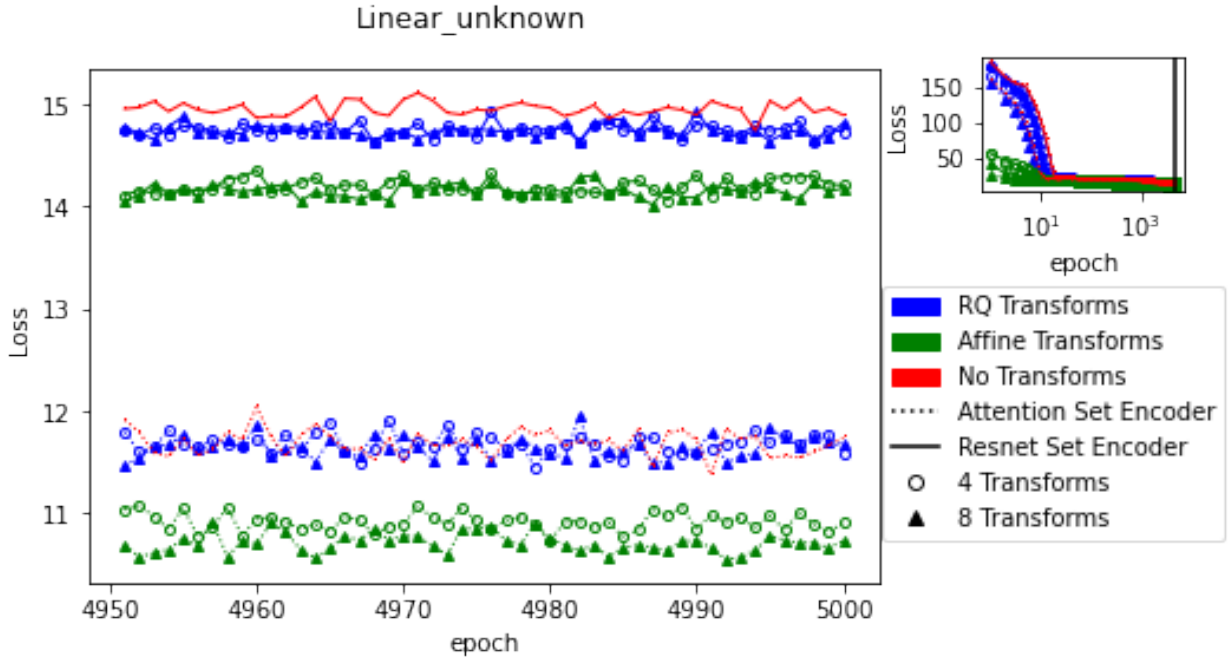

Figure 4.9: Results for our architecture experiments on the linear unknown model with 5 predictors and diagonal prior covariance 25. We vary the architecture of the set encoder (attention vs. resnet), the normalizing flow transform type (affine coupling, affine spline, or no transform), and the number of transforms (4 or 8). The main plot shows the loss for the posterior estimator over the last 50 steps of training; each step is performed over a batch of 50 random designs (2500 designs total). The inset plot shows the loss curves over all 5000 training steps, indicating all architectures have converged. Further discussion is given in the text.

transforms perform no better than having no transform. This is because the RQ transforms are restricted to the range of $(0, 1)$, with linear tails outside. Even after training, almost all parameter samples are outside this range by the time they reach the spline; the linear tails effectively skip the flow layers completely, explaining why its performance is comparable to models with no transform layers. Figure 4.10 shows the performance of a subset of these models at evaluating the EIG for 50 random designs, highlighting that the loss curves' values are directly related to the accuracy and sample efficiency of the EIG estimate. Figures 4.11 and 4.12 show the same plots for the binomial model and further support these conclusions.

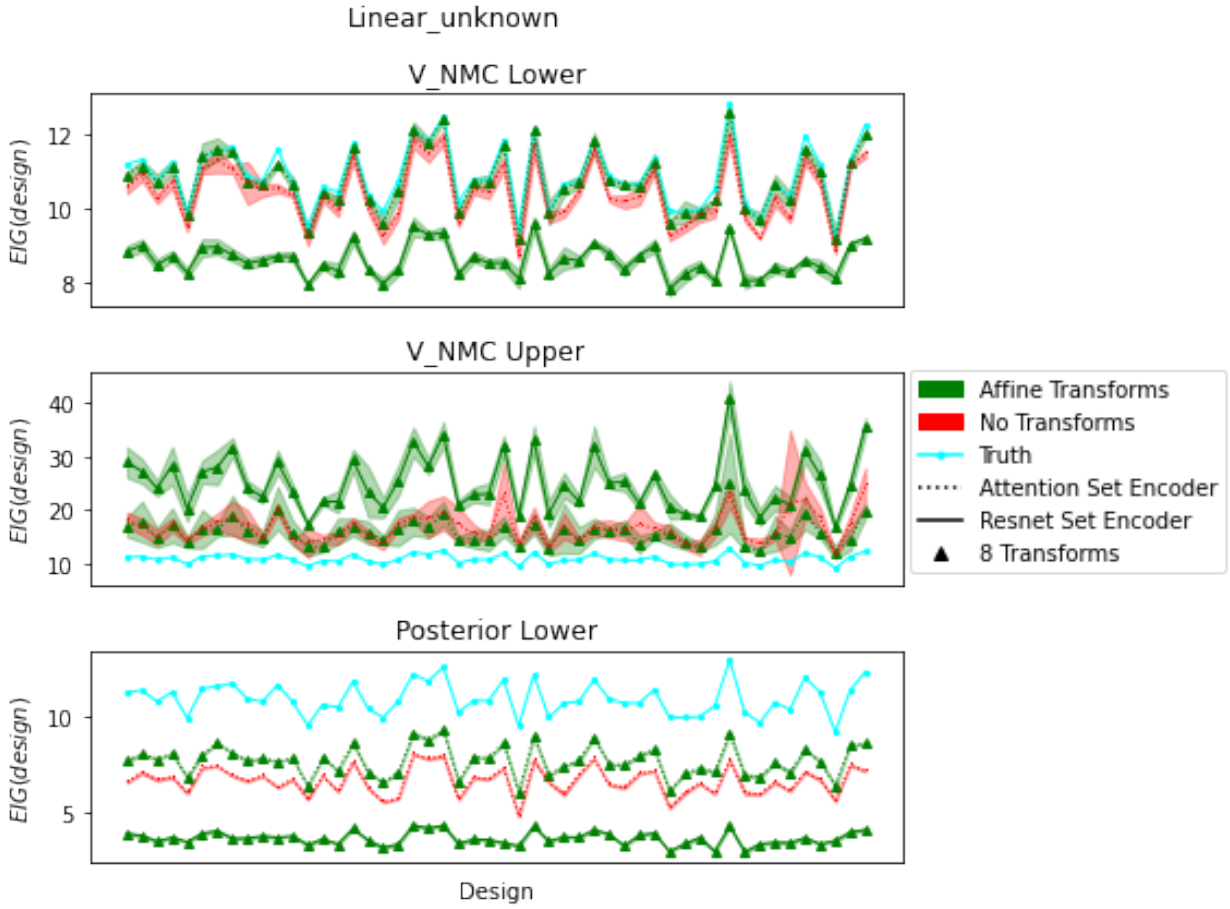

Figure 4.10: Here we show that the loss values in figure 4.9 provide meaningful difference in the quality of EIG approximation. For a subset of the architectures trained we estimate the two VNMC based and the posterior bound using the same number of samples. We can see that the architectures that achieved lower loss values during training achieve greater estimation accuracy given the number of samples.

4.4.4 Full Architecture Details

For the design amortization experiments and the model experiments in sections 4.4.1 and 4.4.2 we used a common neural network architecture across all models types, which we specify in detail for reproducibility. Note that, between all neural network layers described in the sequel is a ReLU activation function.

For a given experimental design matrix D of size $N_E \times (N_p + 1)$ we simulate from the model the expected outcomes y, a vector of size N_E. We concatenate these together to construct

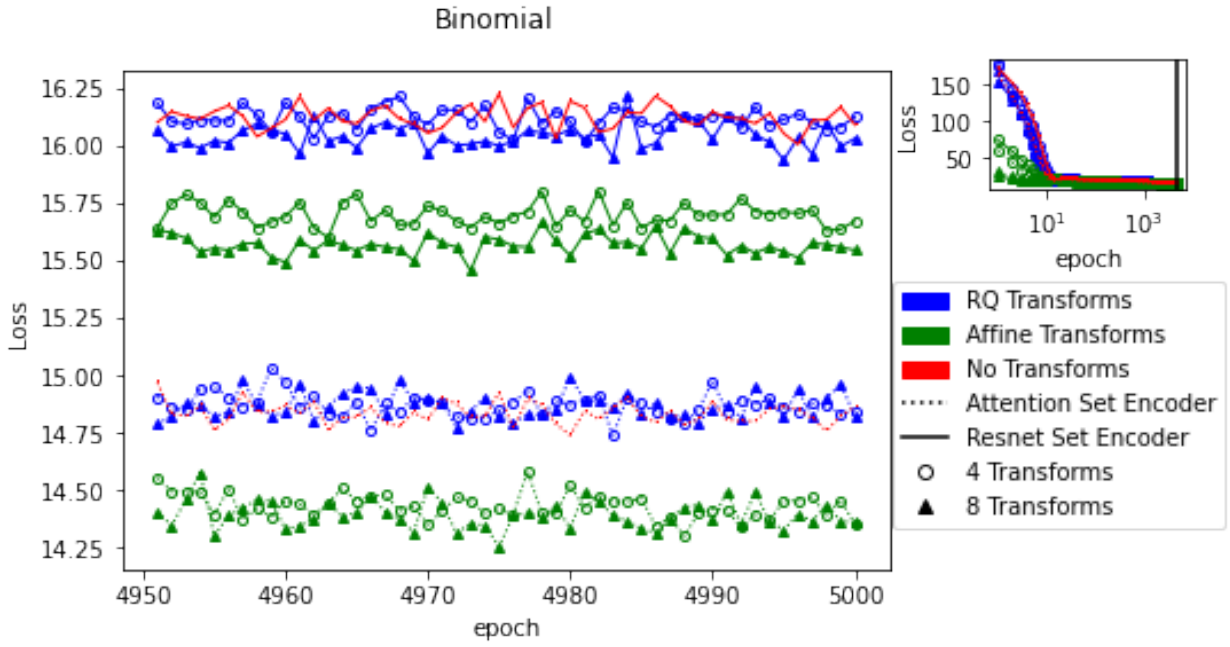

Figure 4.11: Same as figure 4.9, but for the binomial model.

an input matrix, C, of size $N_E \times (N_p + 1 + 1)$ to our Set Invariant Model. Each row of this input matrix is then passed through an embedding network, which is a residual network with two residual blocks of dimension 64 (we define a residual block as two linear layers where the input to the first is added to the output of the second), creating an internal representation R of size $N_E \times 120$. We now pass R through two attention layers with 12 heads (head dimension is 10). Each attention layer is followed by a dropout layer, then a linear projection with 32 dimensions and another dropout layer; each dropout layer has a dropout probability 0.1. This completes the set encoder, creating an internal representation for each experimental unit. These representations are then passed through the permutation invariant aggregator function, for which we use summation.

Aggregation produces a single vector, regardless of the number of experimental units. This representation is then passed through the Emitter Network of the Set Invariant Model. The Emitter Network is simply a residual network with two residual blocks each with linear layers of dimension 128. This completes the Set Invariant Model which creates the design context

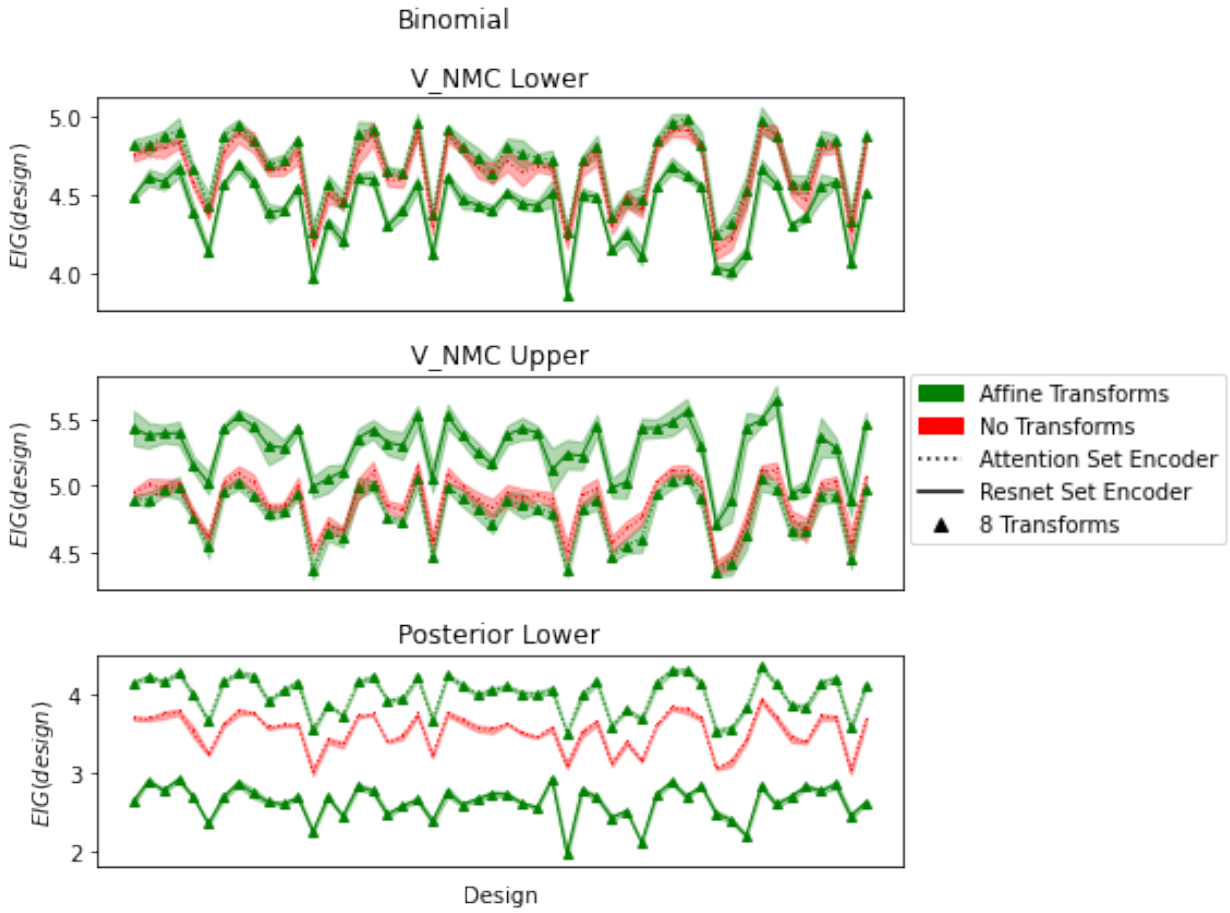

Figure 4.12: Same as in figure 4.10, but for the binomial model.

$c_{y,d}$.

We next provide the details of our conditional normalizing flow, following the sampling direction from the base distribution to the transforms. For the base distribution, we use a full rank multivariate normal distribution conditioned on the design context $c_{y,d}$. This distribution is parameterized by a residual network with two blocks and linear dimension of size 64. The last layer produces the mean vector of the normal distribution, and the entries of a lower triangular matrix that represents the Cholesky decomposition of the the covariance matrix. This lower triangular matrix is then left-multiplied with its transpose to creating the covariance of the base distribution. We can sample from this distribution straightforwardly.

The samples are then passed through four conditional affine coupling layers (unless stated otherwise in the chapter). Each affine coupling layer contains a residual neural network consisting of two residual blocks with linear layers of dimension 128. The residual network takes as inputs the samples produced from the base distribution, concatenated with the context $c_{y,d}$. The network outputs the parameters for an affine transformation that is applied to the samples, which are then passed through a random permutation before the next affine coupling layer. The final affine coupling layer outputs the samples θ from our variational model $q_\phi(\theta|y, d)$.

The forward procedure produces samples from our variational model, $q(\theta|y, d)$. In order to evaluate $q(\theta|y, d)$ on samples simulated from the model, $p(y, \theta|d)$, we simply pass y and d through the set invariant model to compute the design context, $c_{y,d}$, and then pass the simulated samples θ back through the inverse of the affine coupling layers to the base distribution, which can be evaluated. This now defines both directions of our variational model.

This completes the architecture details used in the Model Experiments. In the architecture experiments, we use the same architectural skeleton, but alter it as necessary. Specifically, we test using a ResNet within the Set Encoder instead of the attention layers; this ResNet consists of four residual blocks with linear layers of dimension 64. We also test using rational quadratic coupling splines in place of the affine coupling layers; these spline transformations also use ResNets with two residual blocks of 128 dimensions for the linear layers which output the parameters of an RQ spline with 20 buckets and linear tails.

All models are trained with the AdamW optimizer [Loshchilov and Hutter, 2019] with a learning rate of 5×10^{-4} and $\beta_0 = 0.9$ and $\beta_1 = 0.999$ for 5000 steps, where each step consists of a batch of 50 randomly generated designs. During training we use the posterior estimator to define the loss, and set $N = 50$ for the Monte Carlo estimator. We found no meaningful variation over the random seeds.

4.5 Conclusion

In this chapter we expand on the work of Foster et al. [2019], which proposed variational bounds for estimating the EIG for Bayesian optimal experimental design. In particular we propose a deep learning architecture incorporating set invariance and conditional normalizing flows that allows us to train a single model capable of estimating the EIG across the design space. Our experiments show that this architecture is highly effective at estimating EIG, and that design amortization provides significant computational speed ups. For cases where ground truth can be calculated, our model's VNMC bounds are nearly exact, while in cases without ground truth our VNMC upper and lower bounds are often sufficiently tight to suggest they are exact. These estimates are significantly more accurate than those of standard NMC while requiring far ($167\times$) fewer samples, as well as far more accurate and efficient than the simpler, non-amortized variational forms used in Foster et al. [2019]. We also demonstrate that we can train our model using the much cheaper posterior estimator bound, with cost $\mathcal{O}(N)$, then evaluate using this fitted model within the more accurate but costly VNMC bounds, $\mathcal{O}(NM)$. Together, we provide a method for faster and more accurate approximation of the EIG across many possible designs. In the next chapter we extend our approach to design optimization tasks.

Chapter 5

Amortized Optimization for Variational BOED

5.1 Introduction

In this chapter we focus on the problem of optimizing the EIG with respect to the design variables. In any application of OED the experimenter is confronted with constraints that determine the space of feasible designs, D, that can be run. The constraints may pertain to the values of the design variables, d, that are allowed, the total number of experimental units that can be used, the total cost of the experiment and many other constraints which reflect the practical hurdles faced when operating an experiment. We seek to find the design that optimizes the EIG over the feasible region. In this chapter, We show how the variational form introduced in the preceding chapter facilitates the use of a wide variety of optimization algorithms, making it possible for experimenters to apply and flexibly adapt variational approaches to BOED to suit their specific needs. In particular we extend and improve the state of the art optimization methods from Foster et al. [2020], and show how design amortization can greatly improve the efficacy of "black box" optimization methods, which may have significant benefits to experimenters in certain settings.

We start in section 5.2 with a broad discussion of optimization algorithms, which we divide into two broad categories. "Two stage" optimization algorithms first train a variational model, which is then fixed and used to compute the evaluation objective in "black box" methods such as the coordinate exchange algorithm and Bayesian optimization. Then, we discuss algorithms that simultaneously optimize the variational parameters, ϕ, and design variables, d, in particular a gradient based approach for optimizing both d and ϕ, and deep adaptive design (DAD), an approach introduced in Foster et al. [2021] which is analogous to applying reinforcement learning to BOED. DAD provides a competitive benchmark to test our other methods. In Section 5.3 we extend the methods of Foster et al. [2020] and provide experiments demonstrating our method's improvements over the start of the art. Finally, in Section 5.4 we show head to head experiments comparing the performance of all algorithms discussed.

5.2 Optimization Frameworks for BOED

In general, there are two high-level paradigms for searching for an optimal design d. We can perform a local optimization approach, in which we adapt d in very small ways, for example estimating the gradient of the EIG with respect to the design d and taking a step (*gradient-based* BOED). Or, we can propose more significant changes to d, altering one or more of the experimental units in a non-local manner. Examples include the classic *coordinate exchange* (CE) algorithm [Meyer and Nachtsheim, 1995b], and modern *Bayesian optimization* (BO) [Frazier, 2018] techniques that use the history of evaluated designs to propose a new design to evaluate. We call these methods *two stage* optimization, since the design evaluation and optimization processes are effectively decoupled, iterating between the two steps of proposing new design values and then evaluating them. Within these two frameworks, Foster et al. [2020] suggests significant advantages for gradient-based optimization of designs. As

discussed in the previous chapter, variational techniques for estimating the EIG operate by training a variational distribution q to be used in the estimation process, and this training processes can be a significant part of the computational cost. Further as the design changes so too does the optimal choice of q; however, the small design updates of gradient methods allows the previous q to be re-used as initialization, effectively amortizing the training of q across the optimization path of the the designs. In contrast, the distant designs proposed by two-stage methods change q in significant ways. However, using our design-amortized model, we can share work across designs even in the two-stage setting, leading to significant improvements.

5.2.1 Two Stage Optimization Algorithms

Our two-stage algorithms generally follow the pattern of first training a variational posterior, $q_\phi(\theta|y, d)$ that can be used to approximate the EIG over the entire set of feasible designs. Subsequently this trained model can be used as the objective function in any applicable optimization algorithm. Here, we consider the coordinate exchange algorithm (CE) and Bayesian optimization (BO). Note that this framework can also support algorithms that use gradient information, but we do not explore them here.

Coordinate Exchange Algorithm: The CE algorithm is widely used across OED, and supported in many popular software libraries used in industry [Goos and Jones, 2011]. The CE algorithm is based on searching local neighborhoods, starting with a random initial design d, then iteratively traversing each component d_i and considering a predetermined set of values for that component, selecting the value that achieves the highest EIG. The algorithm terminates when an entire traversal is completed with no changes to the current design settings, ensuring that we converge to a local maximum of the EIG. CE is often run with multiple random starts. Although a relatively simple optimization method, the CE algorithm also has some advantages: for example, the CE algorithm is the only algorithm

we consider that works on discrete design variables with no adjustments. Although discrete designs are frequently encountered in real world problems (e.g., categorical data), prior work on variational methods for optimizing EIG has only considered continuous design variables [Foster et al., 2020, 2021, Ivanova et al., 2021]. For a more detailed discussion on the CE algorithm see [Meyer and Nachtsheim, 1995a, Goos and Jones, 2011, Chapter 2].

Bayesian Optimization: BO is a widely used class of optimization algorithms for objective functions that are expensive to evaluate and may be noisy [Brochu et al., 2010]. It builds a surrogate to the objective function, including a noise model capturing its confidence in the surrogate function in different regions of the design space; Gaussian processes [Rasmussen and Williams, 2005] are commonly used as surrogate functions. BO then defines an acquisition function that determines the next point in the design space to evaluate, and updates the surrogate based on this evaluation. Note that the acquisition function used by BO is different from the EIG, which is the objective we are trying to optimize. BO has been highly successful and works well with multi-objective and multi-fidelity optimization problems [Frazier, 2018]. This is of particular importance for BOED, as there are often multiple criteria that experimenters must consider when designing their experiments, and multi-fidelity techniques can be used to tune the number samples for stochastic estimates of the objective (such as our variational approaches) [Frazier, 2018]. BO techniques are also flexible, with the ability to incorporate gradient information or be adapted to discrete optimization problems [Wu et al., 2017, Luong et al., 2019]. Finally, there are many excellent software libraries for BO that can be easily used by experimenters; in this work we use Balandat et al. [2020]. Thus, BO would seem to be one of the most promising optimization algorithms for BOED. Although Foster et al. [2020] concludes that BO is outperformed by gradient methods and is not competitive, we show that by using design amortization, BO can be made as or more effective (both in computational expense and accuracy) than the gradient based algorithms discussed in the sequel.

5.2.2 Gradient Based Algorithms

Gradient BOED: We discuss gradient-based BOED methods in detail in the next section, including a generalization of the work in Foster et al. [2020] and experimental assessments. Broadly, this class of algorithms apply stochastic gradient methods that simultaneously optimize the variational parameters, ϕ, and design variables, d. Gradient-based BOED can be very effective for optimizing continuous designs; moreover, with design amortization we can optimize multiple starting designs using only a single variational model and optimization process, significantly improving efficiency. The main drawback of gradient-based BOED is that it is not applicable to discrete design variables, which can be common in practice. An interesting direction for future work could be to study the applicability of straight-through estimators and other relaxation techniques for discrete design variables.

Deep Adaptive Design: Proposed in Foster et al. [2021], deep adaptive design (DAD) takes a different approach to optimizing the EIG. Instead of learning a variational approximation to the posterior, $q_\phi(\theta|y, d)$, it trains a model that takes as input a history of design observation pairs, $(d_1, y_1), (d_2, y_2), \ldots, (d_t, y_t)$ and outputs the next design, d_{t+1} under which to run an experiment. It is trained on simulated trajectories of these histories, using bounds similar to those in Chapter 4 as the loss function. Although training time can be significant, especially for longer trajectories, once trained DAD is very efficient at suggesting future designs. Moreover, since it is trained on entire trajectories, it can learn to make non-myopic sequential design decisions, without requiring, e.g., backwards induction. Like gradient BOED, the main drawback of DAD is that it can only be run for continuous design variables. Additionally, it is not clear how to handle many complex constraints. For example, consider a budget constraint – although costs can be incorporated into the loss function, it is difficult to ensure that the deep learning model does not recommend designs that go over budget. In real experiments, the feasible region may be dictated by any number of practical constraints that could be hard to enforce in the network outputs. Such issues are far more studied in the

other algorithms (which have a much longer application history), but could make for interesting future work on DAD. Ivanova et al. [2021] also extended DAD for implicit likelihood models.

5.3 Generalizing Gradient Based BOED

Foster et al. [2020] proposed a stochastic gradient algorithm that simultaneously optimizes the variational parameters, ϕ, and design variables, d. In that work, the authors showed significant computational gains over two-stage approaches, placing gradient-based VBOED as a state-of-the-art technique. In this section we discuss how their stochastic gradient algorithm can be generalized when amortizing over designs. We first start by showing that our variational distribution from Chapter 4, which incorporates designs explitcly, is far more effective at estimating the EIG. We then analyze the impact of this change – specifically, once q depends on the design, the gradient update to d is Foster et al. [2020] is no longer correct, and we re-derive the correct gradients. This is followed by experiments that show the efficacy and advantages of our generalized approach. In our experiments, we use the same classes of GLM models evaluated in Chapter 4.

5.3.1 Evaluation for Fixed Designs

We first study the importance of including the design variables as input to the variational model when estimating the EIG for a single, fixed design. For each GLM type we vary the model in two ways: the number of predictor variables in the model (1 vs. 5) and the number of experimental units in the design (1 vs. 5). We then select a design at random, and train two different variational forms for estimating the EIG of the selected design: one that includes the design variables as inputs, $q_\phi(\theta|y,d)$ (labeled "Design Encoded") and a variational form that does not, $q_\phi(\theta|y)$ (labeled "Design Not Encoded"). Otherwise, the

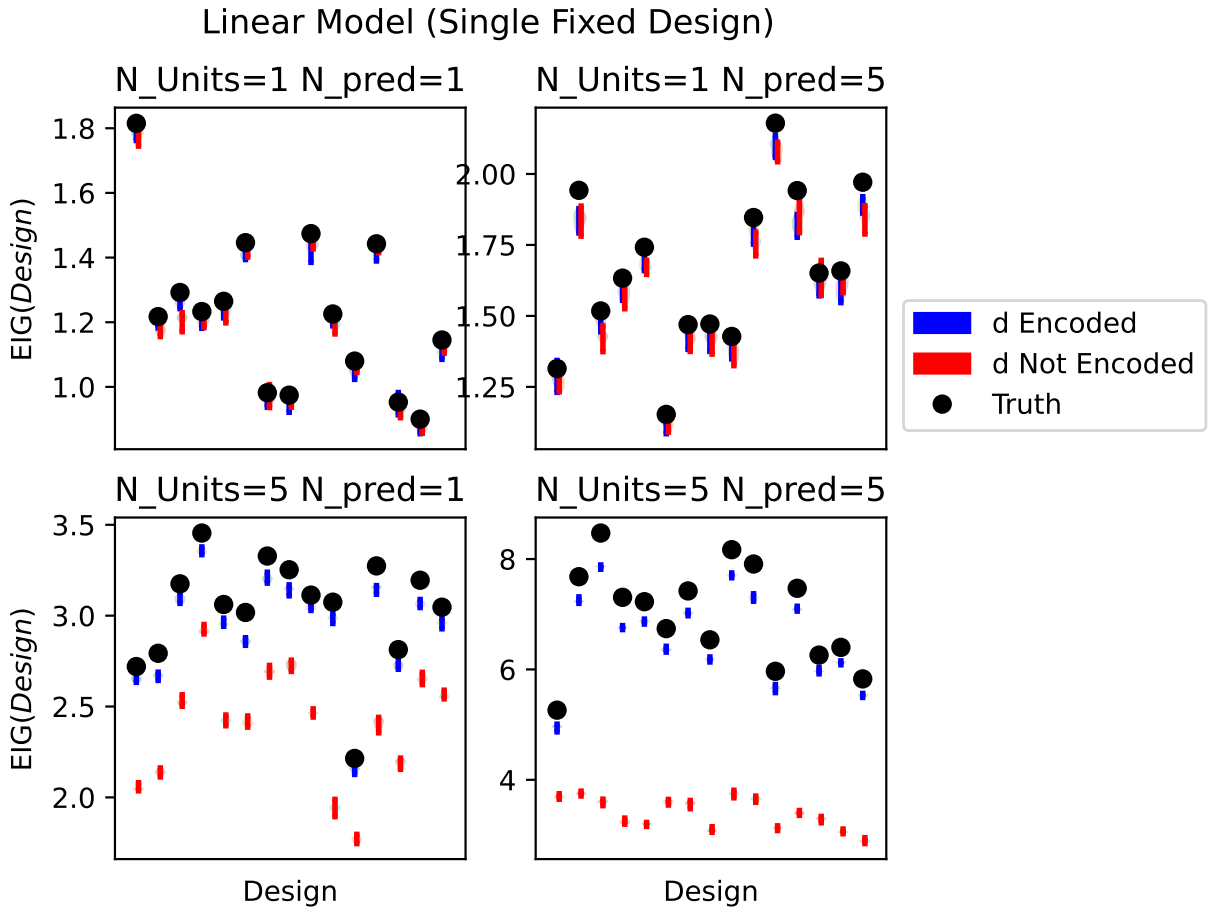

Figure 5.1: Effects of directly encoding design variables as inputs to our variational posterior on evaluation quality. We generate 15 random designs (x-axis) and train two variational models to estimate its EIG: one that encodes d directly, and the other without access to d. For each model, we plot the posterior estimator using 5000 samples. We can see that for designs with one experimental unit, encoding d has little effect, but for designs with 5 experimental units, encoding d results in substantially better accuracy. We also vary the number of predictors (columns) and find this has much less impact on estimation quality than the number of experimental units. Error bars reflect one standard deviation over 20 runs.

variational forms are the same and as described in Kennamer et al. [2022]. Figure 5.1 shows the posterior estimator for 15 randomly selected designs for each combination of number of predictors and number of experimental units. We clearly see that both variational forms are successful when the number of experimental units is one, but in the more complicated case of evaluating designs with five experimental units, the variational form that directly encodes

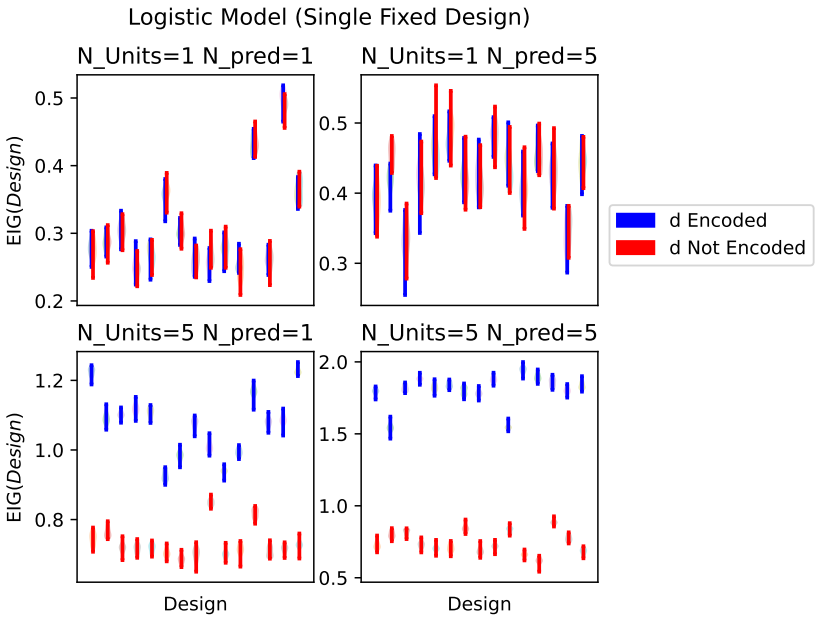

Figure 5.2: Same as figure 5.1, but for the logistic model.

the design variables performs substantially better than the form that does not. Figures 5.2, 5.3, 5.4, and 5.5 show the same experiment for the other GLM types. In the latter settings, we cannot compute ground truth but can use the fact that the posterior estimator is a lower bound to the EIG to gauge improvement. We again find that for a single experimental unit, both variational forms perform similarly, but once the experimental units are increased to five, the variational form that encodes the design performs substantially better across all GLM types.

In principle, both variational forms should be able to perform well in this case, since we are only trying to evaluate a single, fixed design. However, empirically we can see for complex designs it is much easier to achieve good performance by directly encoding the design variables. This suggests the importance of this modeling decision for any future work developing variational forms for BOED, and also helps explain why we see much better optimization performance from our generalized gradient algorithm compared to the gradient training proposed in Foster et al. [2020].

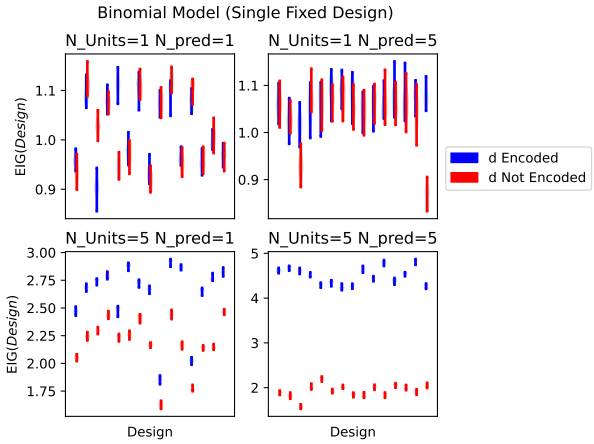

Figure 5.3: Same as figure 5.1, but for the binomial model.

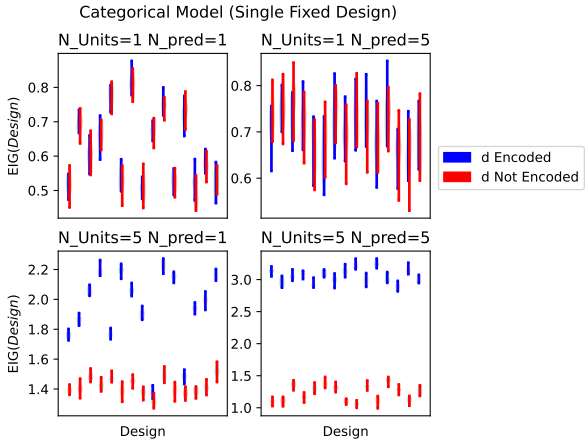

Figure 5.4: Same as figure 5.1, but for the categorical model.

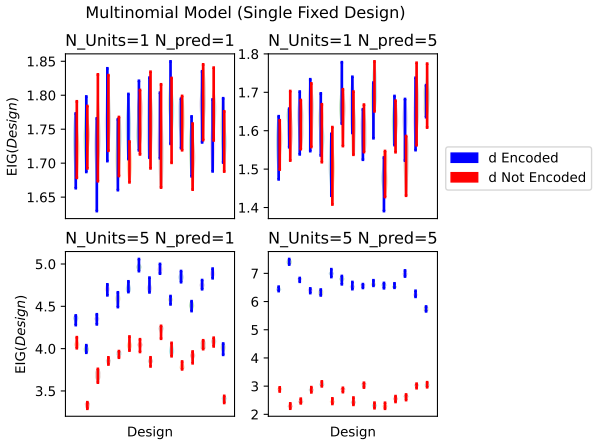

Figure 5.5: Same as figure 5.1, but for the multinomial model.

5.3.2 Unified Stochastic Gradients

For concreteness the following discussion is formulated with respect to the posterior estimator; however it is easily adapted to other estimators like ACE. Foster et al. [2020] only considered variational models of the form $q_\phi(\theta|y)$, in which information about the design is encoded implicitly in y due to forward sampling through the model: $\theta \sim p(\theta)$, then $y \sim p(y|\theta, d)$. In this paradigm, the assumption is that this indirect signal is strong enough to effectively learn variational parameters ϕ that can be successful at evaluation and optimization. However, as demonstrated in the preceding subsection, this assumption does not hold as the complexity of the designs increases. Using variational models of this form, Foster et al. [2020] derived gradients with respect to both ϕ and d, in which the gradients with respect to d pass only through the likelihood, $p(y|\theta, d)$ (since the variational model is not differentiable with respect to d).

In the preceding chapter we proposed variational models of the form $q_\phi(\theta|y, d)$, directly encoding the design variables as input into the variational model. We have seen that these models are highly effective at estimating the EIG, both in settings in which we amortize over many designs, and even evaluation of a single complex design. However, this form requires us to re-derive the gradients from Foster et al. [2020] in order to incorporate the fact that $q_\phi(\theta|y, d)$ now explicitly depends on and is differentiable with respect to d.

We first consider the case where the likelihood, $p(y|\theta, d)$ can be re-parameterized with respect to an independent random variable, i.e., we can write $y = g(\epsilon, d, \theta)$ where $\epsilon \sim p(\epsilon)$. Then, we obtain stochastic gradients of the posterior estimator [Kingma and Welling, 2013, Rezende et al., 2014]:

$$\frac{\partial L_{post}}{\partial \phi} \approx \frac{1}{N} \sum_{n=1}^{N} \frac{\partial}{\partial \phi} \log q_\phi(\theta_n | y_n = g(\epsilon_n, d, \theta_n), d)$$

$$\frac{\partial L_{post}}{\partial d} \approx \frac{1}{N} \sum_{n=1}^{N} \frac{\partial}{\partial d} \log q_\phi(\theta_n | y_n = g(\epsilon_n, d, \theta_n), d)$$

where $\epsilon_n, \theta_n \sim p(\theta)p(\epsilon)$.

Re-parameterized gradient estimators are typically lower variance than alternative approaches, but is not always possible – say, when our likelihood is a discrete distribution. For this setting, we derive a score function estimator for the gradients with respect to d [Schulman et al., 2015].

$$\frac{\partial L_{post}}{\partial \phi} \approx \frac{1}{N} \sum_{n=1}^{N} \frac{\partial}{\partial \phi} \log q_\phi(\theta_n | y_n, d) \tag{5.1}$$

$$\frac{\partial L_{post}}{\partial d} \approx \frac{1}{N} \sum_{n=1}^{N} \log q_\phi(\theta_n | y_n, d) \frac{\partial}{\partial d} \log p(y_n | \theta_n, d) + \frac{\partial}{\partial d} \log q_\phi(\theta_n | y_n, d) \tag{5.2}$$

where $y_n, \theta_n \sim p(y, \theta | d)$. These equations are direct generalizations of Eq. (9)–(10) in Foster et al. [2020].

Generalizing the stochastic gradient algorithm of Foster et al. [2020] has two major advantages. First, we can optimize many designs simultaneously using a single variational model, whereas in Foster et al. [2020] each optimization process could optimize only a single design; multiple starting points require completely separate runs. Multiple starts is a common technique in mathematical optimization, and we show that integrating it in a single run yields significant computational savings. Second, our experiments show that directly encoding the designs gives far better performance, both in evaluation and optimization, with increasing benefits in more complex design spaces.

5.3.3 Gradient BOED Experiments

In these experiments we use the same GLM models from last chapter with 5 predictors and either 1 or 5 experimental units. We then optimize the design in a feasible region where each design variable can be varied continuously in the range $[-1, 1]$. This reflects a common parameterization where design variables are linearly transformed to reside in a common coded region [Goos and Jones, 2011, Montgomery, 2017]. For the linear models, we can

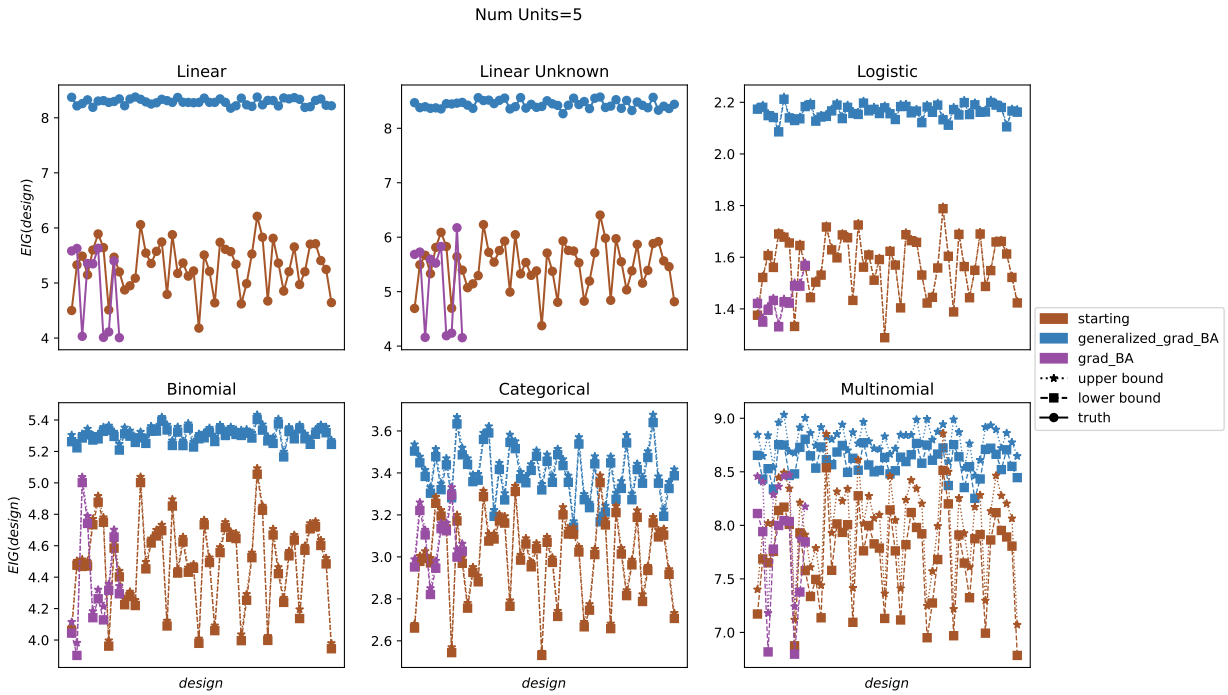

Figure 5.6: We compare two stochastic gradient algorithms for simultaneously optimizing the variational parameters ϕ and design variables d: the method proposed in Foster et al. [2020] and our generalized version, both using the BA bound as the objective. We generate 50 random starting designs and optimize them in parallel with our generalized approach. When not encoding d, we optimize the first 10 independently. For the linear GLMs we report the ground truth EIG, while for other GLM types we use the VNMC and ACE upper and lower bounds. Each designs consists of 5 experimental units. We see that our generalized algorithm achieves much better optimization performance.

calculate the ground truth exactly and use this value to plot the EIGs of the optimized designs. For the nonlinear models, we estimate the EIG using the VNMC and ACE bounds (with 1000 and 31 outer and nested samples, respectively) from an independently trained variational posterior amortized over the entire feasible region.

We compare our generalized stochastic gradient algorithm of Section 5.3 with its counterpart proposed in Foster et al. [2020]. Both algorithms simultaneously optimize the variational parameters and design variables using the BA bound; they differ in whether the variational form includes the design variables as input and is differentiable to it: $q_\phi(\theta|y,d)$ vs $q_\phi(\theta|y)$. We start with 50 random designs, which our generalized algorithm optimizes simultaneously.

Model Type	generalized_grad (s/# d)	grad (s/# d)
Linear	12.5	194.4
Linear Unk	12.4	195.3
Logistic	10.2	195.2
Binomial	10.4	200.4
Categorical	10.5	199.8
Multinomial	10.8	200.3

Table 5.1: Times for gradient BOED with d encoded vs d not encoded, Number of experimental units = 5

When not including d, we must perform a separate run for each starting design; due to the heavy computational burden, we elect to only optimize the first 10. For each GLM variant with 5 experimental units in the design, Figure 5.6 shows the EIG of the starting designs and the EIG of the final designs from both optimization algorithms, and Table 5.1 shows the seconds per design for each optimization process. Figure 5.7 and Table 5.2 show the corresponding plots for 1 experimental unit. From these experiments we see that with more complex designs (more experimental units), only our generalized stochastic gradient algorithm is able to optimize substantially above the initial designs, while the version of Foster et al. [2020] fails to meaningfully optimize at all. In contrast, when only optimizing designs with one experimental unit, both algorithms achieve roughly the same optimization quality. In both cases our generalized version was much faster, with a roughly $40 - 45\times$ speed-up for designs with one experimental unit and $15 - 20\times$ speed up for designs with 5 experimental units. Note that these numbers are dependent upon the number of starting designs – as this number increases so does the efficiency of our proposed generalized stochastic gradient algorithm.

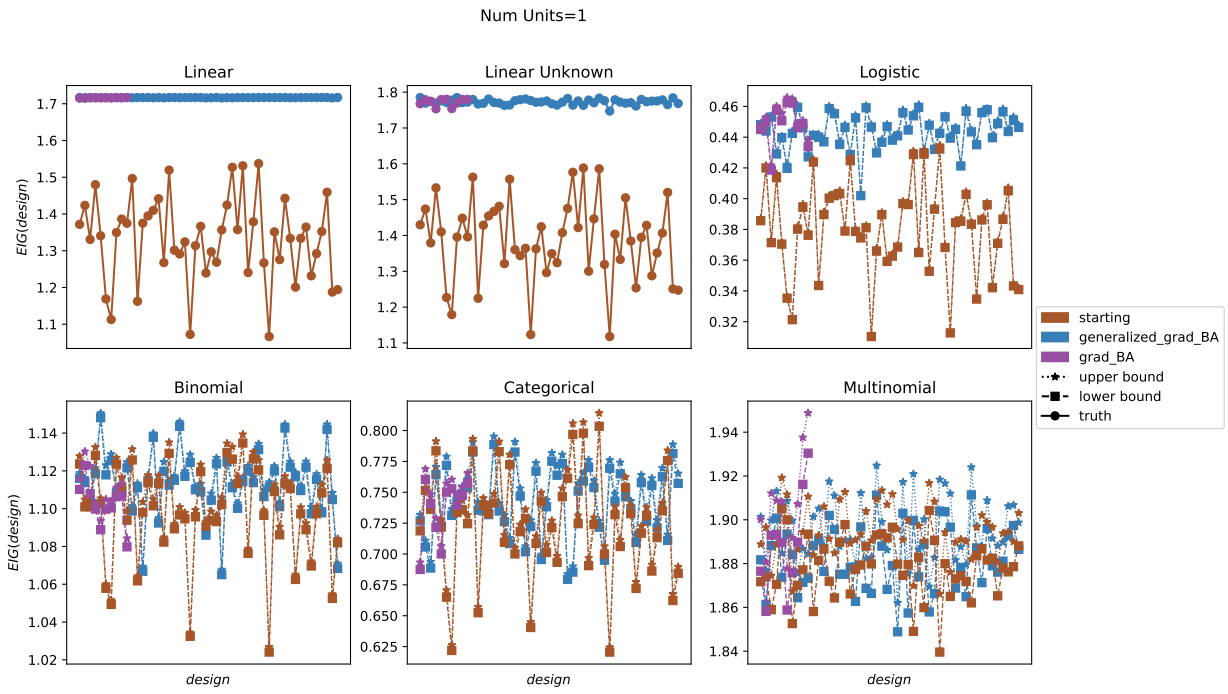

Figure 5.7: Here we compare two stochastic gradient algorithms for simultaneously optimizing the variational parameters ϕ and design variables d. The method proposed in Foster et al. [2020], labeled "optimized_no_d" with our generalized version labeled "optimized_with_d". We generated 50 random starting designs (brown) and were able to optimize them in parallel with our generalized approach and had to optimize them one at a time when not encoding d (here we only optimized the first 10 random starts due to the computational burden). For the linear based GLMs we could compute the ground truth of the EIG exactly and for the other variants we used the VNMC and ACE upper and lower bounds. Each designs consists of 1 experimental units and we can see both algorithms achieved roughly the same performance. Contrasting this to figure 5.6 we can see using our generalized approach is very important for complex designs with more experimental units and less so for simpler designs. However our generalized approach does result in substantial computational savings in both cases, see tables 5.2 and 5.1.

Model Type	generalized_grad (s/# d)	grad (s/# d)
Linear	5.0	199.9
Linear Unk	5.0	209.5
Logistic	4.5	205.3
Binomial	4.6	202.3
Categorical	4.5	199.7
Multinomial	4.7	200.2

Table 5.2: Times for gradient BOED with d encoded vs d not encoded, Number of experimental units = 1

Global Optima Experiments

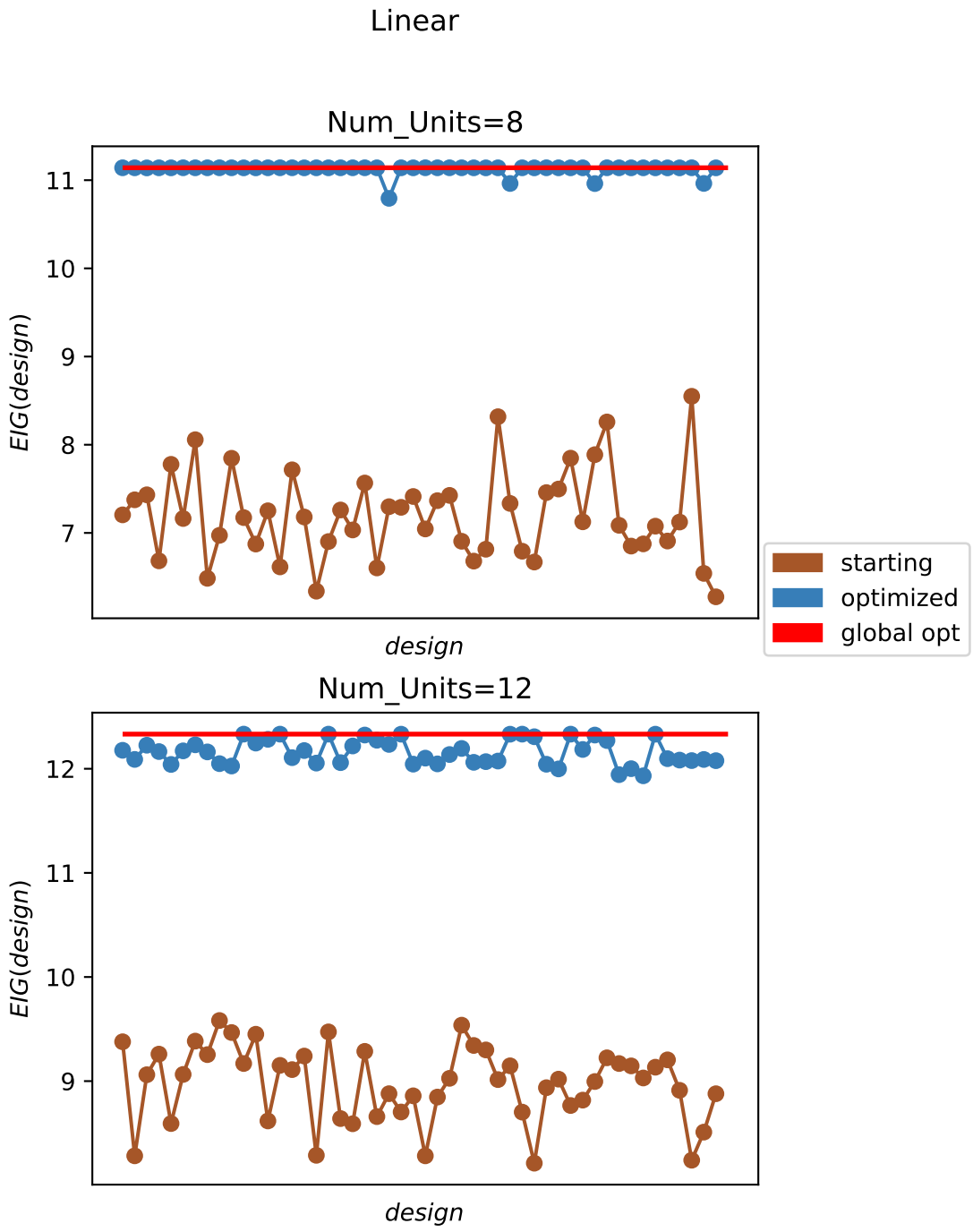

Figure 5.8: In this figure we show the results of our gradient-based algorithms for optimizing the design when the global optima is known. We consider the linear model with 8 and 12 experimental units and show that our model successfully optimizes to the global optima.

For the case of the linear model where the number of experimental units is a multiple of 4

that is greater than the total number of effects, the globally optimum design is known to be the orthogonal array designs [Goos and Jones, 2011, Montgomery, 2017]. Figure 5.8 shows the results of our optimization algorithm for the linear model with 5 predictors using 8 and 12 experimental units. We simultaneously optimize 50 designs and show that in both cases we successfully optimize to the global optima.

5.4 Head-to-Head Experiments

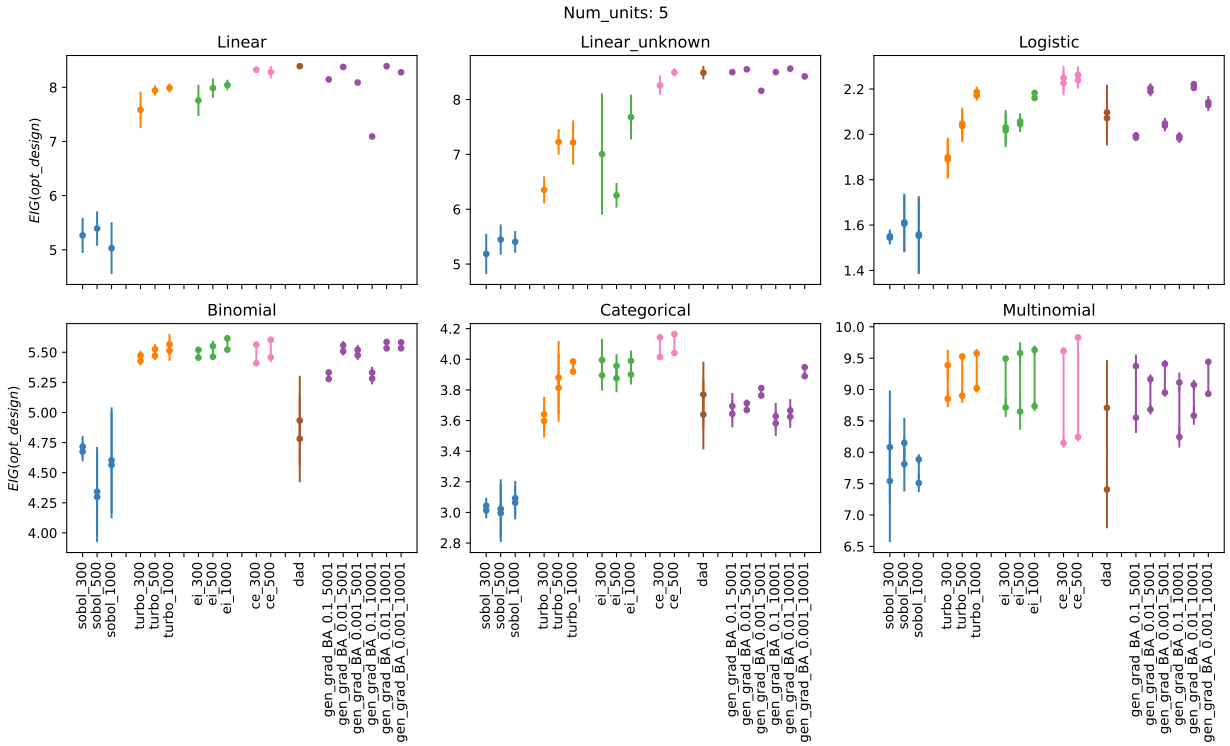

Figure 5.9: For designs of 5 experimental units we show the EIG of the optimized design for each optimization algorithm. For the linear models we use the ground-truth EIG, while for the others we use VNMC and ACE to compute lower and upper bound estimates of the EIG. Error bars are one standard deviation estimates from 5 independent runs.

We now present head-to-head optimization experiments of all the algorithms discussed in Section 5.2. For the two-stage methods (BO and CE), we first train a design-amortized variational model using the posterior estimator. Once trained, we use the variational model

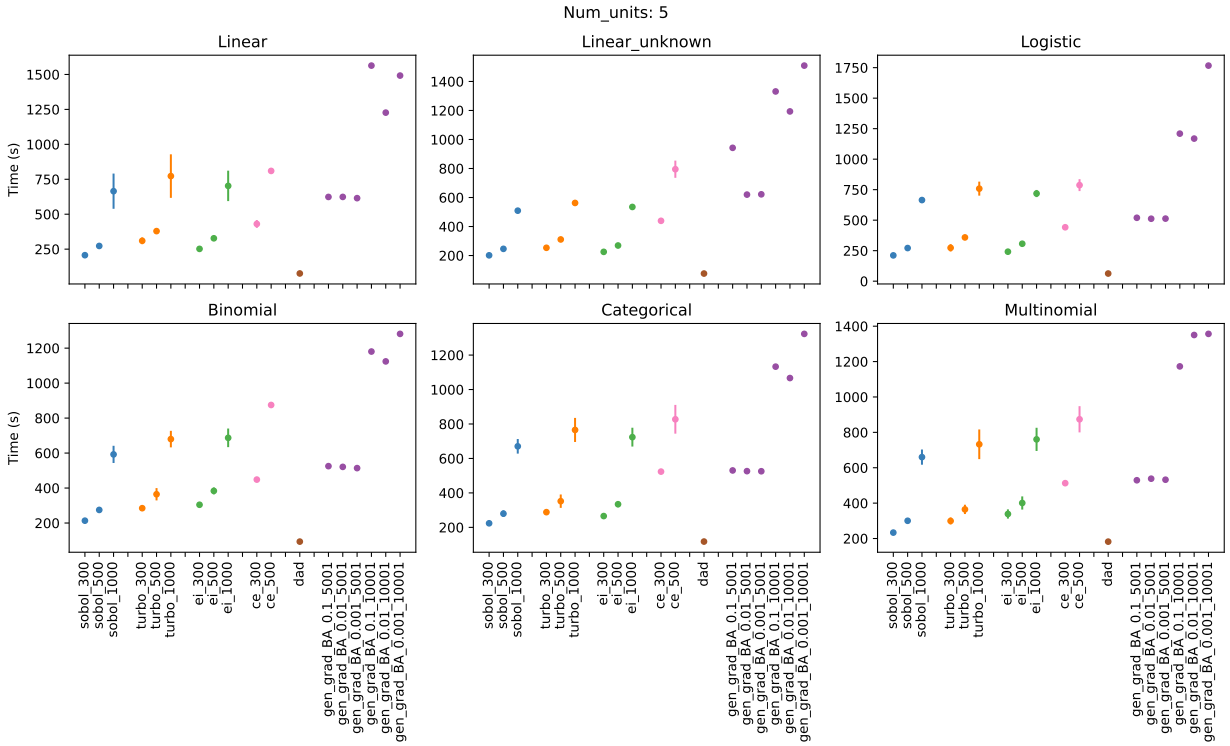

Figure 5.10: Run times for each optimization algorithm from figure 5.9. Error bars represent one standard deviation estimates from 5 independent runs for each optimization algorithm, except for the gradient methods where multiple starting designs were optimized in parallel using the same model.

and VNMC bound as the objective function to be optimized. For BO, we try two acquisition functions: expected improvement (EI) and trust region Bayesian optimization (TURBO), and test at three different fidelity levels: outer sample sizes N =300, 500, and 1000 (inner sample size M is set to $M = \sqrt{N}$) [Frazier, 2018, Eriksson et al., 2019]. For CE we use fidelity levels 300 and 500, and each run is trained with 3 random restarts with maximum 10 iterations for each trial. For the generalized stochastic gradient algorithm (Section 5.3) we try 5000 and 10000 optimization steps with design learning rates of 0.1, 0.01, and 0.001. We also compare to DAD using PCE as a our loss function [Foster et al., 2021] and train for 5000 steps. In order to keep the design suggested by DAD within the coded region of -1 to 1 we use the inverse hyperbolic tangent function at the last layer of its neural network (full details of the network used are in Section **??** of the appendix to the chapter). As a baseline,

we report the best design generated from a sequence of Sobol samples using the same number of samples that was used by TURBO. Figure 5.9 shows the estimated EIG of the maximum design for each optimization algorithm when optimizing over designs with 5 experimental units, and Figure 5.10 shows the total time for each optimization algorithm, including the time to train the variational model in the case of the two-stage methods. Each optimization algorithm was run for 5 independent trials and we include one sigma error bars for each EIG value (truth or estimator) computed. Figures 5.11 and 5.12 show the corresponding plots when optimizing designs with one experimental unit.

With the exception of the Sobol sampling method, all optimization algorithms are reasonably competitive with one another in terms of their best EIG design. When optimizing designs with 5 experimental units, DAD under-performs all other methods for the binomial model, while CE and BO are the top performers for the logistic and categorical models. We also observe that for designs with one experimental unit, BO, CE and DAD are the fastest performing methods, while gradient-BOED is somewhat slower (although this is dependent upon the number of optimization steps). For designs with 5 experimental units, DAD is consistently the fastest. One important observation is that the two-stage methods, BO and CE, are competitive with the gradient based methods both in terms of optimization quality and computational efficiency. This contradicts the experiments in Foster et al. [2020], which showed major advantages for gradient-BOED. The difference is attributable to their lack of design amortization when running two-stage algorithms, which is critical for remaining efficient under large or non-local changes in the design. This result is very encouraging, since gradient-based methods are less widely applicable in many real-world settings compared to two stage algorithms (see Section 5.2).

Figure 5.10 shows the wall-clock times for our head to head experiments of designs with 5 experiment units. As discussed in the main text all algorithms (except Sobol) received similar optimization performance with a slight edge to the CE algorithm for the logistic

and categorical model and with DAD performing noticeably worse on the binomial model. However DAD consistently achieves the fastest run-time for all GLM variants in this setting. Figures 5.11 and 5.12 show the corresponding plots for designs with one experimental unit. Here we see in terms of optimization performance all algorithms (except Sobol) are competitive in terms of optimization performance and with BO, CE and DAD achieving the fastest run times.

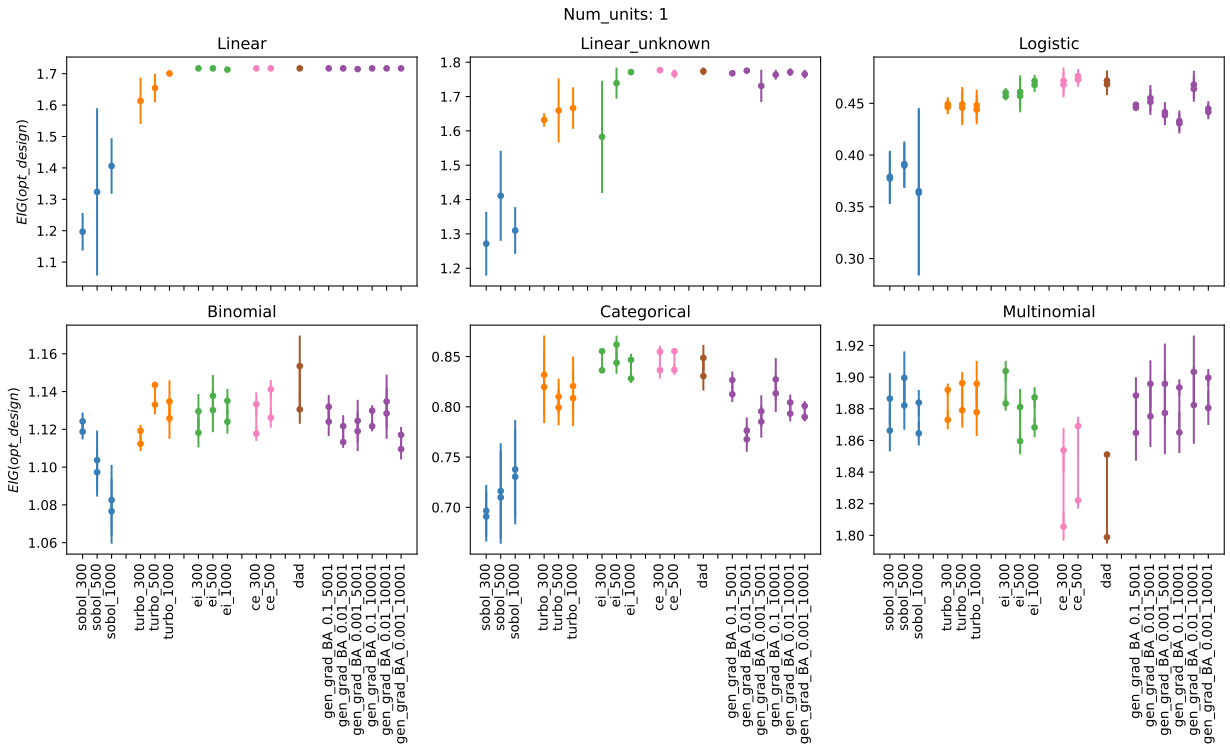

Figure 5.11: Here we show the EIG of the optimized design for each optimization algorithm for designs consisting of 1 experimental unit. For the linear and linear unknown GLMs we can calculate the EIG exactly, for the other GLM variants we use VNMC and ACE to calculate lower and upper bound estimates of the EIG. Error bars are one standard deviation estimates from 5 independent runs of each optimization algorithm.

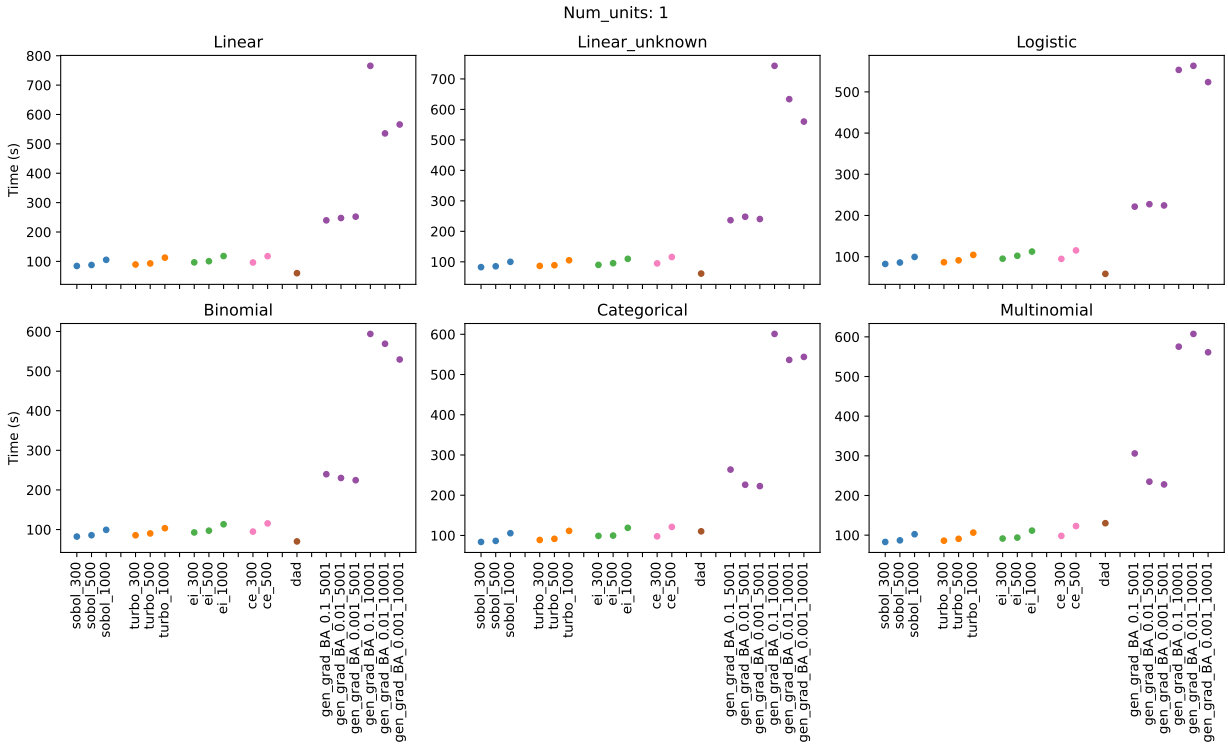

Figure 5.12: Run times for each optimization algorithm from figure 5.11. Error bars represent one standard deviation estimates from 5 independent runs for each optimization algorithm, except for the gradient methods where multiple starting designs were optimized in parallel using the same model.

5.5 Conclusion

In this work we demonstrate how design amortization yields significant benefits for variational methods in BOED. Our work builds on and extends that of Foster et al. [2020], which proposed a stochastic gradient algorithm for simultaneously optimizing the variational parameters ϕ and design variables d. We generalize this approach to a variational approximation that amortizes over designs, allowing us to optimize multiple starts, and propagate the design gradient through both the likelihood and variational approximation. Our approach gives both improved performance and significant computational savings, especially when optimizing "harder" designs with multiple experimental units. In addition we show that design amortization can significantly benefit two-stage optimization algorithms, where we first fit a variational model to estimate the EIG over the entire space of feasible designs, then use the resulting estimator in a wide variety of optimization algorithms, including discrete optimization algorithms such as the coordinate exchange algorithm. Design amortization allows two-stage algorithms to perform competitively compared to gradient methods, in contrast to previous comparisons [Foster et al., 2020]. This finding is important in practice, since it shows that variational methods can be used effectively in a wide variety of settings, including experiments with discrete design variables, and multi-objective optimization problems. This work builds on the variational forms proposed in the last chapter, extending them to perform design optimization, rather than simply evaluation. Taken together, this and prior work [Foster et al., 2019, 2020, 2021, Ivanova et al., 2021, Kennamer et al., 2022] provide strong evidence of variational methods' flexibility and effectiveness for BOED, so that these methods merit serious consideration by researchers when designing their experiments.

5.6 Appendix

Gradient Components

In our generalized gradient algorithm for simultaneously optimizing the variational parameters, ϕ, and design variables, d, there are two gradient paths influencing the design variables. The path through the likelihood, $p(y|\theta, d)$, and the path through the variational posterior, $q_\phi(\theta|y, d)$. We now try to understand the effects of these components on the optimization of d. In figures 5.13, 5.14, 5.15, 5.16, 5.17, and 5.18 we plot the results of our generalized gradient optimization algorithm using the BA bound considering three alternatives: taking gradients through both the likelihood and variational posterior (labeled "model_q_diff"), taking gradients with respect to the design through just the likelihood (labeld "model_diff") and taking gradients through just the variational posterior (labeled "d_diff"). For each GLM variant we try three different learning rates on the design variables of .1, .01 and .001 (columns) and consider designs with 1 and 5 experimental unit(s) (rows) and train for 5000 steps. We can see that in the majority of cases most of the optimization comes from the gradients that come through the likelihood while the gradients coming through the variational posterior contribute very little on their own. There are some note able exceptions to this, for the binomial, categorical and multinomial models we see that only taking gradients through the variational posterior results in better optimization performance when we use larger learning rates.

We investigate this further using the linear model where ground truth can be calculated analytically with little computational cost. In figure 5.19 we optimize 50 starting designs and vary the number of experimental units between: 1, 5, 24, and 32 (rows of figure) for 1000 steps and compute the EIG of the optimized designs every 100 steps. We do this for the case where we use gradients through both the likelihood and through the variational posterior and for the case where we only use gradients through the likelihood to optimize

the designs (columns of figure). We can see that for the cases with smaller number of experimental units per designs (1 and 5) there is little difference between the two methods however for the designs with larger number of experimental units (24 and 32) we see much better performance, both in terms of rate of convergence and final optimization status, for the case that we use gradients through both the likelihood and variational posterior to optimize the design variables.

The two sets of experiments above suggest that in simpler cases it is probably fine to only take gradients through the model provided we have tuned our learning rate correctly, but both paths of the gradient signal become important as the complexity of the design variables being optimized increases.

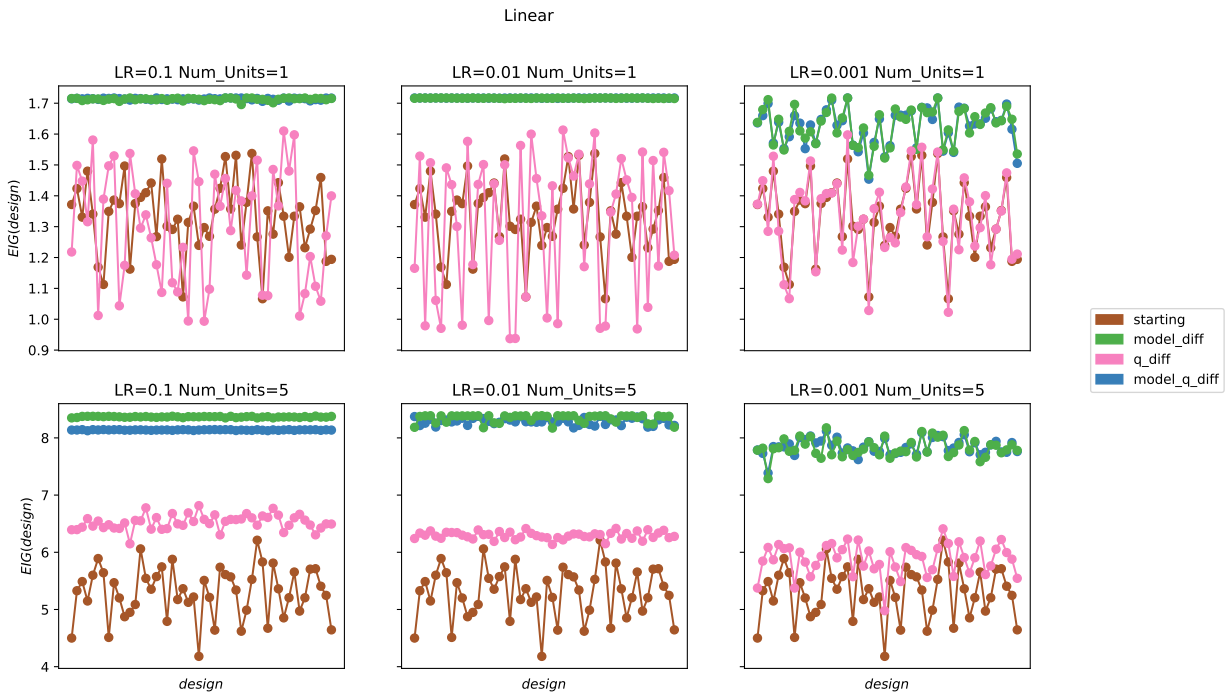

Figure 5.13: Here we compare the effects of the possible gradient paths on design optimization for the linear model. We consider taking gradients with respect to the design through both the likelihood and variational posterior (labeled "model_q_diff"), through just the likelihood (labeled "model_diff") and taking gradients through just the variational posterior (labeled "d_diff"). We vary both the number of experimental units per designs between 1 and 5 and the learning rate used to optimize the design variables between 0.1, 0.01, and 0.001. We plot ground truth EIG for the optimized and starting designs.

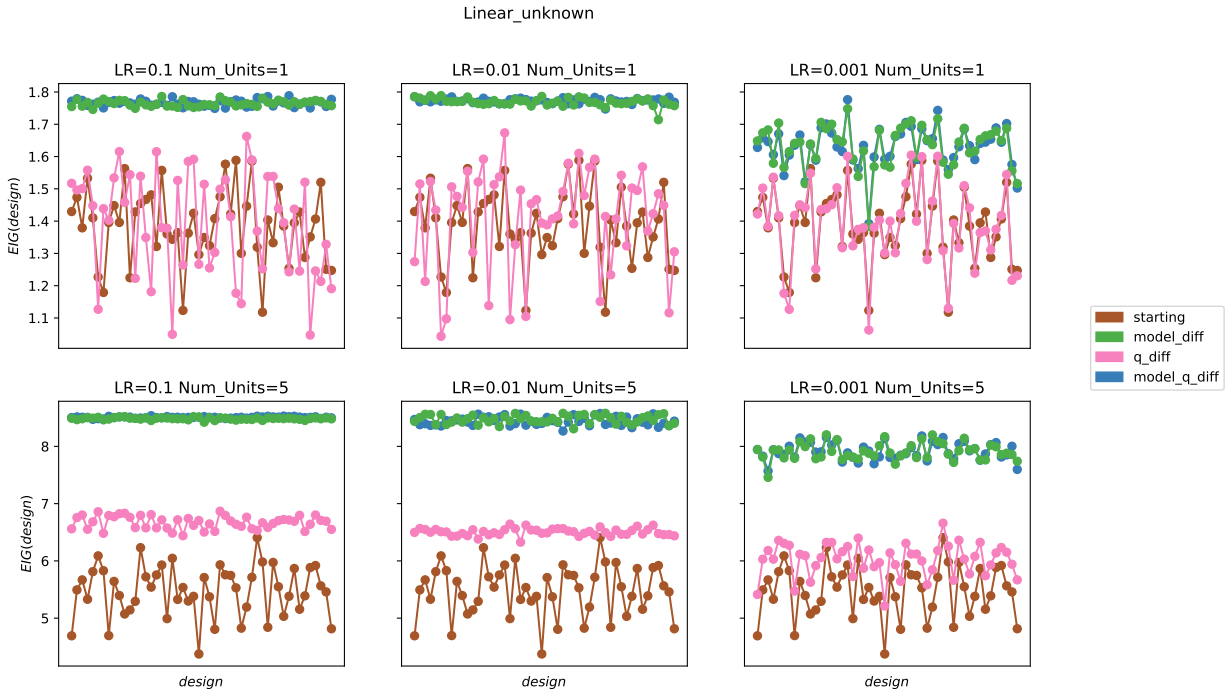

Figure 5.14: Same as figure 5.13 for the linear unknown model.

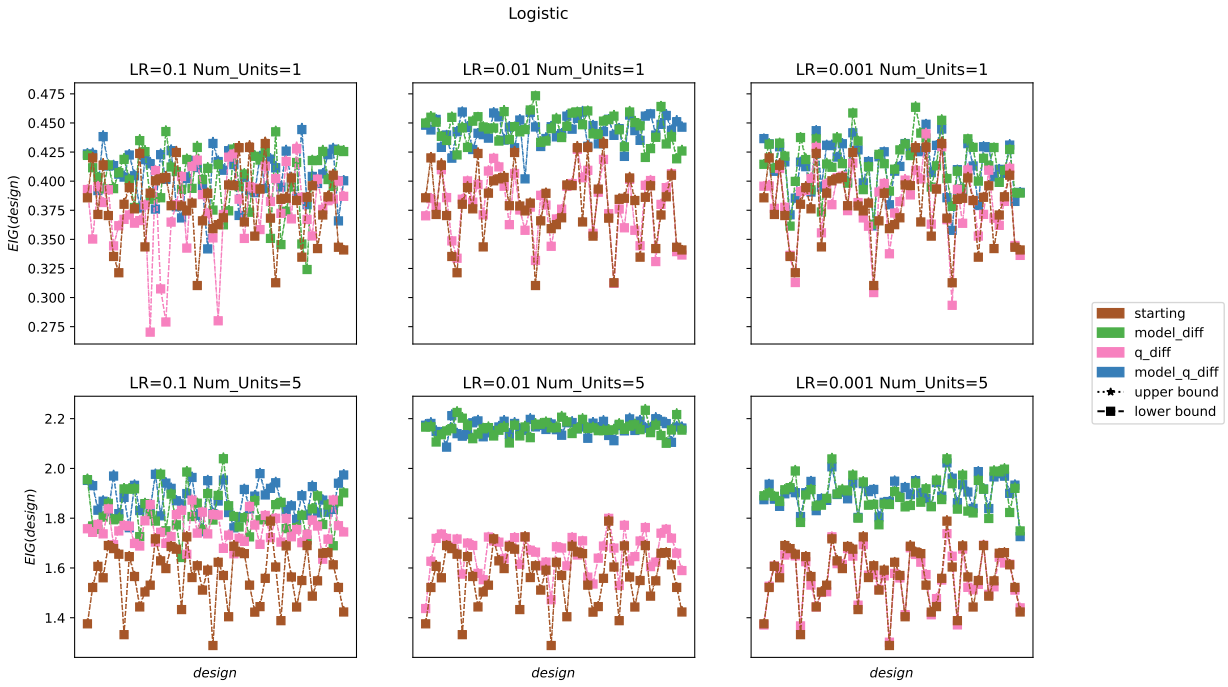

Figure 5.15: Same as figure 5.13 for the linear unknown model.

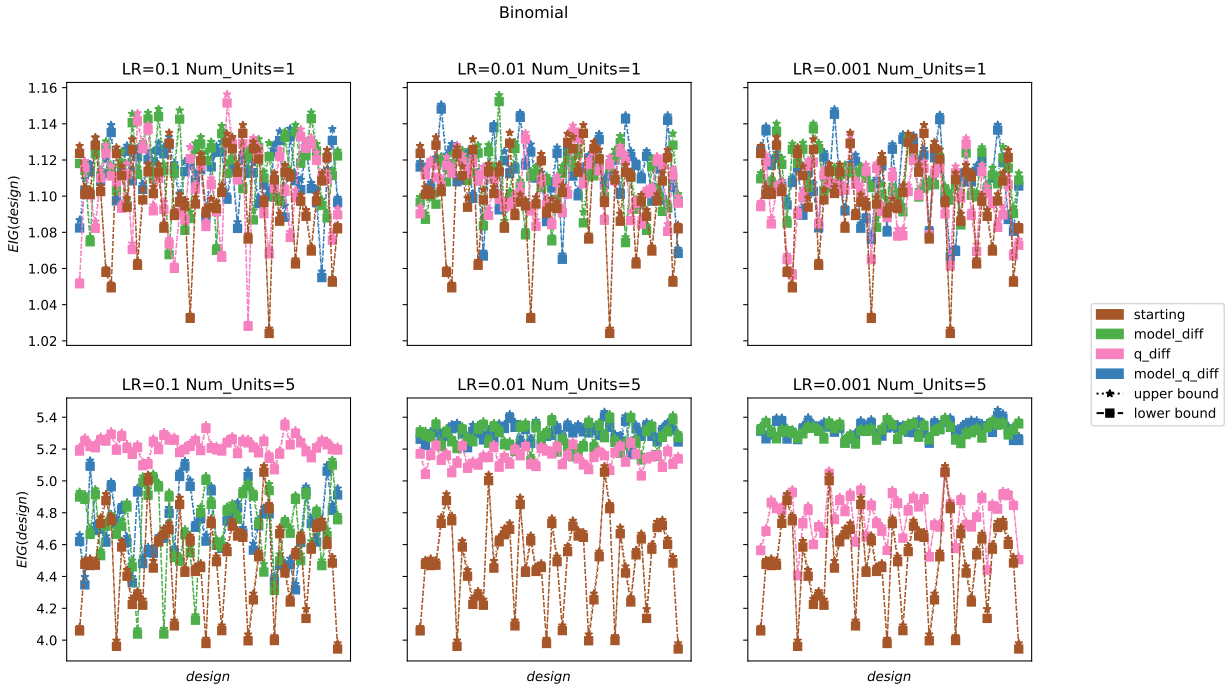

Figure 5.16: Same as figure 5.13 for the binomial model. Here we cannot compute ground truth so plot the the VNMC and ACE upper and lower bounds with 1000 outer samples and 31 nested samples.

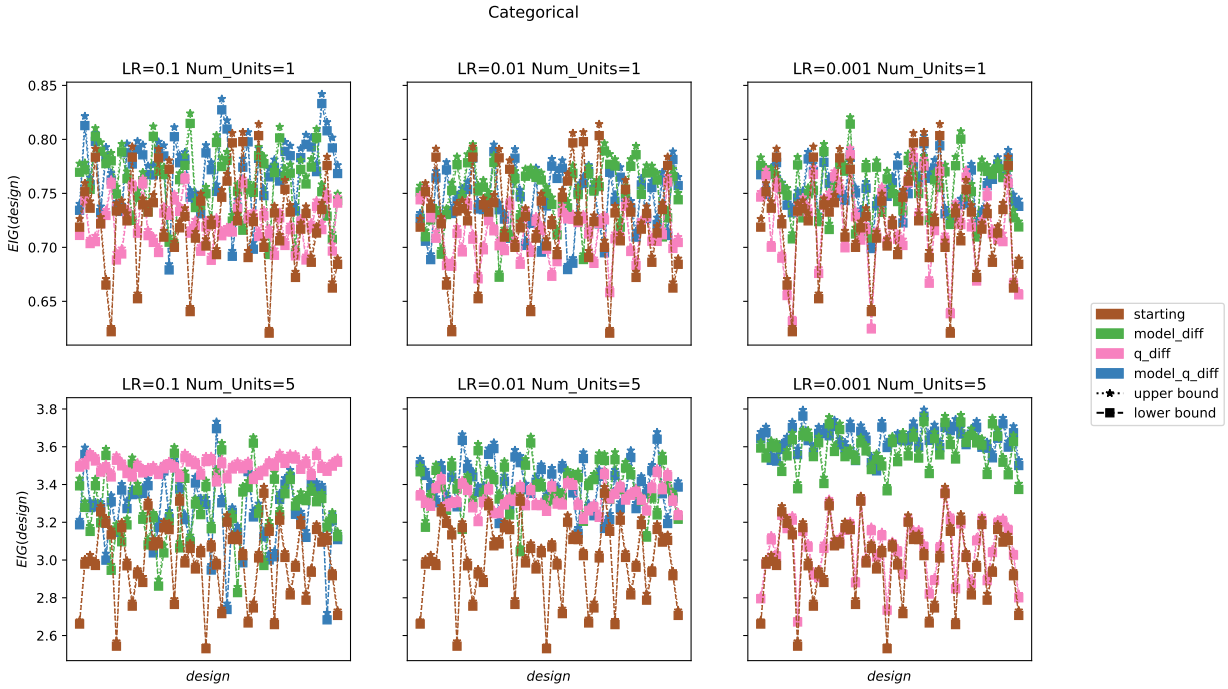

Figure 5.17: Same as figure 5.13 for the categorical model. Here we cannot compute ground truth so plot the the VNMC and ACE upper and lower bounds with 1000 outer samples and 31 nested samples.

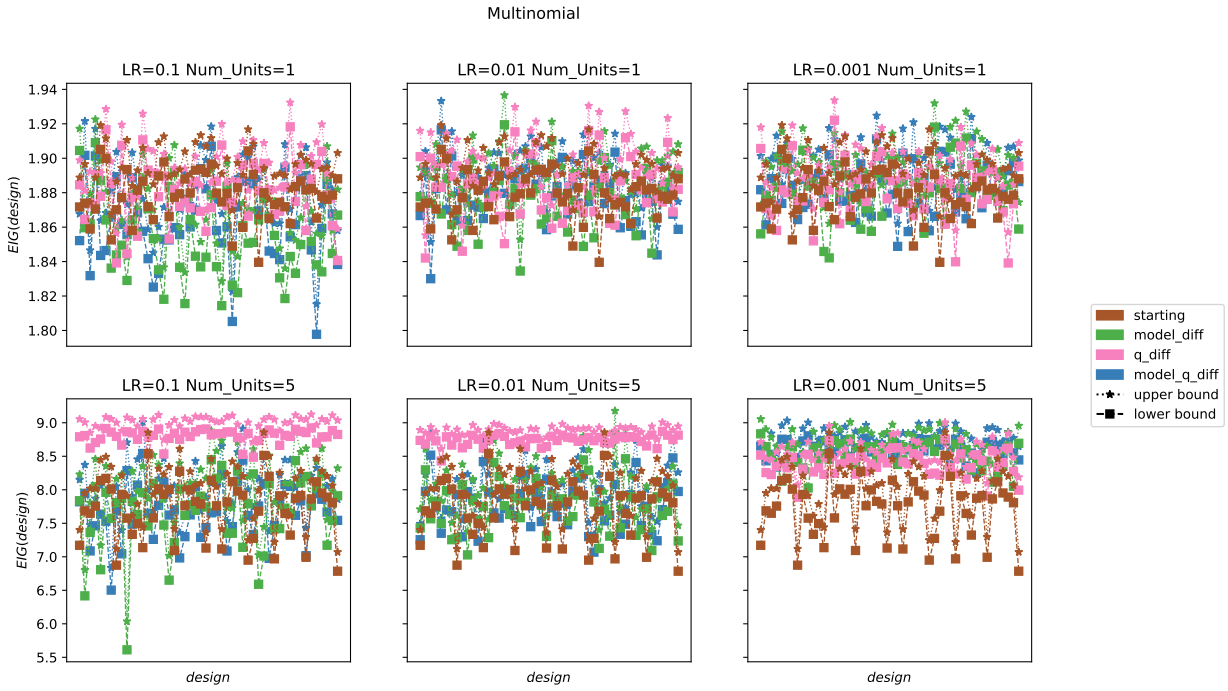

Figure 5.18: Same as figure 5.13 for the multinomial model. Here we cannot compute ground truth so plot the the VNMC and ACE upper and lower bounds with 1000 outer samples and 31 nested samples.

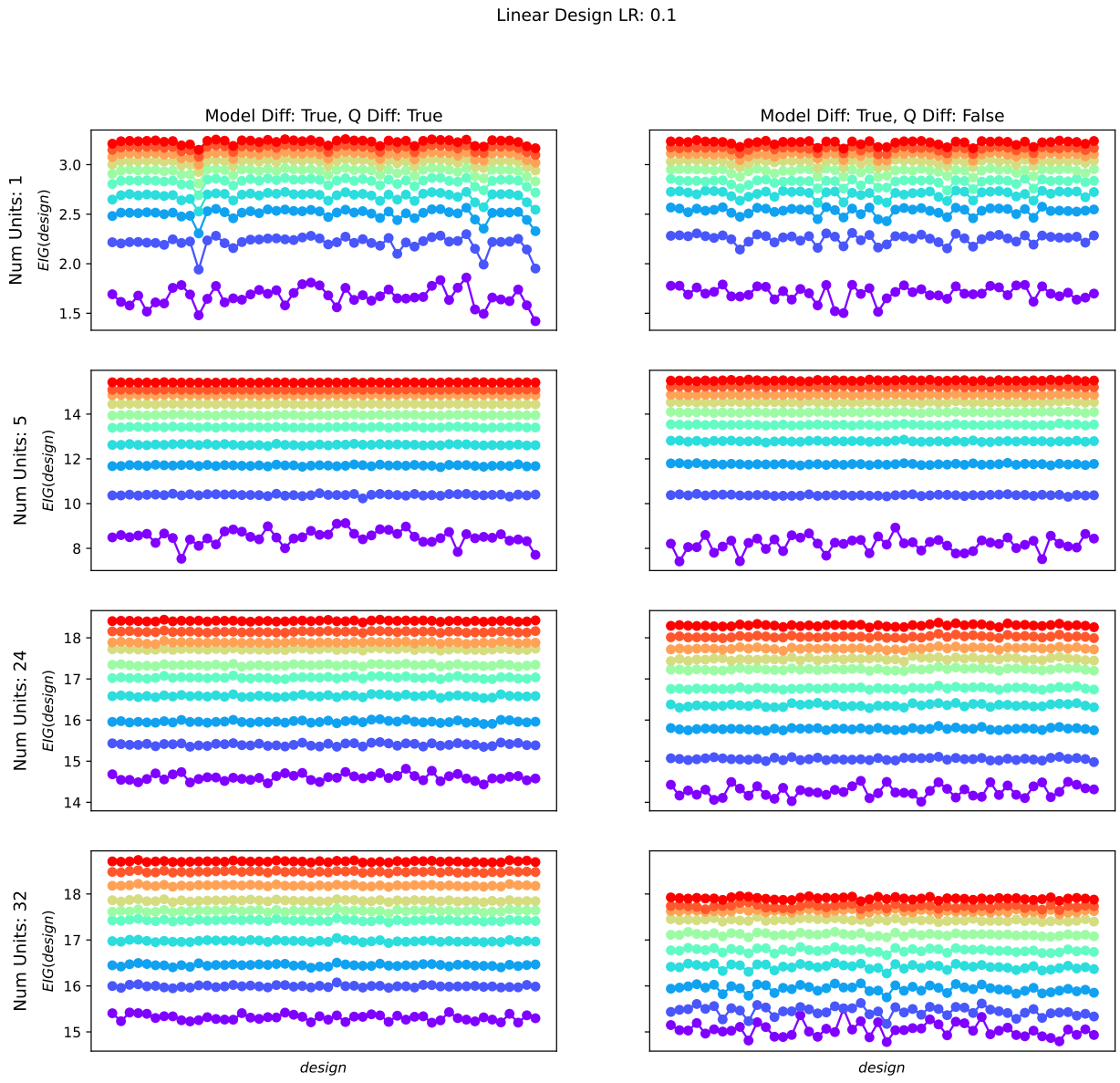

Figure 5.19: Here we optimize 50 starting designs for the linear model and vary the number of experimental units between: 1, 5, 24, and 32 (rows of figure) for 1000 steps and compute the EIG of the optimized designs every 100 steps (from purple to red). We do this for the two cases: first we use gradients through both the likelihood and through the variational posterior and for the case where we only use gradients through the likelihood to optimize the designs (columns of figure). We plot ground truth EIG of each design.

DAD Architecture

Given that in our experiments the experimental units within a design have no inherent ordering we used a set invariant architecture of the form presented in Zaheer et al. [2017] for our DAD network. It consists of an encoding network receiving as input a history of design and observation pairs, $(D_1, y_1), (D_2, y_2), \ldots, (D_t, y_t)$ where D_i is a matrix of shape $N_E \times N_p + 1$ and y_i is a vector of N_E dimensions. for each time step the designs are concatenated with the observations column wise and all time-steps are concatenated together row-wise. We first use a residual embedding network with 2 residual blocks (2 linear layers with 64 neurons) to encode each row to a vector of 120 dimensions. We then pass through this through 2 attention layers followed by dropout a linear projection with 32 neurons with a final dropout layer (dropout prob of 0.1). This completes the set encoder giving a vector representation for each row which are then aggregated together using the sum function. This aggregated representation is passed through an emitter network, which is a residual network with 2 residual blocks where the final output layer is sized according to the number of design variables and passed through the inverse tangent function constrain design variables within the coded region between -1 and 1. To start the process, before any designs were run, we follow the procedure from Foster et al. [2021] and encode the input as single vector of zeros of size $N_p + 1 + 1$ to match what will be the number of design variables and size of single units output (1 in our case). In our experiments we consider only optimizing a single design and set $t = 0$ meaning that our network always takes this zero vector as input.

Chapter 6

Active Learning for Spectroscopic Follow Up

> The researcher hoping to break new ground in the theory of experimental design should involve himself in the design of actual experiments. The investigator who hopes to revolutionize decision theory should observe and take part in the making of important decisions.
>
> George E. P. Box

In this chapter we discuss an applied active learning project to the field of astronomy. This work was started in 2017 at the Cosmostastics Initiative's (COIN) fourth resident program in 2017 held in Clermont-Ferrand France. COIN is an organization aimed at fostering collaborations between scientists in the astrophysical sciences and in the statistical sciences and has produced many successful interdisciplinary projects. This work is no exception and has benefited from the expertise of a diverse group of scientist with specialities in observational astronomy, supernova physics, machine learning and statistics. My contributions to

the project were centered around developing and implementing many of the active learning techniques used in this work. This chapter is closely based on our published work in Ishida et al. [2019] and Kennamer et al. [2020].

The chapter is organized by first introducing the problem and motivating the use of active learning for spectroscopic follow up in section 6.1. We then discuss the dataset, feature extraction and evaluation metrics for all experiments in sections 6.2 and 6.3. In section 6.4 we cover the various active learning strategies used throughout this project. Our early work in Ishida et al. [2019] used rather simple active learning techniques and ignored some important realistic constraints; this was done to reduce complexity as we built our pipeline and began our initial investigation of active learning in this setting, which was very novel at the time. We then expanded this initial investigation in Kennamer et al. [2020], adding in a great deal of real world complexity while also investigating more advanced active learning techniques. We present the experiments in this chapter in order of increasing complexity, starting with the experiments in Ishida et al. [2019]: this begins with a static full light curve analysis in section 6.5 followed by a real-time analysis in section 6.6 and finally examining batch selection for fixed batch sizes in section 6.7. We then discuss the experiments of Kennamer et al. [2020] in section 6.9, first discussing the additional real world constraints and experimental design, and finally presenting our experimental results which incorporate all of these constraints. We end with a concluding discussion in section 6.10. Code reproducing this work can be found here [1]

6.1 Introduction

The standard cosmological model rests on three observational pillars: primordial Big-Bang nucleosynthesis [Gamow, 1948], the cosmic microwave background radiation [Spergel et al.,

[1] https://github.com/COINtoolbox/RESSPECT

2007, Planck Collaboration et al., 2016], and the accelerated cosmic expansion [Riess et al., 1998, Perlmutter et al., 1999] – with Type Ia supernovae (SNe Ia) playing an important role in probing the last one. SNe Ia are astronomical transients that are used as standardizable candles in the determination of extragalactic distances and velocities [Hillebrandt and Niemeyer, 2000, Goobar and Leibundgut, 2011]. However, between the discovery of a SN candidate and its successful application in cosmological studies, additional research steps are necessary.

Once a transient is identified as a potential SN, it must go through three main steps: i) classification, ii) redshift estimation, and iii) estimation of its standardized apparent magnitude at maximum brightness [Phillips, 1993, Tripp, 1998]. Ideally, each SN thus requires at least one spectroscopic observation (preferably around maximum – items i and ii) and a series of consecutive photometric measurements (item iii). Since we are not able to get spectroscopic measurement for all transient candidates, soon after a variable source is detected a decision must be made regarding its spectroscopic follow-up, making coordination between transient imaging surveys and spectroscopic facilities mandatory. From a traditional perspective, taking a spectrum of a transient that ends up classified as a SNIa results in the object being included in the cosmological analysis. On the other hand, if the target is classified as non-Ia, spectroscopic time for cosmology is essentially considered "lost"[2].

In the last few decades, a strong community effort has been devoted to the detection and follow-up of SNe Ia for cosmology. Classifiers (human or artificial) on which follow-up decisions are based have become increasingly efficient in identifying SNe Ia from early stages of their light curve evolution – successfully targeting them for spectroscopic observations [e.g. Perrett et al., 2010]. The available cosmology data set has grown from 42 [Perlmutter et al., 1999] to 740 [Betoule et al., 2014] in that period of time. This success helped build

[2]This is strictly for cosmological purposes; spectroscopic observations are extremely valuable, irrespective of the transient in question, though for different scientific goals.

consensus around the paramount importance of SNe Ia for cosmology. It has also encouraged the community to add even more objects to the Hubble diagram and to investigate the systematic uncertainties which currently dominate the dark energy error budget [e.g. Conley et al., 2011]. Henceforth, SNe Ia are major targets of many current - e.g. *Dark Energy Survey*[3] (DES), *Panoramic Survey Telescope and Rapid Response System*[4] (Pan-STARRS) - and upcoming surveys - e.g. *Zwicky Transient Facility*[5] (ZTF) and the *Rubin Observatory Legacy Survey of Space and Time*[6] (LSST).

LSST will produce measurements of flux (brightness) within broad regions of the electromagnetic spectrum (filters). These *photometric* observations can be obtained in seconds to minutes for all sources within the telescope field of view, in effect providing a snapshot of that region of the sky at that moment in time. The survey is expected to cover the entire southern sky every three days for a total period of ten years. Nevertheless, to obtain reliable classifications, it is necessary to scrutinize each object with high resolution *spectroscopic* observations. These allow the astronomer to identify the presence of individual chemical elements, which facilitates assigning it to the correct group within the astronomical zoo. This labeling process requires more telescope time (on the order of hours), a different type of instrument, and sometimes significant effort from an experienced observational astronomer who can reduce the data and translate it into a label. Although the availability of spectroscopic resources is also expected to increase during the next decade, it will always be orders of magnitude lower than its photometric counterpart.

Full cosmological exploitation of wide-field imaging surveys necessarily requires a framework able to infer reliable spectroscopically-derived features (redshift and class) from purely photometric data. Provided a particular transient has an identifiable host, redshift can be

[3]https://www.darkenergysurvey.org/
[4]https://panstarrs.stsci.edu/
[5]http://www.ptf.caltech.edu/ztf/
[6]https://www.lsst.org/

obtained before/after the event from the host observations (spectroscopic or photometric) or even from the light curve itself [e.g. Wang et al., 2015]. On the other hand, classification should primarily be inferred from the light curve[7]. This work concerns itself with the latter.

It is important to keep in mind that, regardless of the method chosen to circumvent these issues, photometric information will always carry a larger degree of uncertainty than those from the spectroscopic scenario. Photometric redshift estimations are expected to have non-negligible error bars and, at the same time, any kind of classifier will carry some contamination to the final SNIa sample. Nevertheless, if we manage to keep these effects under control, we should be able to use photometrically observed SNe Ia to increase the statistical significance of our results. The question whether the final cosmological outcomes surpass those of the spectroscopic-only sample enough to justify the additional effort is still debatable. Despite a few reports in this direction using real data from the *Sloan Digital Sky Survey*[8] [SDSS - Hlozek et al., 2012, Campbell et al., 2013] and Pan-STARRS [Jones et al., 2017], the answer keeps changing as different steps of the pipeline are improved and more data become available. Nevertheless, there seems to be a consensus in the astronomical community that we have much to gain from such an exercise.

The vast literature, with suggestions on how to improve/solve different stages of the SN photometric classification pipeline, is a demonstration of the positive attitude with which the subject is approached. For more than 15 years the field has investigated attempts relying on a wide range of methodologies: colour-colour and colour-magnitude diagrams [Poznanski et al., 2002, Johnson and Crotts, 2006], template fitting techniques [e.g. Sullivan et al., 2006], Bayesian probabilistic approaches [Poznanski et al., 2007, Kuznetsova and Connolly, 2007], fuzzy set theory [Rodney and Tonry, 2009], kernel density methods [Newling et al., 2011] and more recently, machine learning-based classifiers [e.g. Richards et al., 2012a, Ishida and

[7] Although, see Foley and Mandel 2013.
[8] http://www.sdss.org/

de Souza, 2013, Karpenka et al., 2013, Lochner et al., 2016b, Möller et al., 2016, Charnock and Moss, 2017, Dai et al., 2017].

In 2010, the *SuperNova Photometric Classification Challenge* [SNPCC - Kessler et al., 2010] summarized the state of the field by inviting different groups to apply their classifiers to the same simulated data set. Participants were asked to classify a set of light curves generated according to the DES photometric sample characteristics. As a starting point, they were provided with a spectroscopic sample enclosing $\sim 5\%$ of the total data set, and for which class information was disclosed. The organizers posed three main questions: full light curve classification with and without the use of redshift information (supposedly obtained from the host galaxy) and an early epoch classification – where participants were allowed to use only the first 6 observed points from each light curve. The goal of the latter was to access the capability of different algorithms to advise on spectroscopic targeting while the SN was still bright enough to allow it. A total of 10 groups replied to the call, submitting 13 (9) entries for the full light curve classification with (without) the use of redshift information. No submission was received for the early epoch scenario.

The models competing in the SNPCC were quite diverse, including template fitting, statistical modelling, selection cuts and machine learning-based strategies [see summary of all participants and result in Kessler et al., 2010]. Classification results were consistent among different methods with no particular model clearly outperforming all the others. The main legacy of this initiative however, was the updated public data set made available to the community once the challenge was completed. It became a test bench for experimentation, specially for machine learning approaches [Newling et al., 2011, Richards et al., 2012a, Karpenka et al., 2013, Ishida and de Souza, 2013].

One particularly challenging characteristic of the SN classification problem, also present in the SNPCC data, is the discrepancy between spectroscopic and photometric samples. In

a supervised machine learning framework, we have no alternative other than to use spectroscopically classified SNe as training. This turns out to be a serious challenge, since the training set is is over a different distribution of objects compared to the test set. Due to the stringent observational requirements of spectroscopy, the distributions between spectroscopic and photometric astronomical samples will never align. But the situation is even more drastic for SNe, where the spectroscopic follow-up strategy was designed to target as many Ia-like objects as possible. Albeit modern low-redshift surveys try to mitigate and counterbalance this effect (e.g. ASASSN[9], iPTF[10]), the medium/high redshift ($z > 0.1$) spectroscopic sample is still heavily under-represented by all non-Ia SNe types. Spectroscopic observations are so time demanding, and the rate with which the photometric samples are increasing is so fast, that the situation is not expected to change even with dedicated spectrographs [OzDES - Childress et al., 2017]. This issue has been pointed out by many post-SNPCC machine learning-based analysis [e.g. Richards et al., 2012a, Karpenka et al., 2013, Varughese et al., 2015, Lochner et al., 2016b, Charnock and Moss, 2017, Revsbech et al., 2017]. In spite of the general consensus being that one should prioritize faint objects for spectroscopic targeting, as an attempt to increase representativeness [Richards et al., 2012a, Lochner et al., 2016b], the details on how exactly this should be implemented are yet to be defined.

Thus the question still remains: how do we optimize the distribution of spectroscopic resources with the goal of improving photometric SN identification? Or, in other words, how do we construct a training sample that maximizes accurate classification with a minimum number of labels, i.e., spectroscopically-classified SNe? The above question is similar in context to that addressed by an area of machine learning called *active learning* [Settles, 2012, Balcan et al., 2009, Cohn et al., 1996].

Active Learning (AL) iteratively identifies which objects in the target (photometric) sample

[9]http://www.astronomy.ohio-state.edu/ãssassin/index.shtml
[10]https://www.ptf.caltech.edu/iptf

would most likely improve the classifier if included in the training data – allowing sequential updates of the learning model with a minimum number of labelled instances. It has been widely used in a variety of different research fields, e.g. natural language processing [Thompson et al., 1999], spam classification [DeBarr and Wechsler, 2009], cancer detection [Liu, 2004] and sentiment analysis [Kranjc et al., 2015]. In astronomy, AL has been successfully applied in multiple tasks: determination of stellar population parameters [Solorio et al., 2005], classification of variable stars [Richards et al., 2012b], optimization of telescope choice [Xia et al., 2016], static supernova photometric classification [Gupta et al., 2016], and photometric redshift estimation [Vilalta et al., 2017]. There are also reports based on similar ideas by Masters et al. [2015], Hoyle et al. [2016].

In this work, we show how active learning enables the construction of optimal training datasets for SNe photometric classification, providing observers with a spectroscopic follow-up protocol on a night-by-night basis. The framework respects the time evolution of the survey providing a decision process which can be implemented from the first observational nights –avoiding the necessity of adapting legacy data and the consequent translation between different photometric systems. As a case study, we focus on the problem of active learning for the binary classification problem of Type Ia vs non-Ia, but the overall structure can be easily generalized for multi-classification tasks.

6.2 Data

All experiments are conducted on the data released from the SNPCC. This is a simulated data set constructed to mimic DES observations. The sample contains 20216 supernovae (SNe) observed in four DES filters, $\{g, r, i, z\}$, among which a subset of 1103 are identified as belonging to the *spectroscopic* sample. This subset was constructed considering observations through a 4m (8m) telescope and limiting r-band magnitude of 21 (23.5) [Kessler et al., 2010].

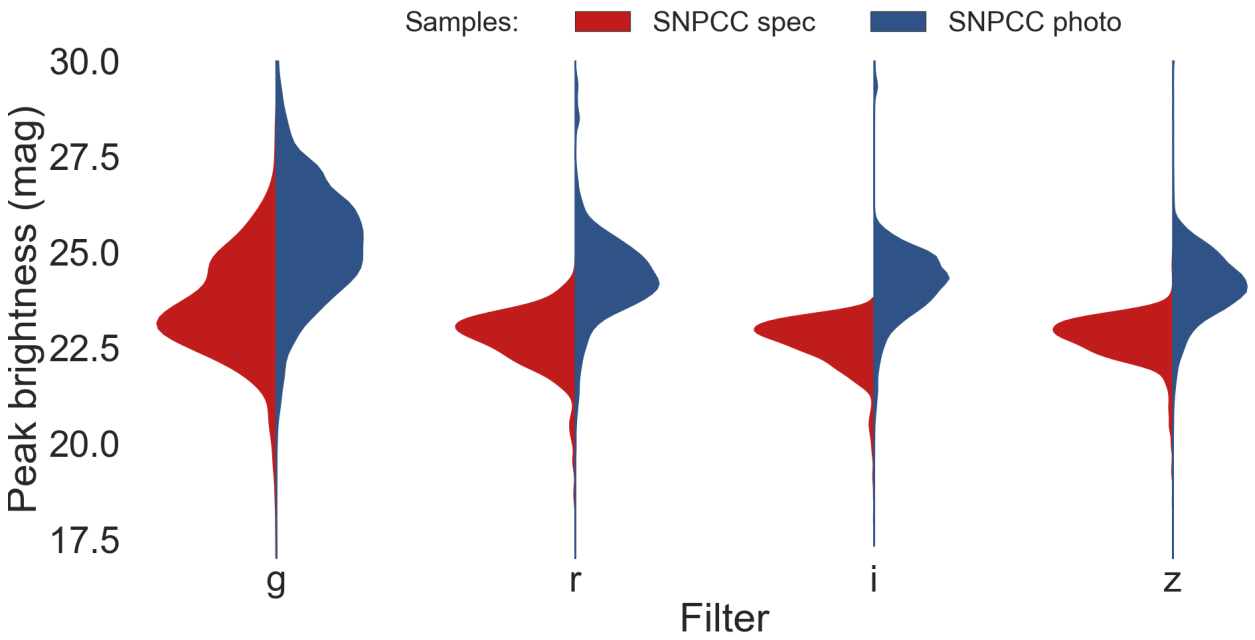

Figure 6.1: Comparison between simulated peak magnitudes in the SNPCC spectroscopic (red - training) and photometric (blue - target) samples. Violin plots show both distributions in each of the DES filters.

Thus, it resembles closely biases foreseen in a realistic spectroscopic sample when compared to the photometric one. Among them, we highlight the predomination of brighter objects (figure 6.1) with higher signal to noise ratio (SNR, figure 6.2), and the predominance of SNe Ia over other SN types (figure 6.3). Hereafter, the spectroscopic sample will be designated *SNPCC spec* and the remaining objects will be addressed as *SNPCC photo*.

6.3 Preprocessing and Metrics

6.3.1 Feature extraction

For each supernova, we observe its light curve, i.e., the evolution of brightness (flux) as a function of time, in four DES filters $\{g, r, i, z\}$. For most machine learning applications, this information needs to be homogenized before it can be used as input to a learning algo-

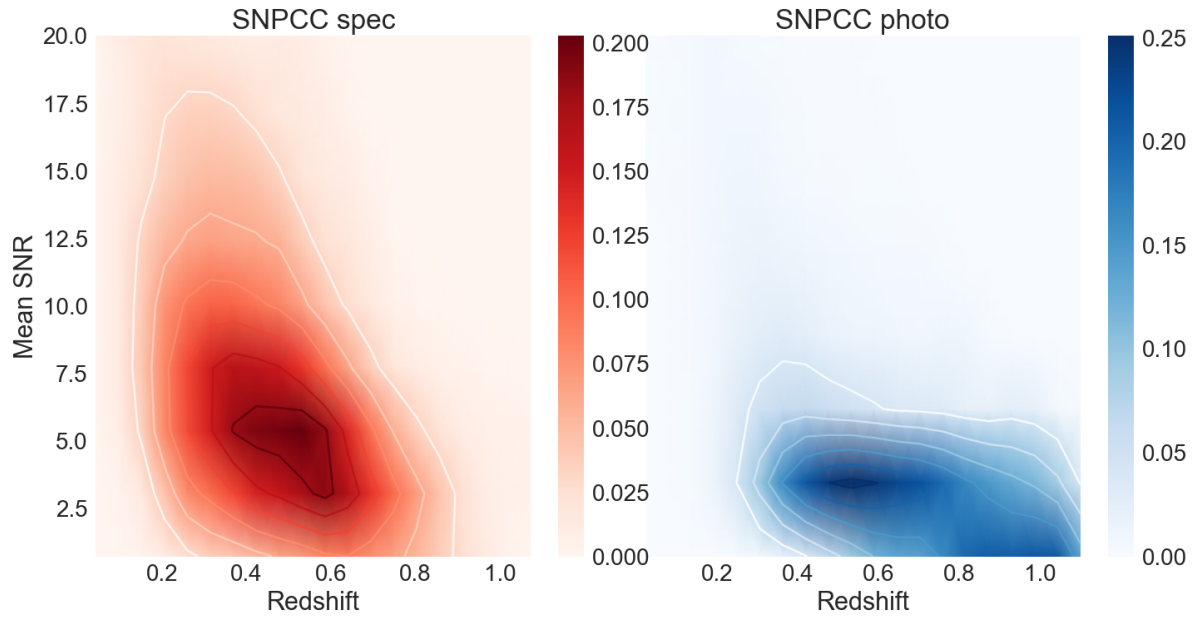

Figure 6.2: Distribution of mean signal to noise ratio (SNR) in the SNPCC spectroscopic (red - training) and photometric (blue - target) samples.

rithm[11]. There are many ways in which this feature extraction step can be performed: via a proposed analytical parametrization [Bazin et al., 2009b, Newling et al., 2011], comparisons with theoretical and/or well-observed templates [Sako et al., 2008] or dimensionality reduction techniques [Richards et al., 2012a, Ishida and de Souza, 2013]. The literature has many examples showing that, for the same classification model, the choice of the feature extraction method can significantly impact classification results [see Lochner et al., 2016b, and references therein].

In what follows, we use the parametrization proposed by Bazin et al. [2009b],

$$f(t) = A\frac{e^{-(t-t_0)/\tau_f}}{1+e^{(t-t_0)/\tau_r}} + B, \tag{6.1}$$

where A, B, t_0, τ_f and τ_r are parameters to be determined. We fit each filter independently

[11]Exceptions include algorithms able to deal with a high degree of missing data [e.g. Charnock and Moss, 2017, Naul et al., 2018].

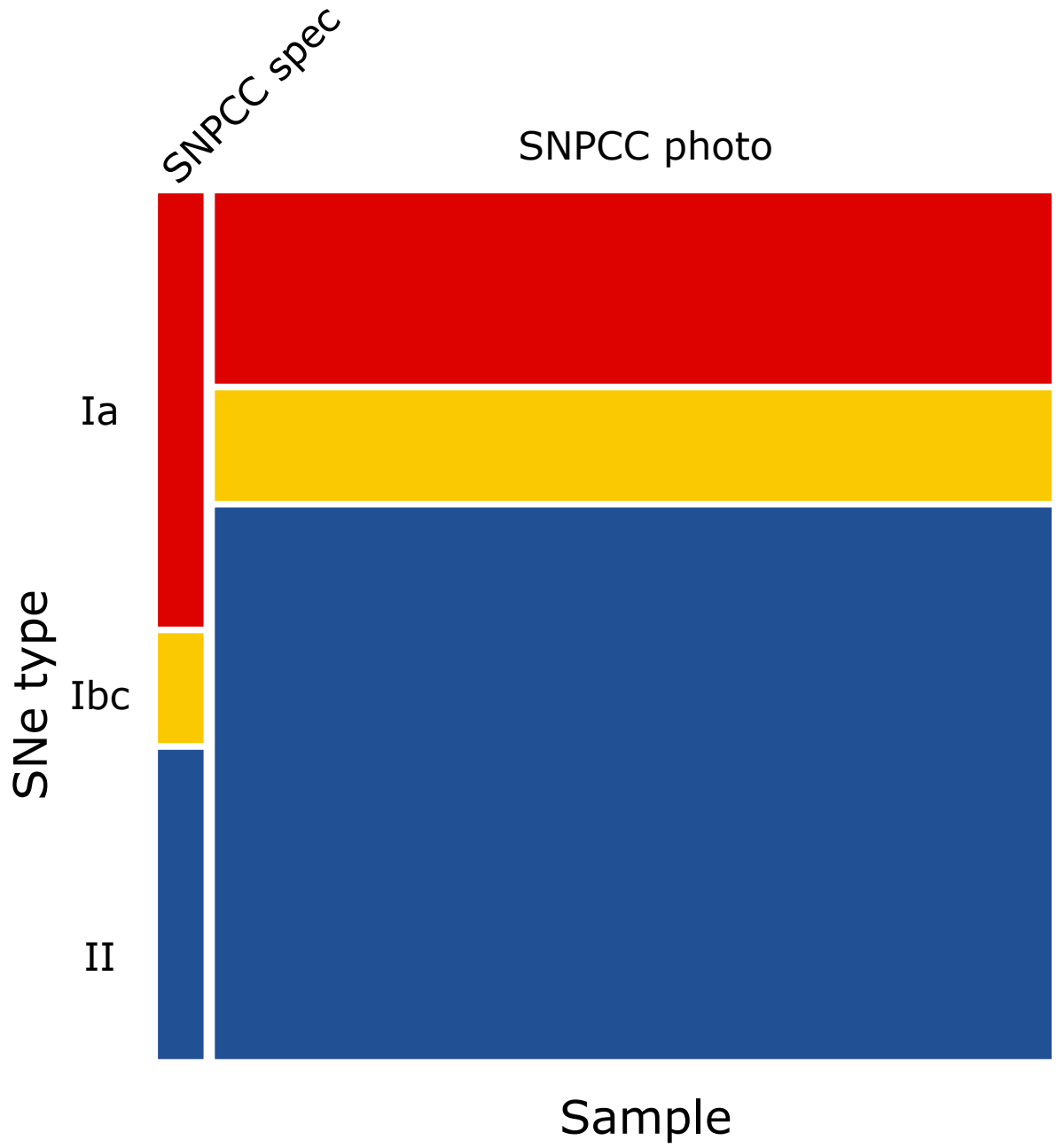

Figure 6.3: Populations of different supernova types in the SNPCC spectroscopic (training) and photometric (target) samples. The spectroscopic sample holds 599 (51%) Ia, 144 (13%) Ibc and 400 (36%) II, while the photometric sample comprises 4326 (22%) Ia, 2535 (13%) Ibc and 12442 (65%) II.

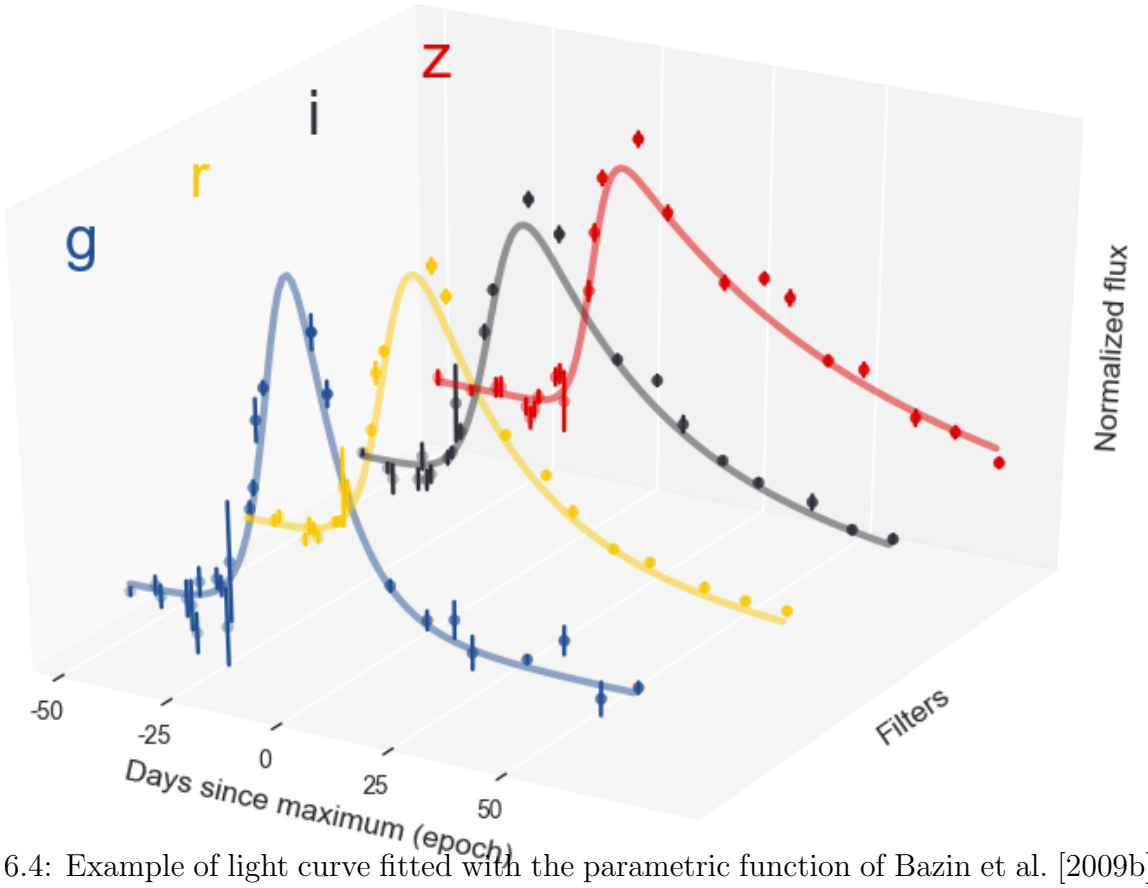

Figure 6.4: Example of light curve fitted with the parametric function of Bazin et al. [2009b] - equation (6.1). The plot shows measurements for a typical well-sampled type Ia at redshift $z \sim 0.20$, in each one of the 4 DES filters (dots and error bars) as well as its best fitted results (lines).

in flux space with a Levenberg-Marquardt least-square minimization [Madsen et al., 2004]. Figure 6.4 shows an example of flux measurements, corresponding errors and best-fit results in all 4 filters for a typical, well-sampled, SN Ia from SNPCC data.

Although not optimal for such a diverse light curve sample, the parametrization given by equation (6.1) was chosen for being a fast feature extraction method. Moreover, as any parametric function, it returns the same number of parameters independently of the number of observed epochs, which is crucial for dealing with an inhomogeneous time-series which changes on a daily basis. We stress that a more flexible feature extraction procedure still holding the characteristics described above would only improve the results presented here.

6.3.2 Metrics

Our choice of a metric to quantify classification success goes beyond the use of classical accuracy (equation (6.2)) – especially when the populations are unbalanced (figure 6.3). In order to optimize information extraction, this choice must take into account the scientific question at hand.

In the traditional SN case, the goal is to improve the quality of the final SNIa sample for further cosmological use. In this context, a false negative (a SNIa wrongly classified as non-Ia) will be excluded from further analysis posing no damage on subsequent scientific results. On the other hand, a false positive (non-Ia wrongly classified as a Ia) will be mistaken by a standard candle, biasing the cosmological analysis. Thus, purity (equation (6.4)) of the photometrically classified SNIa set is of paramount importance. At the same time, we wish to identify as many SNe Ia as possible (high efficiency – equation (6.3)), in order to guarantee optimal exploitation of our resources. Taking such constraints into consideration, Kessler et al. [2010] proposed the use of a figure of merit which penalizes classifiers for false positives (equation (6.5)). Throughout our analysis, classification results will be reported according to these 4 metrics:

$$\text{accuracy} = \frac{N_{\text{sc}}}{N_{\text{tot}}}, \tag{6.2}$$

$$\text{efficiency} = \frac{N_{\text{sc,Ia}}}{N_{\text{tot,Ia}}}, \tag{6.3}$$

$$\text{purity} = \frac{N_{\text{sc,Ia}}}{N_{\text{sc,Ia}} + N_{\text{wc,nIa}}}, \tag{6.4}$$

$$\text{figure of merit} = \frac{N_{\text{sc,Ia}}}{N_{\text{tot,Ia}}} \times \frac{N_{\text{sc,Ia}}}{N_{\text{sc,Ia}} + W N_{\text{sc,nIa}}}, \tag{6.5}$$

where N_{sc} is the toal number of successful classifications, N_{tot} the total number of objects in

the target sample, $N_{\mathrm{sc,Ia}}$ the number of successfully classified SNe Ia (true positives), $N_{\mathrm{tot,Ia}}$ the total number of SNe Ia in the target sample, $N_{\mathrm{wc,nIa}}$ the number of non-Ia SNe wrongly classified as SNe Ia (false positives) and W is a factor which penalizes the occurrence of false positives. Following Kessler et al. [2010] we always use $W = 3$.

In the AL framework we propose, the metrics above are used to quantify the classification results in the target sample. They were calculated after the classifications were performed and had no part in the decision making algorithm (further details in Section 6.4.

6.4 Active Learning

The label constrained environment described above is a prime candidate to benefit from active learning. However due to real world constraints there are a number of practical challenges that are often not considered in other studies of active learning. First, the population that can be spectroscopically observed will always differ from the target population. This requires active learning to perform well when the pool set (the set that can be queries) is not representative of the validation and test sets. Second, we must choose to label a light curve before fully observing it – since the object must be observed near maximum brightness. Finally we must also include non-constant costs in the selection of our batch sizes. Most active learning strategies assume constant costs and thus restrict the queried batch to a fixed size per iteration – these are known as *cardinality constraints*. In our case, each object has a different cost (time necessary to get a label) and our total budget is constrained by the number of hours of spectroscopic telescope time available per night. These are known as *knapsack constraints* and have been studied in the context of discrete optimization Krause et al. [2008], Krause and Golovin [2014]. These challenges make our work an excellent case study to stress test how standard and commonly used active learning algorithms hold up to real world conditions and using modern machine learning classifiers. These contraints will

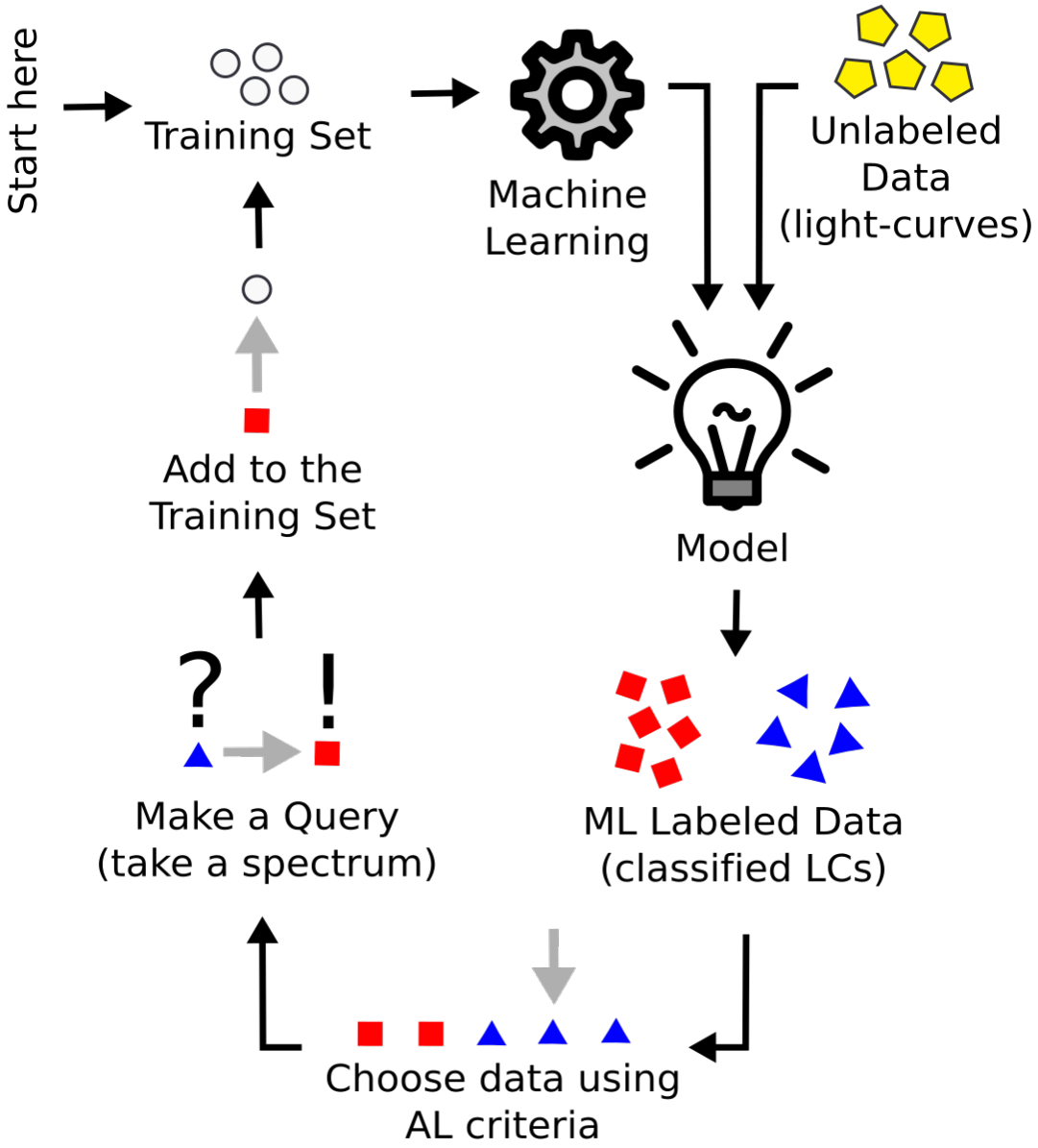

Figure 6.5: Schematic illustration of the Active Learning (AL) workflow in the context of photometric light curve classification. Starting at the top left, the training set (spectroscopic sample, grey circles), is used to train a machine learning model; the resulting model is then applied to the unlabelled data (photometric light-curves, yellow pentagons). This initial model returns a classification for each data point of the unlabelled set (now represented as red squares and blue triangles). The AL algorithm is then used to choose a data point from the unlabelled data with highest potential to improve the classification model (identified by the grey arrow). The label of this point is then queried (a spectrum is taken). Once the true label of the queried point is known, it is added to the training set (converted into a grey circle), and the process is repeated.

be further studies in section 6.9.

We formulate our problem in terms of pool-based active learning pictorially represented in figure 6.5, coupled with uncertainty sampling driven techniques Settles [2012].

In the first set of experiments from Ishida et al. [2019] we employed uncertainty sampling on a single random forest classifier with 1000 trees. For each query-able light curve the classifier outputs its probability estimate of it being a Type Ia supernova and we queried the object with probability closest to 0.5 (most uncertain). This simple choice of classifier and AL strategy was useful to reduce complexity as we implemented the pipeline and ran experiments on this new application for active learning. In Kennamer et al. [2020] we added a great deal of complexity taking into account more realistic constraints and employing more advanced AL methods, which are described below.

Specifically we used query by committee by performing bagging over a random forest classifier Freund et al. [1997], Seung et al. [1992], Breiman [1996]. Query by committee is a known active learning strategy that invokes a set of classifiers (committee) for each object's label estimation. In this context, the queried object will be the one that exhibits strong disagreement between the members of the committee. In bagging, the training data is sub-sampled with replacement and each subset is used to train a different model (using Random Forests) – each of these models is then considered a member of the committee. The criteria used to quantify the disagreement between the output of committee members is called a query selection strategy. In all experiments presented here we considered a committee of size 10, each composed of 100 trees, and only varied the query selection strategy.

Let (x, y) denote feature and label pairs where in our case x corresponds to the concatenated best fit parameters (Equation (6.1)) for the 4 DES filters, measured from a single object and y is a binary label identifying Ia/non-Ia SNe. Let $P_\theta(y|x)$ denote the predictive probability output from a single committee member, where θ encompasses the parameters of the learning

model. Since each member of the committee generates a predictive probability over the estimated class, we can define the average committee predictive probability as

$$P_C(y|x) = \frac{1}{N_C} \sum_c P_{\theta_c}(y|x), \tag{6.6}$$

where N_C is the committee size and the sum runs over all committee members. We use this distribution to build all other selection strategies.

One of the most common selection strategies is the soft vote entropy Settles [2012]. In information theory, entropy measures the expected (average) amount of information uncovered by identifying the outcome of a random trial MacKay et al. [2003]. In this context, if a given object has a high probability of belonging to a given class, it is unlikely that labeling it will add new information to the model. On the other hand, if an object has equal probability of belonging to all possible classes, labeling it will uncover currently missing information and certainly improve our model. Considering the prediction of each committee member as a vote, this strategy will choose to query the object with highest entropy among all committee members. Mathematically, we have

$$x^* = \arg\max_x \left(- \sum_y P_C(y|x) \log P_C(y|x) \right), \tag{6.7}$$

where x^* is the queried object.

We also use the average Kullback-Leibler (KL) divergence between the individual committee members and the average committee probability as a query selection strategy Kullback and Leibler [1951], Settles [2012],

$$x^* = \arg\max_x \left(\frac{1}{N_C} \sum_c KL(P_{\theta_c}(y|x)||P_C(y|x)) \right). \tag{6.8}$$

Thus, selecting the objects with the most disagreement among the committee members. This selection strategy is equivalent to the one defined by Bayesian Active Learning by Disagreement (BALD) Houlsby et al. [2011].

6.4.1 Batch Strategies

The query strategies described above target one individual object per active learning cycle. When moving to batch queries (targeting multiple objects per night), these strategies can face serious challenges, such as querying redundant data points Settles [2012]. The problem of querying diverse batches can typically be framed as a discrete optimization problem and is known to be computationally challenging. However, in practical applications, selecting multiple queries at a time is a requirement. Here we assume constant cost of acquisition across all data points; this requirement will be relaxed in the next subsection. An efficient approach if the query selection strategy is monotonic submodular, is to use a greedy algorithm which provides batches with a $(1 - 1/e)$ approximation to the optimal solution Nemhauser et al. [1978], Krause et al. [2008]. Both of the query strategies given above are monotonic submodular Kirsch et al. [2019]. While in Kirsch et al. [2019] this technique was called BatchBALD, we refer to it as *BatchKL* since our technique for approximating the disagreement region is not Bayesian.

Let the sets $x_1, ..., x_b$ and $y_1, ..., y_b$ be denoted as $x_{1:b}$ and $y_{1:b}$, where b is the batch size. Using the definition of mutual information, $\mathcal{I}$, for two sets of random variables we have,

$$\mathcal{I}(y_{1:b}, \theta | x_{1:b}, \mathcal{D}_{train}) = \mathcal{H}(y_{1:b} | x_{1:b}, \mathcal{D}_{train}) - \mathbb{E}_{p(\theta|\mathbf{D}_{train})} \mathcal{H}(y_{1:b} | x_{1:b}, \theta, \mathcal{D}_{train}), \tag{6.9}$$

where $\mathcal{H}$ refers to the entropy, $\mathbf{D}_{train}$ the training data and $\mathbb{E}$ is an expectation. The mutual information can be seen as the intersection of the information content between two sets of random variables Yeung [1991]. This strategy accounts for overlaps in the information con-

tent between different data points, $x_{1:b}$, and model parameters, θ. By accounting for these overlaps we can avoid querying redundant data points. This function is monotonic submodular and thus, when optimized with a greedy algorithm, provides a $(1 - 1/e)$ approximation to the optimal solution Kirsch et al. [2019]. We use equation (6.9) to define the BatchKL strategy as:

$$x^*_{1:b} = \arg\max_{x_{1:b}} \mathcal{I}(y_{1:b}, \theta | x_{1:b}, \mathcal{D}_{train}). \tag{6.10}$$

Note that the first term on the right hand side of equation (6.9), the joint entropy, is also monotonic submodular. We use it to define the strategy we call BatchEntropy:

$$x^*_{1:b} = \arg\max_{x_{1:b}} \mathcal{H}(y_{1:b} | x_{1:b}, \mathcal{D}_{train}). \tag{6.11}$$

In addition to these two batch strategies we will also test a strategy that takes the top b points from equation (6.7). We will refer to this strategy as Uncertainty Sampling Entropy (USE).

6.4.2 Non-Constant Cost

As mentioned, each object in our pool sample has a different cost (in our setting, the telescope time required for labeling). In addition, our budget (in terms of telescope time) is very limited and needs to be used as efficiently as possible. We assume we have access to 6 hours of observation in 4m-class telescopes and 6 hours in 8m-class telescopes per night. The batch strategies defined in the last section assumed cardinality constraints where all objects had identical costs. We now consider the case where each object has different cost and we have a fixed budget each night (knapsack constraints [see, e.g., Krause and Golovin, 2014]). We show results where we fill up objects to each telescope, without considering their individual cost, until the budget of each telescope is full. We first assign objects to the 4m telescope

until the budget is exhausted, at which point objects are assigned to the 8m telescope. We also tested strategies where we scale the query metrics by the cost of each object and greedily select objects after scaling[12]. However, we do not include these results as they were nearly identical to the simpler approach.

6.5 Static Full Light Curve Analysis

We begin by applying the complete framework described in the previous subsections to static data. This is the traditional approach, where we consider that all light curves were completely observed at the start of the analysis. Although this is not a realistic scenario (one cannot query, or spectroscopically observe, a SN that has already faded away), it gives us an upper limit on estimated classification results.

For each light curve and filter, all available data points were used to find the best-fit parameters of equation (6.1) following the procedure described in section 6.3. Best-fit values for different filters were concatenated to compose a line in the data matrix. In order to ensure the quality of fit, we considered only SNe with a minimum of 5 observed epochs in each filter; this reduced the size of our spectroscopic and photometric samples to 1094 and 20193 objects, respectively.

Sub-samples

The iterative framework presented above corresponds to the AL strategy for choosing the next object to be queried. In this description, we have 2 samples: labelled and unlabelled. In case we wish to quantify the performance of the ML model after each iteration, the recently re-trained model must be used to predict the classes of objects in a third sample –one that did not take part in the AL algorithm. Classification metrics are then calculated,

[12]For more detail on these approaches see Krause [2008], Chapter 5.

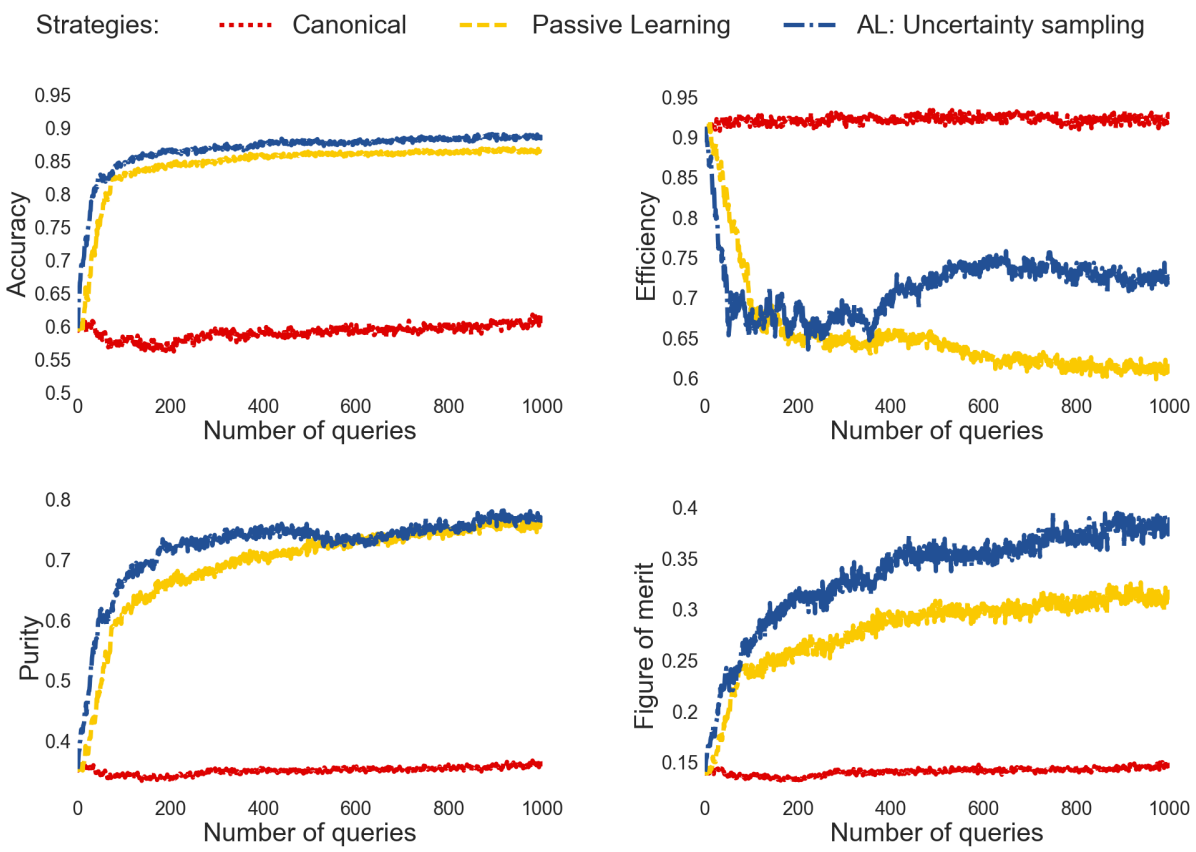

Figure 6.6: Evolution of classification results as a function of the number of queries for the static full light curve analysis.

after each iteration, from predictions on this independent sample. In this scenario we need 3 samples: training, query, and target. The *query sample* corresponds to the set of all objects *available for query* upon which the model evolves[13]. On the other hand, the *target sample* corresponds to the independent one over which diagnostics are computed. In the results presented in this sub-section, *SNPCC photo* was randomly divided into query (80%) and target (20%) samples.

Finally, we quantified the evolution of the classification results when new objects are added to the training sample according to the canonical spectroscopic follow-up strategy, by constructing a *pseudo-training* sample. For each element of *SNPCC spec*, we searched for its

[13]Not to be confused with the *set of queried objects*, which comprises the specific objects added to the training set (1 per iteration).

nearest neighbour in *SNPCC photo*[14]. This allowed us to construct a data set which follows very closely the distribution in the parameter space covered by the original *SNPCC spec*. Thus, randomly drawing elements to be queried from this *pseudo-training* sample is equivalent to feeding more data to the model according to the canonical spectroscopic follow-up strategy.

Results

In this section we present classification results for the static full light curve scenario according to three spectroscopic targeting strategies: canonical, passive learning and active learning via uncertainty sampling. In all three cases, at each iteration one object is queried and added to the training sample. We allow a total of 1000 queries, almost doubling the original training set.

Figure 6.6 shows how classification diagnostics evolve with the number of queries. The red inverse triangles describe results following the canonical strategy (random sampling from the pseudo-training sample), yellow circles show results from passive learning (random sampling from the query sample), and blue triangles represent results for AL via uncertainty sampling. We notice that the canonical spectroscopic targeting strategy does not add new information to the model – even if more labelled data is made available. Thus there is almost no change in diagnostic results after 1000 queries. On the other hand, the canonical strategy is very successful in identifying SN Ia (approximately 92% efficiency); however, by prioritizing bright events, it fails to provide the model with enough information about other SN types. Consequently, its performance in other diagnostics is poor ($\sim 60\%$ accuracy, 36% purity and a figure of merit of 0.15). At the same time, passive learning and AL via uncertainty sampling show very similar efficiency results up to 400 queries. Accuracy levels

[14]This calculation was performed in a 16 dimensions parameter space: type, redshift, simulated peak magnitude, and mean SNR in all 4 filters. For all the numerical features we used a standard euclidean distance.

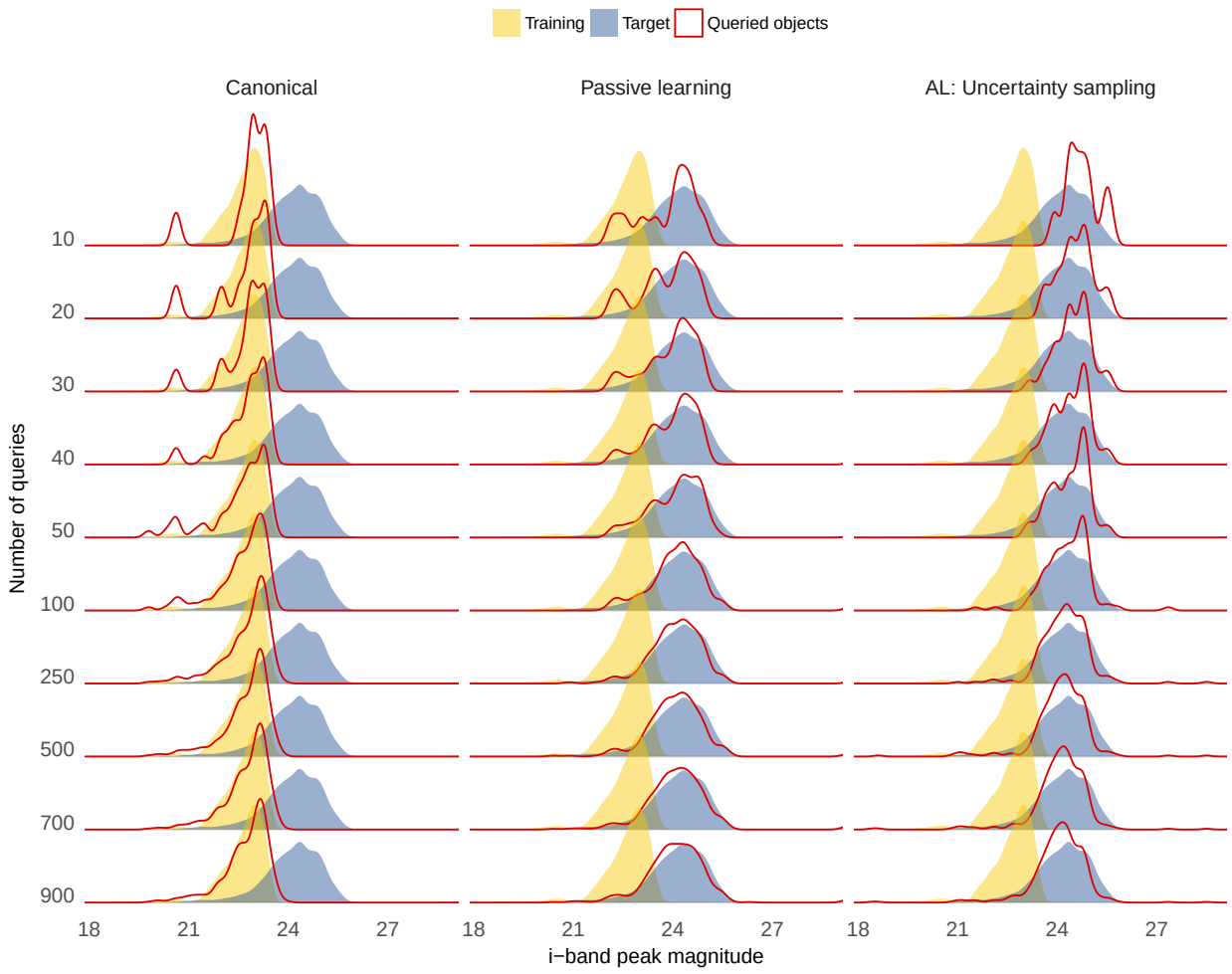

Figure 6.7: Simulated i-band peak magnitude distribution as a function of the number of queries for the static full light curve scenario. The yellow (blue) region shows distribution for the training (target) samples, while the red curves denote the composition of sample queried by AL. Lines go through 10 to 900 queries (from top to bottom). Different columns correspond to different learning strategies: canonical, passive and active learning via uncertainty sampling (from left to right).

stabilize quickly (84%/87% after only 200 queries), followed closely by purity results (73% after 600 queries). The biggest difference appears on efficiency levels. We can recognize an initial drop in efficiency up to 400 queries. This is expected, since both strategies prioritize the inclusion of non-Ia objects in the training sample: passive learning simply led by the higher percentages of non-Ia SNe in the target sample (figure 6.3), and AL by aiming at a more diverse information pool. This implies that high accuracy and purity levels are ac-

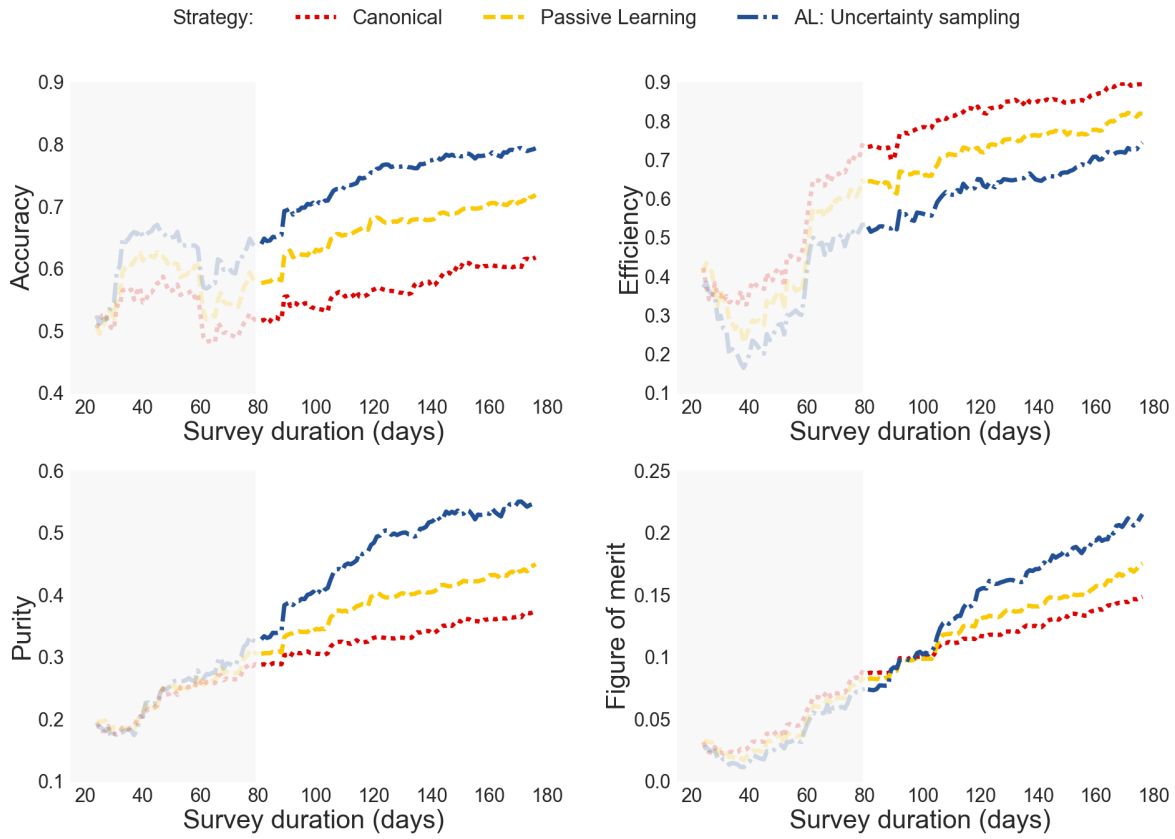

Figure 6.8: Evolution of the classification results as a function of the survey duration for the time-domain AL considering the SNPCC training set as completely given in the beginning of the survey.

companied by a decrease in efficiency (from 92% to 68% at 200 queries). After a minimally diverse sample is gathered, passive learning continues to lose efficiency, stabilizing at 63% after 700 queries, while AL is able to harvest further information to stabilize at 72% after 800 queries. Thus, after 1000 new objects were added to the training sample, passive learning achieves a figure of merit of 0.32 (2.1 times higher than canonical), while AL via uncertainty sampling achieves a figure of merit of 0.39 (2.6 times higher than canonical).

Figure 6.7 illustrates how the distribution of peak i-band magnitude in the set of queried elements evolves with the number of queries. For the sake of comparison, we also show the static distributions for the training (yellow) and target (blue) samples. As expected, the canonical strategy consistently follows the spectroscopic sample distribution. Meanwhile,

passive learning completely ignores the existence of the initial training –consequently, its initial queries overlap with regions already covered by the training sample, allocating a significant fraction of spectroscopic resources to obtain information already available in the training. The AL strategy, even in very early stages, takes into account the existence of the training sample, focusing its queries in the region not covered by training data (higher magnitudes). At 900 queries, the set of queried objects chosen by passive learning (red line, middle column) follows closely the distribution found in the target sample (blue), but this does not translate into a better classification because the bias present in the original training was not yet overcome. On the other hand, the discrepancy in distributions between the target sample (blue region) and the set of objects queried by AL (red line, right-most column) at 900 queries is a consequence of the existence of the initial training[15]. The fact that AL takes this into account is reflected in the classification results (figure 6.6).

These results provide evidence that AL algorithms are able to improve SN photometric classification results over canonical spectroscopic follow-up strategies, or even passive learning in a highly idealized environment[16]. However, in order to have a more realistic description of a SN survey, we need to take into account the transient nature of the SNe and the evolving aspect of an observational survey.

Although we chose to illustrate non-representativeness between samples in terms of peak brightness in different bands (e.g. figures 6.1, 6.7 and 6.12), these features are absent in the input data matrix. Our goal is to emphasize that the underlying astrophysical properties are tracked differently by the AL and passive learning strategies – even if these are not explicitly used.

[15]The reader should keep in mind that after 1000 queries the model is trained in a sample containing the complete SNPCC spectroscopic sample added to the set of queried objects.

[16]A result already pointed out by Gupta et al. [2016].

6.6 Real Time Analysis

In this section, we present an approach to deal with the time evolving aspect of spectroscopic follow-ups in SN surveys. This is done through the daily update of:

1. identification of objects allocated to query and target samples,
2. feature extraction and
3. model training.

We begin considering the full SNPCC spectroscopic sample completely observed at the beginning of the survey; this allows us to have an initial learning model. Then, at each observation day d, a given SN is included in the analysis if, until that moment, it has at least 5 observed epochs in each filter. If this first criterion is fulfilled, the object is designated as part of the *query sample* if its r-band magnitude is lower than or equal to 24 ($m_r \leq 24$ at d); otherwise, it is assigned to the target sample[17]. Figure 6.9 shows how the number of objects in the query (yellow circles) and target (blue triangles) samples evolves as a function of the number of observing days. Although the survey starts observing at day 1, we need to wait until day 20 in order to have at least 1 object with a minimum of 5 observed epochs in each filter. From then on, the query sample begins with 666 objects (at day 20) and shows a steady increase until it almost stabilizes $\sim$ 2100 SNe (around day 60). On the other hand, the target sample shows a steep increase until $d \sim 80$ (hereafter, *build-up phase*) and continuous to grow from there until the end of the survey, although at a lower rate. This behaviour is expected since, in this description, the *query sample* corresponds to the fraction of photometric objects satisfying the magnitude threshold ($m_r \leq 24$) at a specific time. Notice that as the survey evolves, an object whose detection happened in a very early phase will be

[17]We consider an object with r-band magnitude of 24 to have the minimum brightness necessary to allow spectroscopic observation with a 8-meter class telescope.

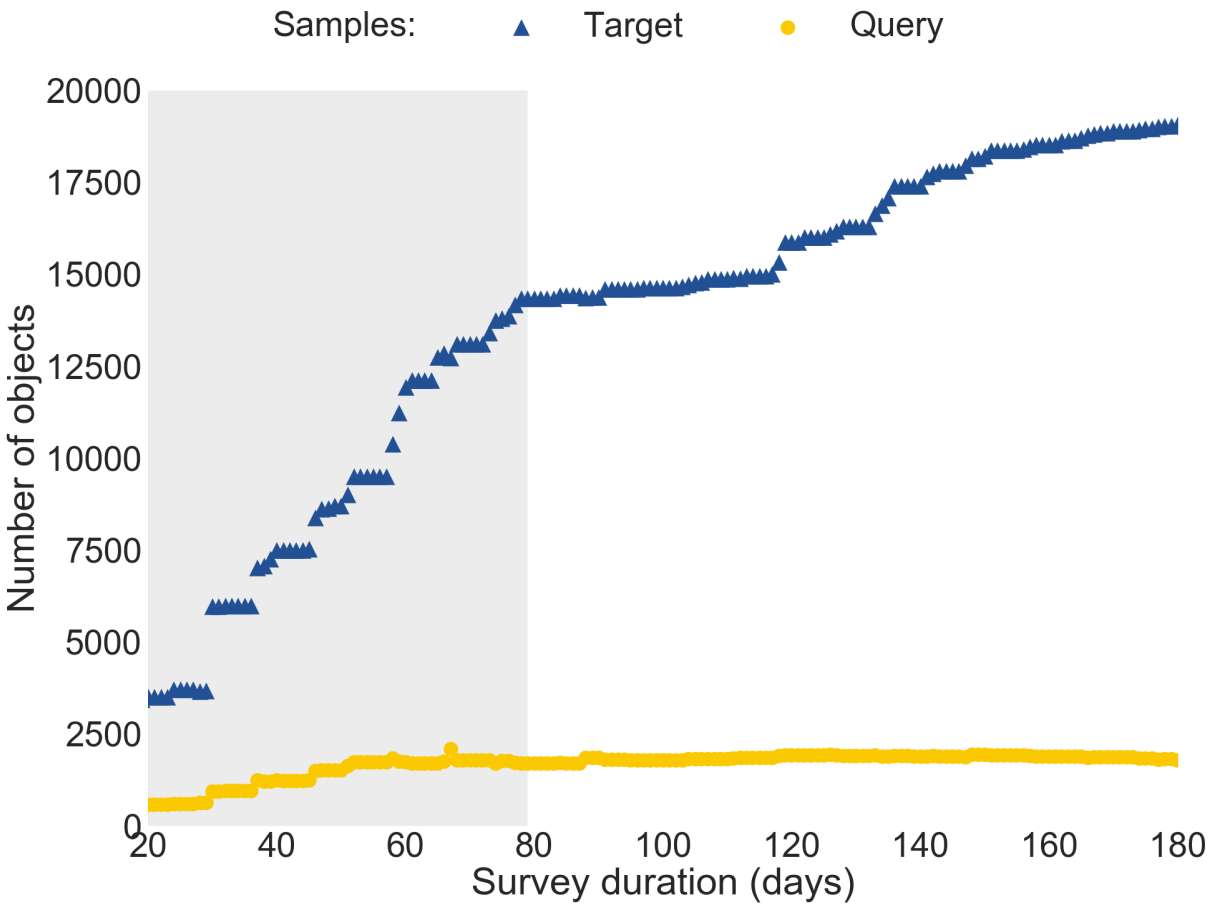

Figure 6.9: Number of objects in the query (yellow circles) and target (blue triangles) samples as a function of the days of survey duration. The grey region highlights the initial build-up phase of the survey, where there is a steep increase in the number of objects in the target sample.

assigned to the target sample during its rising period, but if its brightness increases enough to allow spectroscopic targeting it will move to the query sample, where it will remain for a few epochs. After its maximum passes, the SN will eventually return to the target sample as soon as it fades below the magnitude threshold, remaining there until the end of the survey. Thus, it is important to keep in mind that, despite its size being practically constant after the build-up phase, individual objects composing the query sample might not be the same for consecutive days.

The feature extraction process is also performed on a daily basis, considering only the epochs

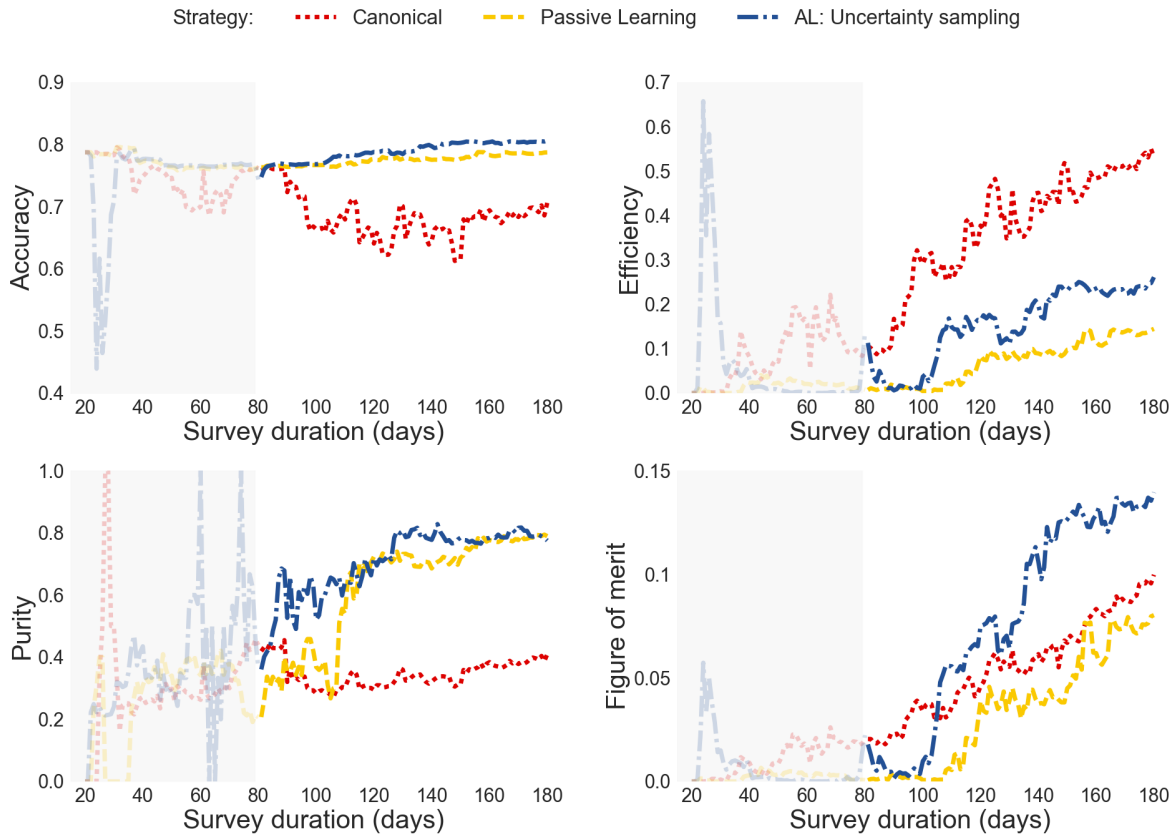

Figure 6.10: Evolution of classification results as a function of survey duration in the real time analysis, with a random initial training of 1 object.

measured until that day. This clarifies why we consider an analytical parametrization a simple, and efficient enough, feature extraction procedure. It reasonably fast and encompasses prior domain knowledge on light curve behaviour while returning the same number of parameters independently of the number of observed epochs. Moreover, it avoids the necessity of determining the time of maximum brightness or performing any type of epoch alignment [see e.g. Richards et al., 2012a, Ishida and de Souza, 2013, Revsbech et al., 2017]. Thus, we are able to update the feature extraction step as soon as a new epoch is observed and still construct a homogeneous and complete low-dimensionality data matrix. The only constraint is the number of observed epochs, which must be at least equal to the number of parameters in all filters.

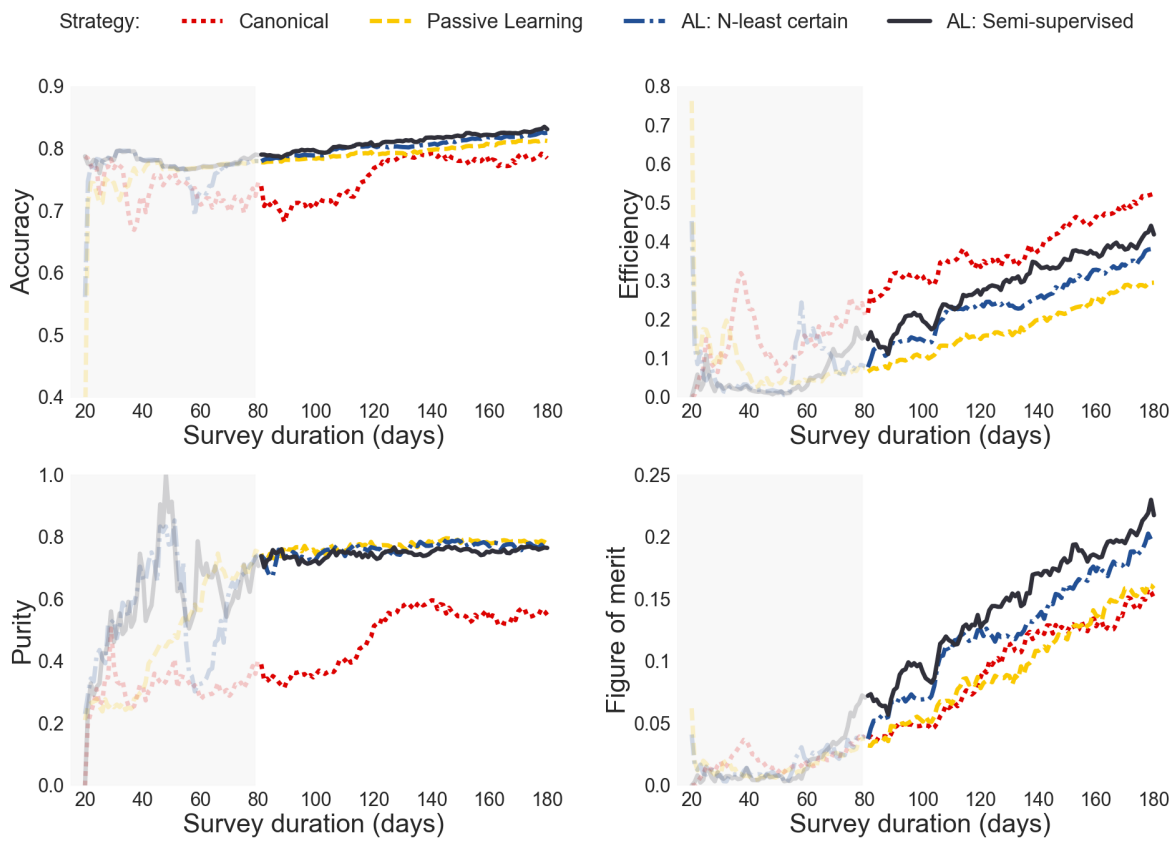

Figure 6.11: Evolution of classification results as a function of survey duration for the batch-mode real time analysis with $N = 5$ and a random initial training of 5 objects.

Finally, at the end of each night, the model is trained using the features and labels available until that point. The AL algorithm is allowed to query only the objects belonging to the *query sample*. Once a query is made, the targeted object and corresponding label are added to the training sample, the model is re-trained and the result applied to the target sample (figure 6.5). Given the time span of the SNPCC data, we are able to repeat this analysis for a total of 180 days.

Figure 6.8 shows the evolution of classification results considering the complete SNPCC spectroscopic sample as a starting point. Here we can clearly see the effect of the evolving sample sizes: accuracy and efficiency results oscillate, while purity and figure of merit remain indifferent to the learning strategy, during the build-up phase (grey region). Once this phase

is over, results start to differ and the AL with uncertainty sampling clearly surpasses the other two, achieving 80% accuracy, 55% purity and a figure of merit of 0.23, while the passive learning only goes up to 72% accuracy, 45% purity and figure of merit of 0.18. The canonical strategy continues to output better efficiency, but its loss in purity does not allow it to overcome even passive learning in figure of merit levels.

6.6.1 No initial training

This leaves one open question: what should we do at the beginning of a given survey, when a training set with the same instrument characteristics (e.g. photometric system) is not yet available? Or even more drastically: if the algorithm is capable of building its own training sample, do we even need an initial training at all? The answer is no.

Figure 6.10 shows how the classification results behave when the initial model is trained in 1 randomly selected object from the query sample, meaning we start with a random classifier. In this context, diagnostics are meaningless until around 100 days (a little after the build-up phase) when all samples involved are under construction. After this period, AL with uncertainty sampling starts to dominate purity and, consequently, figure of merit results. After 150 observation days (or after 130 objects were added to the training), the active and passive learning strategies achieve purity levels comparable to the one obtained in the unrealistic full light curve scenario ($\sim$ 80%). Thus, at the end of the survey, AL efficiency results (27%) are 80% higher than the one obtained by passive learning (15%), which translates into an almost doubled figure of merit (0.14 from AL and 0.08 from the canonical strategy). Compare these results with the *initial state* of the full light curve analysis: figure 6.6 (accuracy 60%, efficiency 92%, purity of 35% and 0.15 figure of merit) was obtained using complete light curves for all objects, all SNe in the original SNPCC spectroscopic sample surviving the minimum number of epochs cuts (1094 objects) and the same random forest classifier. Final results of the real-time AL analysis (figure 6.10)

surpasses the full light curve initial state accuracy results in 33%, more than doubles purity and achieves comparable figure of merit results. All of these while respecting the time evolution of observed epochs of only 161 SNe in the training set, or 15% the number of objects in the original SNPCC spectroscopic sample.

Accuracy levels of real time AL with (figure 6.8) and without (figure 6.10) the full initial training sample are comparable, while efficiency and figure of merit are higher for the former case. However, purity levels are 45% higher without using the initial training. This is a natural consequence of the higher number of SNe Ia in the SNPCC spectroscopic sample (figure 6.9), which requires the algorithm to unlearn the preference for Ia classifications before it can achieve its full potential in purity results. Figure 6.10 also shows that regarding purity, passive learning is able to achieve the same results as those obtained with uncertainty sampling while efficiency is severely compromised –exactly the opposite behaviour shown by the canonical strategy. This is a consequence of the populations targeted by each of these strategies. By prioritizing brighter objects, the canonical strategy introduces a bias in the learning model towards SNIa classifications. On the other hand, by randomly sampling from the target, passive learning adds a larger number of non-Ia examples to the training, introducing an opposite bias, at least in the early stages of the survey.

In summary, given the intrinsic bias present in all canonically obtained samples, we advocate that the best strategy for a new survey is to construct its own training during the firsts running seasons. Letting its own photometric sample guide the decisions of spectroscopic targeting. This is specially important if one has the final goal of supernova cosmology in mind, where the main objective is to maximize purity (minimize false positives) as well as many other scientific SN objectives.

6.7 Fixed Batch Analysis

In this section, we take another step towards a more realistic description of a spectroscopic follow-up scenario. Instead of choosing one SN at a time, spectroscopic follow-up resources for large scale surveys will probably allow a number of SNe to be spectroscopically observed per night. Thus, we need a strategy which allows us to extend the AL algorithm, optimizing our choice from one to a set (or a batch) of objects at each iteration. We focus on two methods derived from the notion of uncertainty sampling: *N-least certain* and *Semi-supervised uncertainty sampling*.

The N-least certain batch query strategy uses the same machinery described in the sequential uncertainty sampling method but, instead of choosing a single unlabelled example, it selects the N objects with highest uncertainties, and queries all of them. This tactic carries a disadvantage, since a set of objects whose predictions exhibit similar uncertainties will probably also be similar among themselves (i.e., will be close to each other in the feature space). Thus, querying for a set of labels is not likely to lead to a model much different than the one obtained by adding only the most uncertain object to the training set. In dealing with a batch mode scenario, we should also require that the elements of the batch be as diverse as possible (maximizing their distance in the feature space).

Semi-supervised uncertainty sampling [e.g. Hoi et al., 2008], in contrast, avoids the need to call the oracle at each individual iteration by using the uncertainty associated to each predicted label as a proxy for class assignment. The model must be trained in the available initial sample in order to create the first batch. The object with the greatest classification uncertainty is then identified. Suppose this object has a probability p of being SN Ia. A pseudo-label is then drawn from a Bernoulli distribution, where success is interpreted as "Ia" label (with probability p) and failure as "non-Ia" (with probability $1 - p$). The object features and corresponding pseudo-label are temporarily added to the training sample and

the model is re-trained. This is repeated until we reach the size of the batch.The benefit of using the model to produce pseudo-labels comes with the inevitable uncertainty attached to model predictions: they come unwarranted. However, the problem attached to the N-least certain strategy is here, to a certain degree, overcome. Similar unlabelled instances are less likely to be included in the same batch.

The optimum number of elements in each batch, N, is highly dependent on the particular combination of data set and classifier at hand. At each iteration, we are actually stretching the capabilities of the learning model in a feedback loop that cannot be expected to perform well for large batches. For the SNPCC data, our tests show that semi-supervised learning outperforms the N-least certain strategy for $N \in [2, 8]$ with maximum results obtained with $N = 5$.

Figure 6.11 shows classification results for canonical and passive learning (both at each iteration drawing 5 random elements from the pseudo-training and target sample respectively), AL via N-least certain and semi-supervised uncertainty sampling, when the initial training consists of 5 randomly drawn objects from the query sample and $N = 5$. We see that in this scenario semi-supervised AL is able to achieve the same figure of merit (~ 0.22) as sequential uncertainty sampling when the entire initial training sample is available (figure 6.8). However, it does so using only 63% of the number of objects for training (or 800 SNe in the training after 180 days, against 1263 SNe in the full training case). Moreover, although efficiency results show a steady increase until the end of the survey, purity achieves saturation levels (~ 0.8 – the same as the final results obtained with the static full light curve scenario, figure 6.6) after only 100 days (corresponding to a training set with 405 objects). A numerical description of the final classification results and corresponding training size is shown in table 6.1.

From figure 6.11 we see that samples containing the same number of objects lead to different

classification results. Moreover, considering that the query sample only contains objects with $m_r \leq 24$, we should not expect the set of objects queried by AL to be representative of the target sample, despite the improvement in classification results driven by AL. This is clearly shown in figure 6.12, where we compare distribution of maximum observed brightness in each filter for the SNPCC spectroscopic (red) and photometric (blue) samples with the set of objects queried by AL (dark grey). The latter provides a slight advantage in coverage when compared to the original spectroscopic sample, but it is still significantly different from the photometric distribution. A similar behaviour is found when we compare the populations of different SN types (figure 6.13) and redshift distribution (figure 6.14). These results confirm that, although a slight adjustment is necessary in order to optimize the allocation of spectroscopic time, a significant improvement in classification results may be achieved without a fully representative sample.

6.8 Telescope allocation time

As a final remark, we must address the question of how much spectroscopic telescope time is required to obtain the labels queried by the AL algorithm – in comparison to the time necessary to get all labels from SNPCC spectroscopic sample (canonical strategy). This will be more thoroughly addressed when present the experiments from Kennamer et al. [2020].

In the realistic case of a survey adoption of the framework proposed here, a term taking into account the telescope time needed for spectroscopy observations must be added to the cost function of the AL algorithm. This was not explicitly taken into account in our paper Ishida et al. [2019], but we considered a constraint on magnitudes for the set of SNe available for spectroscopic follow-up ($r_{mag} \leq 24$). We were able to estimate the integration time required for each object to achieve a given SNR by considering its magnitude and typical values

	static, full LC initial training UNC	time domain initial training UNC	time domain initial training BATCH 5	time domain BATCH 5
training size	2093	1255	1093	810
accuracy	0.89	0.80	0.85	0.83
efficiency	0.73	0.78	0.69	0.44
purity	0.78	0.55	0.69	0.76
figure of merit	0.39	0.23	0.31	0.22

Table 6.1: Classification results for the AL by uncertainty sampling (UNC) and semi-supervised batch mode (BATCH 5) strategies.

for statistical noise of the sources[18]. In the SNPCC spectroscopic sample, we considered the spectra taken at maximum brightness. For the set of AL queried objects, we used the magnitude at the epoch in which the object was queried. Considering a SNR of 10 (more than enough to enable classification) the ratio between the total spectroscopic time needed to get the labels for the SNPCC spectroscopic sample and the set of objects queried by semi-supervised AL is 0.9992. This indicates that the set of objects queried by AL would require less than 2.9s more time than the SNPCC spectroscopic sample to be observed at each hour. Also, if a more realistic estimation had been performed considering instrumental overheads, the set of objects queried by AL would have significant advantage, as it contains 26% less objects than the SNPCC spectroscopic sample. This gives us the first indication that AL-like approaches are feasible alternatives to minimize instrumental usage and, at the same time, optimize scientific outcome of photometrically classified samples.

For the specific case studied here, the high purity values achieved in early stages of the batch-mode AL, accompanied by the steady increase in efficiency (figure 6.11) renders our final SN Ia sample optimally suited for photometric classification in cosmological analysis –albeit being smaller in number of objects and requiring almost the same amount of spectroscopic resources to be secured.

[18]Namely, counts in the sky, ≈ 13.8 $e^-/s/pix$ and read-out noise, ≈ 8 e^-[e.g. Bolte, 2015].

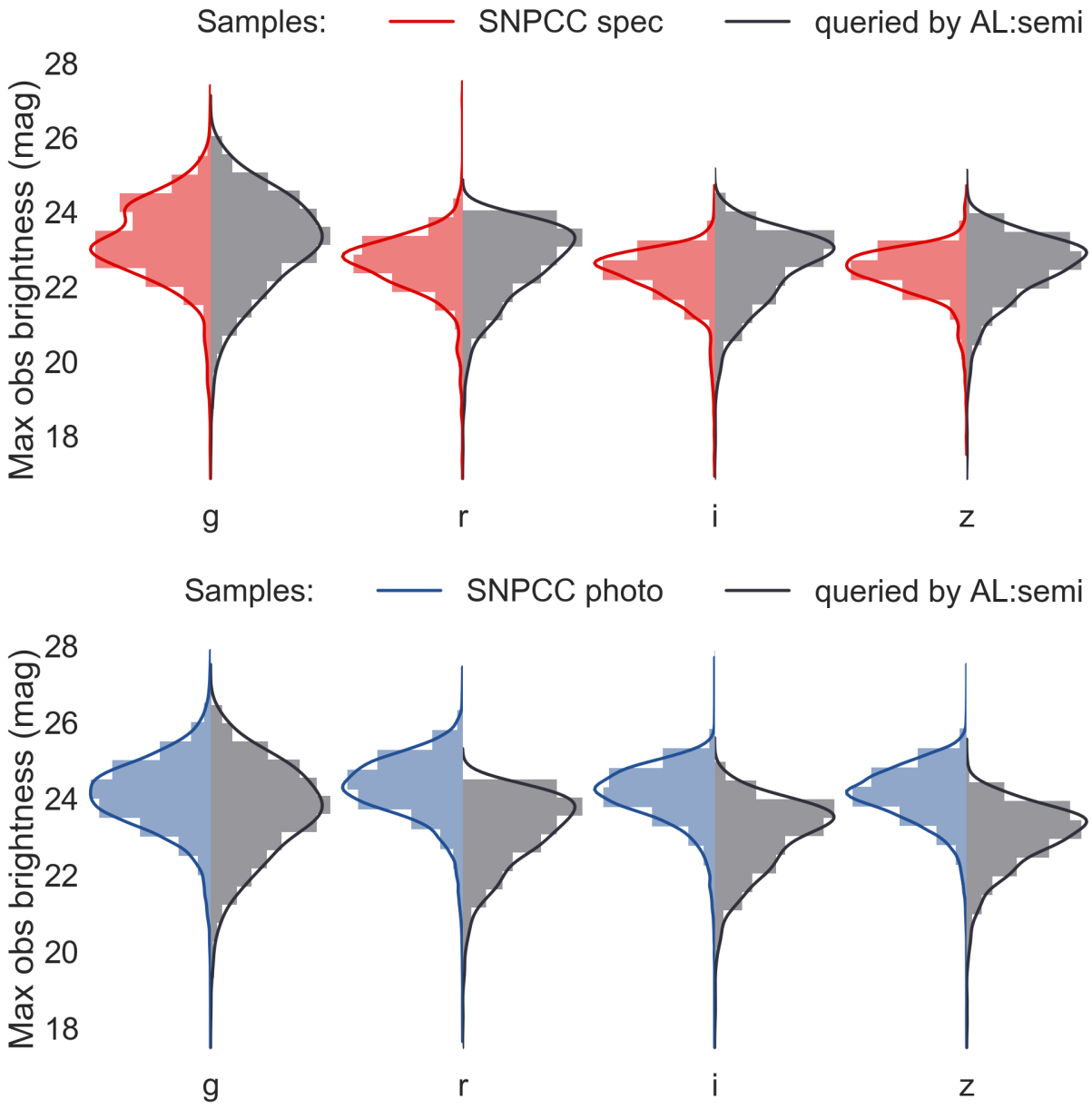

Figure 6.12: Distributions of maximum observed brightness, in all DES filters, for the set of objects queried by AL via batch-mode semi-supervised uncertainty sampling with $N = 5$ (dark grey). This is compared to distributions from SNPCC spectroscopic (red - top) and SNPCC photometric (blue - bottom) samples.

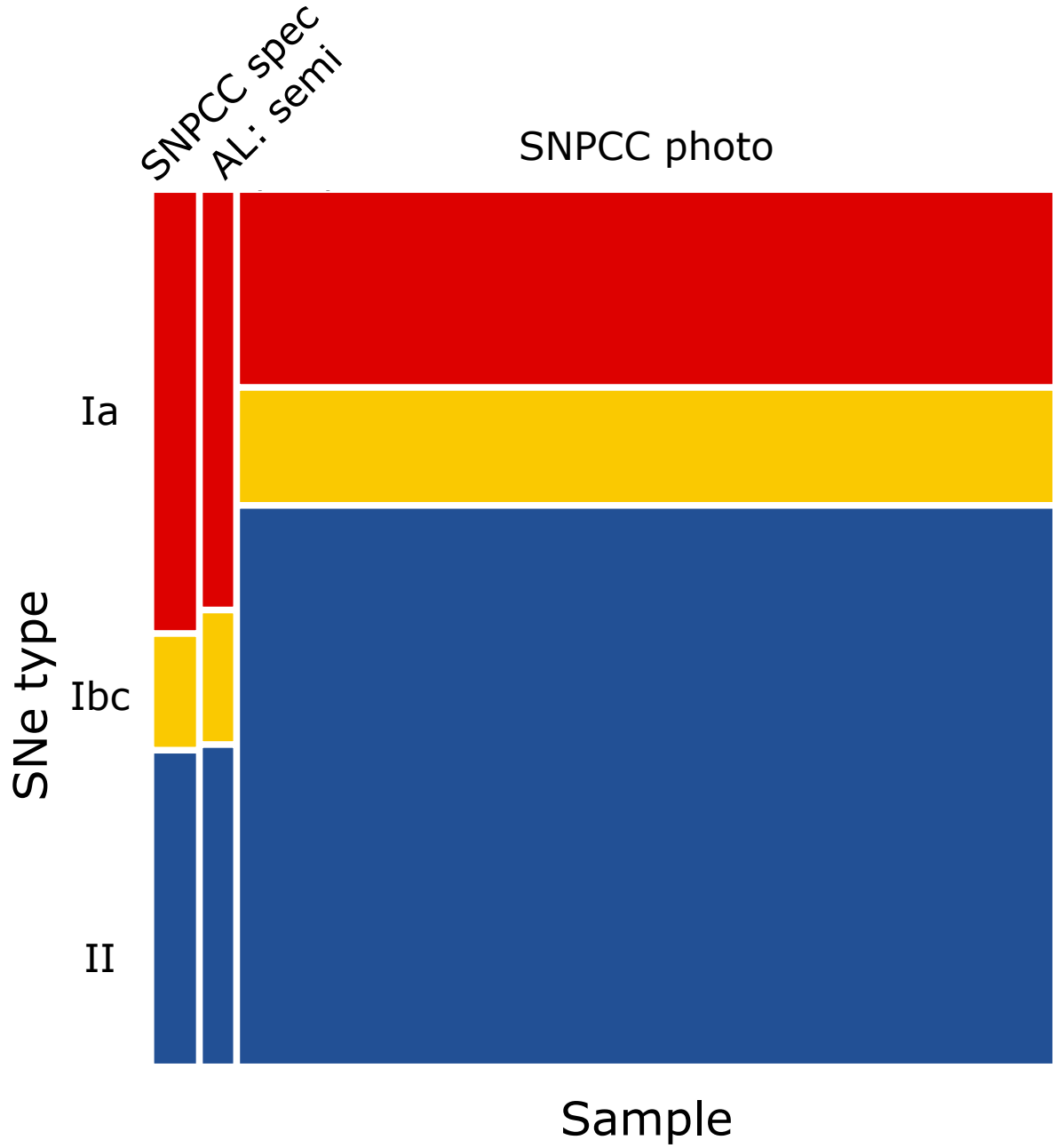

Figure 6.13: Populations of different supernova types in the original SNPCC spectroscopic and photometric samples, and in time domain batch mode ($N = 5$) semi-supervised AL query sample after 180 days of observations. The composition of the SNPCC samples are the same as shown in figure 6.3. The AL query sample holds 390 (48%) Ia, 122 (15%) Ibc and 298 (37%) II.

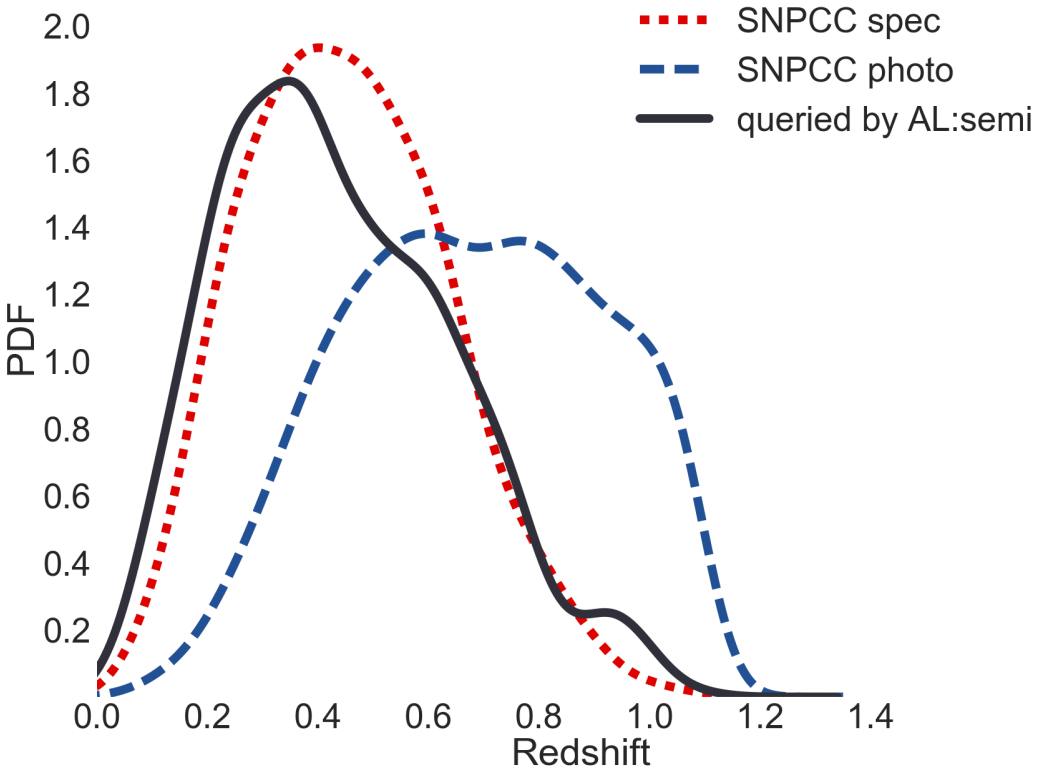

Figure 6.14: Redshift distribution of the original SNPCC spectroscopic (red - dotted) and photometric (blue - dashed) samples, superimposed to the redshift distribution of the AL query set for the time domain semi-supervised batch mode AL strategy without the use of an initial training (dark blue - full). In each observation night, the algorithm queried for 5 SNe. The distribution shows redshift for the query sample after 180 observation nights.
.

Based on the experimental results from Ishida et al. [2019] we can see that active learning is a promising technique for optimizing spectroscopic follow up for astronomical surveys. Following this work we endeavored to make our experiments much more realistic to real world constraints. Incorporating the constraints into our existing framework resulted in the work presented in Kennamer et al. [2020], which we now discuss.

6.9 Realistic Constraints Analysis

The task of classifying astronomical transients poses extra challenges beyond those faced by gathering different types of observations. We describe below some relevant issues that were considered in our experiments. Although this is not an exhaustive list, it is, to our knowledge, a more realistic description than any other found in the literature to date.

Labeling window of opportunity

Once a new source is identified as a supernova candidate, we expect its brightness to evolve and, eventually, fade away. Any spectroscopic analysis should ideally be performed when the transient is near its maximum brightness; this commonly leads to a more reliable, and less time consuming, spectroscopic confirmation. Moreover, distant or intrinsically faint targets may only be bright enough to allow spectroscopic measurements close to maximum brightness, which imposes a small time window during which labeling is possible (typically a few days). Additionally, the decision of labeling one particular target needs to be made with partial information – when one has seen only a few points in the light curve.

Evolving samples

In adapting the supernova classification problem to a traditional machine learning task, we build the initial training and validation/test samples using full-light curves. Our goal is to use active learning to construct a model that performs well when classifying the full light curve test sample. However, the pool sample unavoidably contains partial light curves (Section 6.9). Considering, for the moment, a simplified case of fixed batches containing only 1 object: at each iteration an object is queried and sent for spectroscopic observation. Assuming the labeling process was successful, the chosen object is likely to be close to

its maximum brightness phase. As a consequence, its light curve has only been partially observed. This partial light curve and its corresponding label are transferred from the pool to the training sample, which is now formed by a number of full light curve objects and one additional partial light curve. Since we expect the following day to bring some additional photometric measurements (points in the light curve) for a subset of the objects in the initial pool sample, the result is a continuous update and evolution of the training and pool samples during the entire duration of the survey.

Sources of budget

In our case study, the labeling process is extremely expensive and requires coordination between different telescopes. The power of astronomical telescopes is proportional to the area of their primary mirror. A larger primary mirror means the telescope is able to target fainter, and consequently more distant, sources. We consider the scenario where two spectroscopic telescopes are used for labeling purposes: one telescope with a primary mirror of 4m in diameter and another with 8m. At each night, we considered 6 hours of available observation time per telescope[19] (budget). Since spectroscopic observations of the same object require a different amount of observation time for each of the telescopes, each telescope is considered a distinct budget source.

Evolving costs per object and budget source

For each queried object, the time necessary to take a spectrum (which in turn can be used for labeling) depends on the characteristics of the available spectroscopic telescope and the brightness of the target object, among other factors. As an illustration, an object with a brightness that requires t minutes of spectroscopic analysis using a 4m telescope is also a viable target for the 8m – in which case it would require only a fraction of t to complete the

[19]This is an optimistic estimation of the nightly budget.

observation. On the other hand, a fainter object which can be observed by the 8m telescope given a large enough observation time, might not be a viable target for the 4m. Moreover, as the brightness (measured flux) of each supernova evolves with time, this cost will also depend on the time the query is made. In our case, we update the cost of each queried object for the two different sources of budget (telescopes) at each active learning iteration (night). The maximum allowed observation time for any given object is set to 2 hours. Our exposure time calculator is heavily based on Förster et al. [2016], developed for the High Cadence Transient Survey (HiTS).

6.9.1 Experiment design

We separated our data set into 3 groups: the full training sample, identified as spectroscopically confirmed by the SNPCC data set and formed by 1,103 objects (hereafter, original training); the validation and test samples formed by 1,000 objects each, taken from the 20,216 light curves tagged as purely photometric by the SNPCC data set and following its sub-population distribution; and the pool sample comprising the remaining 18,216 objects.

Since our pre-processing step (Section 6.9.2) requires a minimum of 5 observed points in each filter to deliver meaningful best-fit parameters, a complete input data matrix is only available starting from the 20^{th} day of the survey. This leaves only 160 active learning cycles (days) that we can use to build an optimal training sample. In order to probe the impact of the biases present in current spectroscopic samples, we also considered the situation where the initial training set is formed by only 10 objects (5 SNe Ia and 5 non-Ia) randomly chosen from the original training. This experimental configuration is also a more direct test of our active learning algorithms given that we have limited data and can only simulate the process for a small number of days.

To establish a baseline for comparison of our results, we also created a randomly sampled

training set which follows closely the distribution of the validation/test sample. Results obtained when using this sample to train our learning model correspond to the best possible scenario we can achieve given our data set, labeling budget and classifier combination. The entire SNPCC data was rearranged to build this set of randomly selected training, test and validation samples (each containing 1,000 objects). The remaining objects were then allocated to a pool sample. This configuration was used to provide an upper bound to the performance.

6.9.2 Pre-processing

We followed the feature extraction procedure described in the previous experiments. All light curves with at least 5 flux observations in each filter, were fit to the parametric function suggested by Bazin et al. [2009a] in equation (6.1).

The fit was performed independently for each filter. Objects with less than 5 observed points per filter or for which the parametric fit did not converge were not included in the analysis. Best fit parameter values for $p_X = \{A,\ B,\ t_0,\ \tau_f,\ \tau_r\}$ were concatenated according to the effective wavelength of its corresponding filter, $X = [g, r, i, z]$, to form one line of the input matrix per object.

Since the initial training, validation and test samples contain full light curves, their distribution does not change. Figure 6.15 shows the distribution of best-fit parameters in r-band for 3 of the features considering the original training, validation and test samples.

For the initial pool sample the number of points observed in each light curve changes with time, thus for each day we performed the feature extraction procedure considering all light curve points observed until then. To calculate the cost of labeling, we need to estimate the brightness of the object in each day of the survey. If the last observed light curve point was measured within the last 2 days, we used that measurement as a good estimate of its current

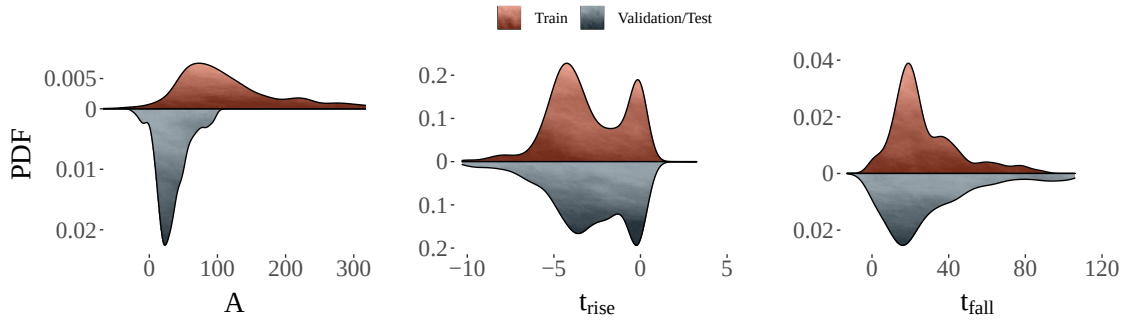

Figure 6.15: Comparison between light-curve features extracted from the original training (top orange) and validation/test (bottom grey) samples in DES r-band.

brightness. Otherwise, we use the result of the parametric fit to estimate its brightness today and use this estimate to calculate the cost of labeling with both telescopes (4m and 8m), as described in Section 6.9. Objects bright enough to be queried by at least one of the two available telescopes form the pool sample for that day. For the 3 example features, Figure 6.16 shows how the distribution of the complete pool sample (orange) changes with the evolution of the survey in comparison with the static validation/test samples (gray) in r-band.

6.9.3 Methodology

Once the training, pool, validation and test samples were properly set up (Sections 6.9.1 and 6.9.2), we recorded the performance of different active learning strategies using Random Forests Breiman [2001]. For the purpose of this work, we will only consider a binary classification problem (SN Ia/non-Ia). For all the experiments described in Section 6.9.1, we applied a naive Random Sampling (RS) strategy, where objects were randomly chosen from the pool without any selection criteria. This will serve as a lower bound for comparison with active learning techniques.

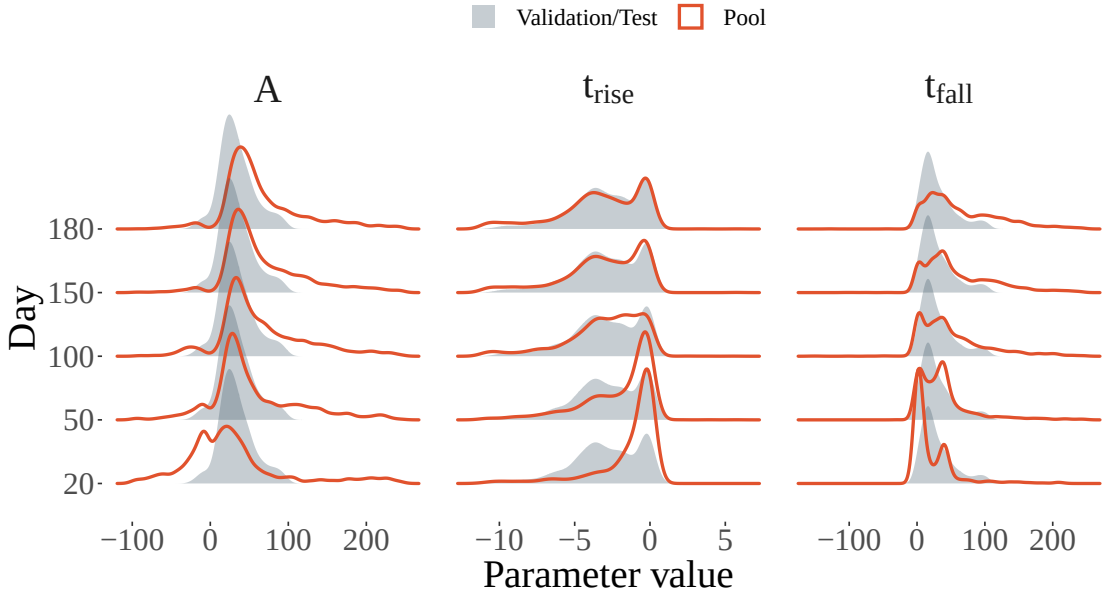

Figure 6.16: Distribution in feature space for the validation/test (filled grey area, fixed) and complete pool (solid orange line, evolving) samples as a function of the days since the beginning of the survey (20 to 180, from bottom to top). All distributions correspond to features extracted from r-band measurements.

Table 6.2: Performance metrics for the different active learning strategies when the entire SNPCC spectroscopic (training, 1103 objects) sample is given at the beginning of the survey. The table show results metric values 180 days after the start of the survey.

Metric	Learning Strategy			
	RS	BatchEntropy	BatchKL	USE
Accuracy	0.87	0.87	0.87	0.88
Efficiency	0.57	0.59	0.56	0.66
Purity	0.78	0.77	0.82	0.77
Figure of Merit	0.31	0.32	0.34	**0.35**

6.9.4 Results

For our first experiment we started from the idealized case of a randomly sampled training, validation and test samples, each containing 1,000 objects. The goal of this exercise was to quantify a set of optimal results given our data, classifier and labeling resources. We used a RS strategy for the entire duration of the survey. After 160 iterations (180 days of observation), we obtained {acc, eff, pur, FoM} = [0.88, 0.62, 0.82, 0.37].

Table 6.3: Performance metrics for the different active learning strategies beginning from a random initial training sample of 10 objects (5 SNe Ia, 5 non-Ias). The table shows results 180 days after the start of the survey.

Metric	Learning Strategy			
	RS	BatchEntropy	BatchKL	USE
Accuracy	0.85	0.87	0.87	0.87
Efficiency	0.42	0.54	0.50	0.55
Purity	0.85	0.84	0.83	0.80
Figure of Merit	0.27	**0.34**	0.31	0.32

We then considered the case where the original SNPCC spectroscopic sample was completely available at the beginning of the survey, thus starting with a training sample of 1,103 objects. We applied RS, BatchEntropy, BatchKL and USE strategies and ran them through all available observation days. The behavior of the diagnostic metrics as a function of the number of active learning iterations (days since the beginning of the survey) is shown in Figure 6.17 (left column). Numerical values for the final state of these models are reported in Table 6.2. After 160 iterations, the final training sample had grown by $\approx$ 1800 objects (for a total of $\approx$ 2900). Observing the behavior of different strategies in Figure 6.17 (left column), we see an improvement in all metrics. However, the difference in FoM results between RS and the best performing active learning strategy (USE) is merely $\approx$ 13% (0.04); active learning strategies struggle to outperform RS.

In order to test if this behavior is derived from the biases known to exist in the original training, we applied the same learning strategies to the case where the initial training sample is composed of only 10 objects randomly chosen from the original SNPCC spectroscopic sample (5 SNe Ia and 5 non-Ia). The evolution of all metrics is shown in Figure 6.17 (right column) and numerical values for their final state are given in Table 6.3. In this scenario, the initial classifier does not contain much information; accuracy, purity and FoM start with lower values. Nevertheless, they quickly improve with each iteration, achieving results as good as those obtained in the previous case. At the final stage, the training samples

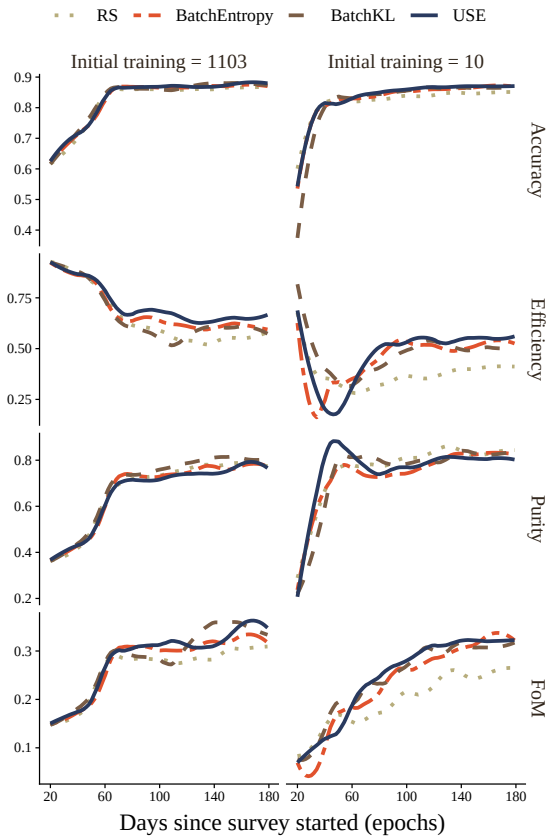

Figure 6.17: Evolution of different performance metrics as a function of the number of days since survey started (active learning iterations) for different learning strategies. All strategies shown here considered non-constant costs. **Left**: initial training corresponding to the original SNPCC spectroscopic sample. **Right**: initial training containing only 10 objects (5 SNe-Ia, 5 SNe-nonIa) randomly chosen from the original SNPCC spectroscopic sample.

contain $\approx$ 1,810 objects. Since a small initial training is less biased and more sensitive to the addition of new data, the active learning strategies clearly outperforms RS. The best performing active learning strategy (BatchEntropy) achieved a FoM of 0.34, while RS delivered a FoM of 0.27, a difference of $\approx$ 26% (0.07) and an increase of 75% when compared to the difference between USE and RS in the previous case case (0.04). This increase comes from a 28% increase in efficiency delivered by BatchEntropy over RS. Figure 6.18 shows the evolution in feature space of the samples queried by RS and BatchEntropy in comparison to the validation/test samples. Comparing Figures 6.16 and 6.18 it is clear that both strategies (RS and BatchEntropy) evolve the queried sample towards the validation set but subjected

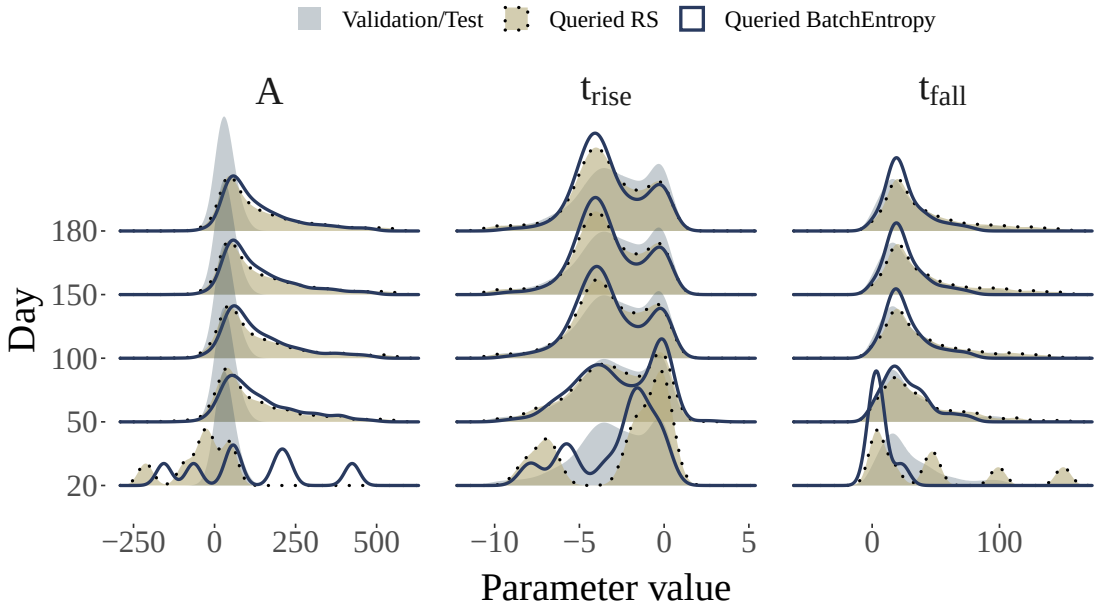

Figure 6.18: Distribution in r-band feature space for 3 different samples: Validation (filled grey), Random Sampling (dotted filled brown), and BatchEntropy (solid dark blue line) as a function of days since the start of the survey (20 to 180, from bottom to top). This results correspond to an initial training of 10 objects.

to the constraints of the available pool sample at each iteration.

6.10 Conclusion

Active learning strategies are promising techniques used to construct optimal training samples given scarce labeling resources. Nevertheless, stress tests probing their robustness under realistic conditions are largely missing in the literature. In many real world situations, the assumptions of sample representativeness or stability between samples are hard to meet, though the necessity to optimize the allocation of labeling resources is paramount.

In this work we focus on the classification of a subclass of extragalactic astronomical transients: supernovae. While this issue has received great attention in the last decade Kessler et al. [2010], Ishida and de Souza [2013], Lochner et al. [2016b], Ishida [2019], Möller and de Boissière [2020], the community is still far from developing a completely automated system

able to optimize the allocation of spectroscopic follow-up resources. In this work, we build upon the efforts reported in Ishida et al. [2019], presented in sections 6.5-6.7, and present for the first time a simulated data environment which simultaneously takes into account: 1) the necessity to estimate the current brightness of an object in order to make a decision about spectroscopic follow-up (only partial information is available at the time of labeling), 2) the evolution of training and pool samples with time, 3) the spectroscopic time required to observe each object in 2 different telescopes as a function of time (different labeling costs per day, object and budget source) and 4) the limited telescope time available per night (knapsack constraints).

We tested the performance of random sampling (RS) as well as three batch active learning strategies based on uncertainty sampling. When using the original training sample provided within the SNPCC data set (1,103 objects) as a starting point, active learning strategies did not significantly improve upon RS. This is a direct consequence of the biases known to exist between spectroscopic and photometric samples, combined with the large size of the initial training set, and the limited number of available nights (active learning iterations). Given these constraints, we constructed a second data scenario with a very small initial training set (10 objects). This initial state contained a negligible amount of information, but it was unbiased and highly sensitive to additional samples. Here, all active learning strategies clearly outperformed RS results. The best strategy (BatchEntropy) improved by 26% the results delivered by RS. The small initial training set achieved the same figure of merit using 1,093 fewer spectroscopically confirmed light curves (labels).

Such results emphasize the importance of planning, in advance, the construction of training samples for machine learning applications. By delegating the complete construction of the training sample to the active learning algorithm, we can ensure optimal classification results and obviate the use of legacy data or the need to model discrepancies between traditional spectroscopic and photometric samples.

Moreover, we showed that active learning strategies are robust in the presence of complex and realistic constraints on data collection. However, the fact that different batch strategies presented similar behavior indicates that our current techniques for acquiring diverse committees can be improved. This is an important issue which will be addressed in a future work.

Finally, we recognize that the scenario presented in this work is still incomplete. We failed to take into account issues such as: uncertainties due to our feature extraction method and the extrapolated brightness used to calculate the cost of each observation[20]; the probability that a labeling request is not fulfilled or that it may be incorrect; the impact of the resulting classifications in further scientific results and observational effects like airmass (position of a given source in the sky) and weather conditions (e.g. seeing, cloud cover). This complex environment makes the classification of transient astronomical sources an excellent test bench for developing learning algorithms. These are all crucial issues which will shape the scientific results from the next generation of large scale astronomical surveys and, consequently, our understanding of the Universe.

[20]This can be generalized as uncertainty in our data points, which has been studied mainly from a theoretical perspective with very ideal types of classifiers and noise models Nowak [2009, 2011].

Chapter 7

ContextNet: Deep Learning for Star Galaxy Classification

In this final chapter we discuss a standalone project to build a star vs. galaxy classification model, which could be used in astronomical pipelines. As we discuss in the sequel, for ground-based observations this involves designing a model capable of reasoning over a spatially and temporally varying confounding factor. We treat a single exposure as a set of images with one object per image and an arbitrary number of objects within the set. Building a classifier to operate over this set connects directly to work in preceding chapters of this thesis, in which we construct a variational posterior distribution conditioned on a set of objects with permutation invariance. In this setting, we do not require permutation invariance, but there remains a common structure between the two architectures. Both models consist of an encoder network that is applied to all objects within the set; an aggregation function, which in previous chapters was simply a sum function, but in this chapter takes the form of another neural network which creates an aggregated representation for all objects within the set; and finally an emitter function which in the preceding chapter was only applied to the aggregated representation, but in this chapter is applied to both the aggregated representation and the individual representation of each object, to produce a distinct classification decision for each object within the set. Both structures exploit the concept of weight sharing to reduce

computation; however, as noted the intermediary network in the model of this chapter allows us to reason over a confounding factor that is necessary to achieve accurate results.

This chapter is based on our published work in Kennamer et al. [2018], and the author would like to thank his collaborators David Kirkby, Javier Sánchez and Alex Ihler.

7.1 Introduction

The Large Synoptic Survey Telescope (LSST) is a ground-based photometric survey that will commence in early 2022 and will observe for 10 years. It will image the entire available sky every three nights and produce over one petabyte of data a year. The science goals of the survey are vast, ranging from the detection of dark energy and dark matter signatures to mapping small objects in the Solar System such as near-Earth asteroids LSST Science Collaboration et al. [2009]. The scale of the data and the types of measurements require sophisticated data analysis methods and present a great opportunity for machine learning scientists to work with domain scientists on fundamental science questions. Conversely, the astronomical data that is collected creates new challenges for the machine learning community and promotes the development of new methods. Collaborations between machine learning scientists and astronomers have been steadily growing and diverse in methods and applications, including a deep learning approach for analyzing strongly lensed systems Hezaveh et al. [2017], a probabilistic graphical model for processing astronomical images Regier et al. [2015], an ensemble approach for classification of supernova Lochner et al. [2016a], and many others. These projects have contributed to both fields, which is also the aim for our work. In this chapter we present the specific application of star galaxy classification, for which we develop a novel framework for composing neural network models in order to make advances in the field of astronomy.

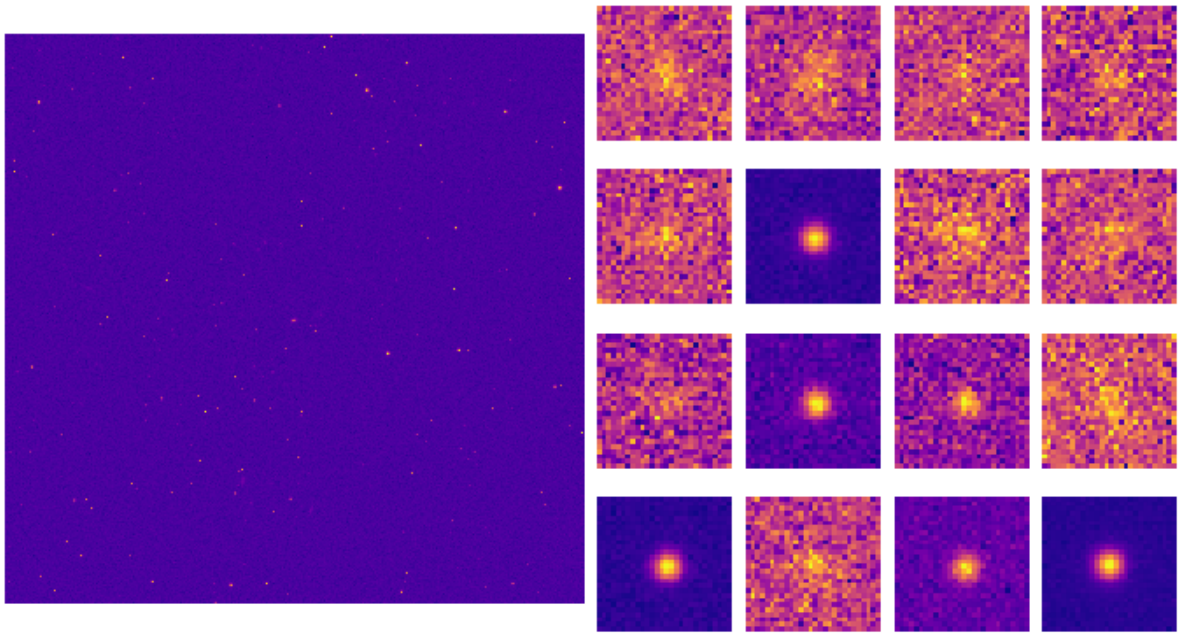

Figure 7.1: Left is an example of a single exposure with pixel values plotted on a log scale. To the right are 16 cutout images of detected sources in this exposure. The entire exposure has 1273 detected sources.

Star galaxy classification is one of the first processing steps in the data analysis pipeline of any astronomical survey; its foundational nature means that it affects almost every subsequent step of the pipeline Jurić et al. [2015]. The inputs to star galaxy classification are a collection of small, cutout images of detected sources in a single exposure, and the outputs are the predictions for each detected source in that exposure. A single exposure records the photon counts of each pixel of a CCD exposed for a short period of time. For LSST, each exposure is a 16 megapixel grid with an exposure time of 15 seconds. The entire telescope is made up of 189 of these CCD detectors Kahn et al. [2010]. It is expected that approximately 1200 sources will be detected and thus need to be classified for each exposure. Figure 7.1 shows a single exposure on a log scale with 16 representative cutouts to the right. Note that this exposure contains 1273 detected objects.

In an ideal world for astronomers (one without an atmosphere), this is a rather simple

problem. Stars can be thought of as point sources of light, while galaxies have some spatial distribution. Thus in order to make an accurate prediction all one has to do is measure the size of a source and, if it is greater than a certain threshold, it can be safely classified as a galaxy. However with a turbulent atmosphere and imperfect optics, light is spread out before it reaches the detector. This spreading function is often called a point spread function and it changes both spatially and temporally. Thus one cannot make predictions assuming that the sources are IID across exposures. Instead one must use a model that that is capable of handling the confounding factor of the point spread function (PSF) in order to make accurate predictions. Figure 7.2 shows an example star and an example galaxy with and without the PSF.

In the last several years there have been tremendous advances in computer vision, especially regarding the use of convolutional networks for classification Krizhevsky et al. [2012]. One of the main motivators of this work is to take advantage of this progress by applying it to the field of astronomy. However this cannot be done without significant modification. Section 7.2 will show how this problem is currently being solved by astronomers and why current vision models from machine learning need to be extended in order to tackle this problem. Section 7.3 details the framework we have developed for composing neural network models to predict on non IID instances and to predict simultaneously for a variable number of instances. Our empirical studies are presented in Section 7.4, showing how our model is able to achieve better results than what is currently used in astronomy. We present our results on simulations of LSST observations using the GalSim image simulation package Rowe et al. [2015], which was designed and developed by a large group of domain scientists. GalSim is designed to meet the stringent requirements of high precision image analysis applications such as weak gravitational lensing, for current datasets and for future astronomical surveys including LSST. In Section 7.5 we discuss future work, focusing on how our compositional framework can be further generalized to handle even greater diversity in inputs.

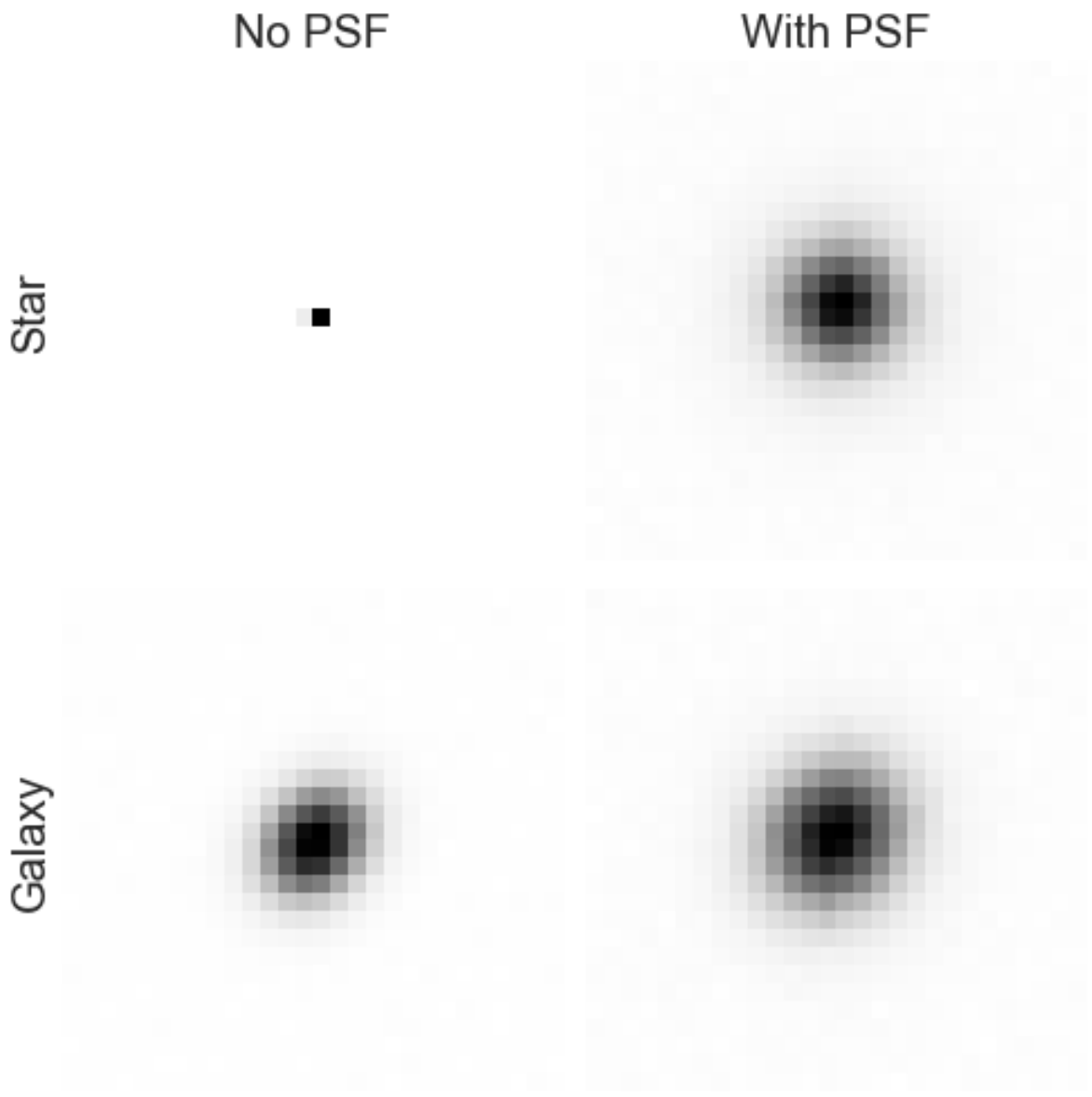

Figure 7.2: Example images of a star and galaxy with and without the point spread function (PSF). The first row shows a star and the second row shows a galaxy. The first column is the object without a PSF applied and the second column shows the object with the PSF applied. As we can see Without the PSF a star is a point like source of light but becomes spatially distributed when the PSF is applied. Thus making the problem of discriminating between stars and galaxies more challenging. Note the difference in noise level between the star and galaxy has to do with the galaxy being sampled at a dimmer magnitude. Stars at similar magnitude will have a similar noise level.

7.2 Related Work

In this section we describe how astronomers currently solve star galaxy classification, and why machine learning models that assume independence between cutouts are inadequate.

7.2.1 Measuring Extendedness

When classifying a cutout as a star or galaxy, the most important signal is whether the object is "extended" or not. Objects that are unextended are thought to be stars (point-like sources of light), while objects that are extended are galaxies. Astronomers have several different methods of measuring extendedness, but as shown by Garmilla [2016], they all achieve similar performance and are closely related. Thus we focus on only one of these related techniques. To measure the extendedness astronomers fit two models to a single cutout object: one for the point spread function and the second a galaxy model. Both of these fits are based on a discrete set of templates and parameterized profiles, a statistical test is performed to identify the best fit for both models. From these two models we can than separately measure, Mag_{psf}, the magnitude of the object weighted by the fitted model of the point spread function and, $Mag_{g_{model}}$, the magnitude measured from the galaxy fitted model. Note the magnitude of an object is a log measure of the brightness of the object: smaller magnitudes means brighter objects and larger magnitudes mean dimmer objects (inverse scale). We can then measure the extendedness of a single object as $Mag_{psf} - Mag_{g_{model}}$. Typical numbers for magnitudes for modern sky surveys are 15-25. Intuitively, a star being a point like source of light, the only process responsible for spreading out its light as it travels through the atmosphere is the PSF. Thus the PSF will be the best fitted model and $Mag_{psf} - Mag_{g_{model}}$ should measure zero for a star. On the other hand, for galaxies $Mag_{psf} - Mag_{g_{model}}$ should differ from zero, since galaxies have an inherent size and the best fitted model will not be Mag_{psf} Garmilla [2016]. An example of a typical size magnitude plot for a single exposure can be seen in Figure 7.3.

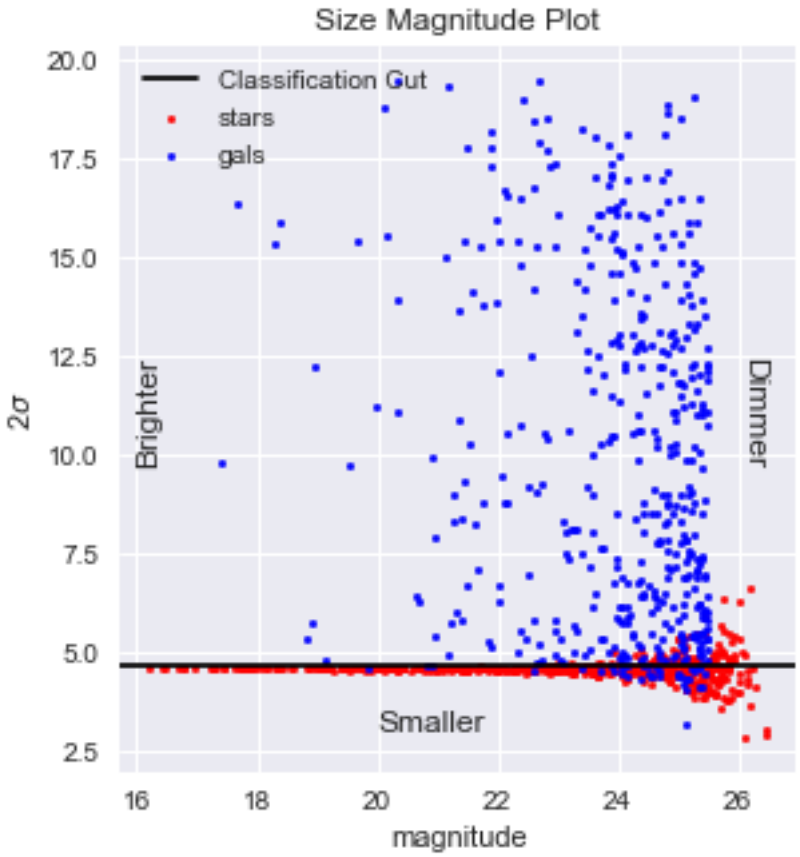

Figure 7.3: A size magnitude plot for a single exposure. In this figure we illustrate how star galaxy classification is typically solved by measuring the size and magnitude of each object in an exposure and making a classification cut on the size. The blue dots indicate galaxies and the red dots stars. We use a popular method used by the Dark Energy Survey Jarvis et al. [2016] to measure size, originally developed by Hirata and Seljak [2003] and improved by Mandelbaum et al. [2012]. A representative classification cut is shown by the black horizontal line . As we can see, this method achieves strong performance for bright objects, but the performance degrades for dim objects.

As this example shows, this method does very well for bright objects (low magnitude), but the star predictions are highly contaminated by galaxies for lower magnitudes.

Astronomers have used the preceding argument to create classifiers based on thresholds of extendedness to determine if an object is a star or galaxy. This method was used for the Sloan Digital Sky Survey (SDSS) Lupton et al. [2001] and has also been adopted by subsequent

Table 7.1: Models like AlexNet achieve excellent performance on standard image classification datasets. However, these models are not well suited to handle non-IID data. This table shows the performance of an AlexNet-like architecture for star galaxy classification. If all objects are blurred with the same PSF, the model does remarkably well, and only degrades slightly when the PSF changes spatially but is held constant across exposures (temporal variation). However, if the PSF varies between exposures the model fails completely, even when the amount of training data is doubled.

PSF Spatial Variation	PSF Temporal Variation	Training Size	Accuracy
Constant	Constant	5000 exposures	0.97
Varying	Constant	5000 exposures	0.92
Varying	Varying	5000 exposures	0.49
Varying	Varying	10000 exposures	0.51

surveys including Hyper Suprime-CAM (HSC) Bosch et al. [2018], Dark Energy Survey (DES) Jarvis et al. [2016] and is intended to be used for LSST Jurić et al. [2015]. However as sky surveys push deeper into space, gathering data on dimmer objects, this method becomes less effective and is unable to distinguish dimmer objects Garmilla [2016]. Thus there is a significant need for new methods that are not just better overall, but well suited to dimmer objects. The method we will discuss in Section 7.3 achieves better performance both overall and specifically on dimmer objects, as shown in Section 7.4.

7.2.2 Non-IID Data and the Role of Context

Deep learning has prompted rapid progress in computer vision especially in regards to object classification and object detection Krizhevsky et al. [2012], Girshick et al. [2014]. This progress provides ample motivation to apply deep learning methods to star galaxy classification. However, it turns out that these methods cannot be applied effectively without significant modification.

In particular, the point spread function acts as a confounding factor on the input data, creating a significant and correlated source of uncertainty. The high variation of the PSF makes it

entirely possible, even common, for a star in one exposure with a large PSF to appear bigger (be more extended) than a galaxy in another exposure with a smaller PSF; this situation can be see in the example in Figure 7.2. The PSF can vary both across exposures and within different spatial regions of the same exposure. However, the only available evidence of the PSF is its effect on this and nearby objects in the exposure (particularly stars).

Thus, the nearby objects in an exposure provide "context" that is necessary to the prediction task: here, information about the possible magnitude and shape of the PSF. However, this information is entangled with the information of interest, i.e., whether the objects are stars or galaxies. We contend that extracting and sharing this contextual information is critical to effective star galaxy classification.

This assertion is borne out empirically: when we evaluate the performance of a standard classifier framework modeled after AlexNet [Krizhevsky et al., 2012], except using only one output neuron for a binary classifier, on the datasets described in Section 7.4 we find that the PSF has a major and deleterious effect. The results are reported in Table 7.1. If we apply a fixed PSF to all objects, the deep convolutional model achieves a strong performance; this is degraded slightly if we allow the PSF to vary over the exposure ("spatial variation") but keep it constant across exposures ("temporal variation"). However, if we move to the more realistic setting of a PSF that varies both spatially in a single exposure and across different exposures, we see performance degrade to random guessing, with little improvement even as the data set size grows.

In our initial work we also tested an R-CNN model Girshick et al. [2014], which solves the slightly more general problem of simultaneously detecting and classifying objects in an image. In star galaxy classification, detection is typically done by an earlier processing step in the pipeline. And star galaxy detection is a much easier problem than general object detection. Unfortunately the R-CNN achieved poor performance on both detection and classification,

perhaps because of the low signal to noise ratio in the data, which is characteristic of the problem. In addition, images are quite large and the vast majority of pixels view empty space (background), suggesting there may be issues due to class imbalance.

These impediments of standard classifier frameworks motivate us to develop a framework for composing neural network models to capture non-IID data effects while also allowing for varying number of objects in a given exposure.

7.3 ContextNet

The question we are trying to solve is thus not a straightforward classification question, but what we call a *contextual classification* question: the classification of one object cannot be determined without taking into account the context of the surrounding objects. We address this problem by dividing up the modeling procedure into three consecutive steps: local modeling, global modeling and predicting. Each of these steps is associated with its own neural network. The local network is designed to capture local features about each individual object in a single exposure, independent from all other objects in the exposure. It is then replicated for however many objects exist in an exposure. The global network is designed to take in all of the local features and produce global features that describe the exposure as a whole. Finally, the prediction network takes in the local and global features to produce a class prediction; like the local network, it is replicated as many times as there are objects and applied independently to each. A pictorial overview of the model can be seen in Figure 7.4.

We define a cutout of a single object $X = (x_1, x_2, ..., x_n)$ where x_i is the value of a single pixel. We define an exposure $E = \{X_j\}_{j=1}^{m}$ as collection of cutout images where there are m cutouts in each exposure. (In the sequel we show how our architecture can be extended

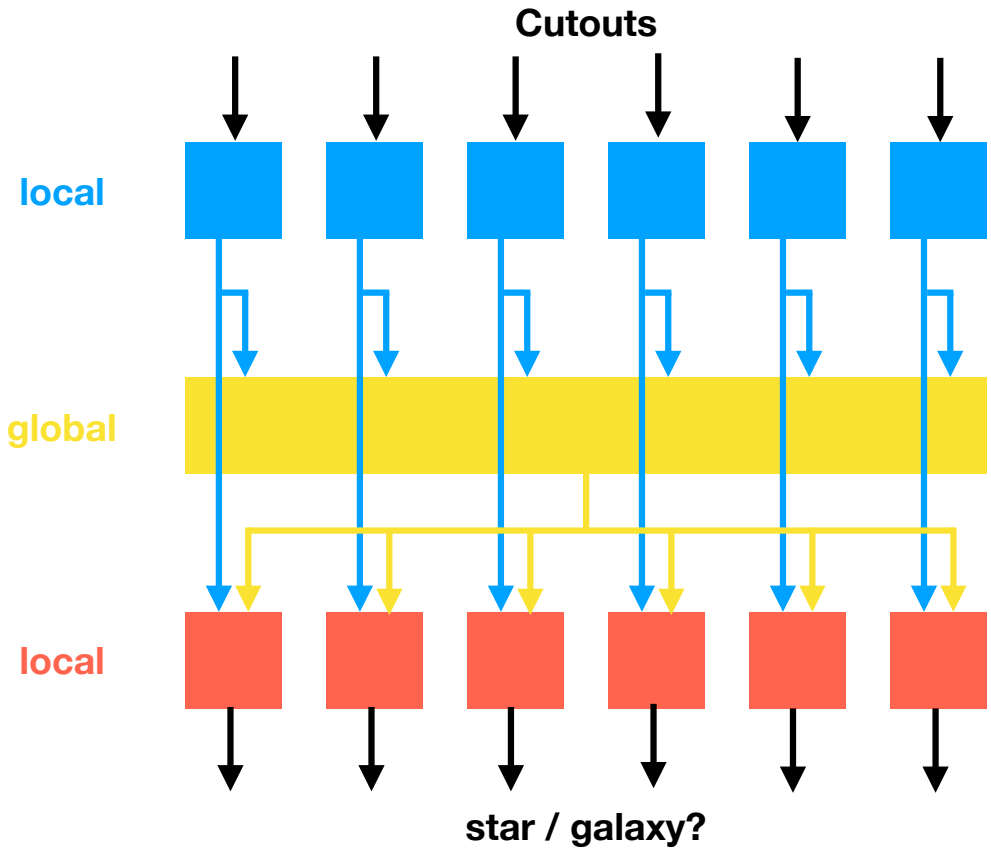

Figure 7.4: High level overview of the ContextNet architecture. Each color represent a different network and a different step in the modeling process. Blue is the local network that takes in cutouts and produces local features for each object. It is replicated as many times as there are objects in the exposure. Yellow is the global model that takes in all the local features and distills that into global features corresponding to entire exposure. Red is the prediction network that takes in both the local and global features and makes the final prediction for each object. It is also replicated as many times as there are objects.

to exposures containing more than m cutouts.) A single input to our model corresponds to one exposure.

The local network LN is a neural network that takes in a single object (cutout image) and outputs a vector of local features:

$$Y_j = (y_1, y_2, ..., y_k) = LN(X_j) \tag{7.1}$$

Note that, for our specific application, we also include the two coordinates representing the

position of the object in the sky as a local feature in Y. The local network is replicated and applied m times to each cutout in the exposure.

The input to the global network, GN, are the concatenated outputs of the local network:

$$G = GN(Y_1, \ldots, Y_m) \tag{7.2}$$

The output of GN is a single vector representing the contextual information in the exposure.

Finally, the prediction network, PN, takes in a single object's local feature vector and the global context vector, and is replicated and applied m times for each cutout image in the exposure:

$$P_i = PN(Y_i, G) \tag{7.3}$$

This results in m predictions, one for each object in the exposure. All three models are trained simultaneously using a binary cross entropy loss for each prediction and propagating the gradients through all three networks. Figure 7.4 shows a high level layout of the model.

Varying numbers of objects. In practice, the number of objects (cutout images) varies by exposure. While our basic model assumes a fixed number of objects per exposure, it is easily extended to predict on exposures with $N > m$ objects. Intuitively, we define a minimum number of predictions, n_p, to make on each cutout. We then create $S_n = \lceil n_p * N/m \rceil$ sets of size m cutouts each. These sets are simply filled with a random selection of cutouts, making sure no cutout is duplicated within the same set, and that each cutout occurs at least n_p times total. We then predict on each of these sets of (fixed) size m, and the predictions are averaged for each cutout. In this setup, as long as we set our minimum m small enough that there will never be an exposure with less than m objects, we can predict on exposures of variable size.

Another important note is that the architectures of the three different neural networks are quite flexible and can be mixed and matched. In our setting, we select to use a convolutional network for the local network and fully connected networks for the global and prediction network. However, it is also possible to construct multiple types of local networks – for example, if there were cutouts of different sizes, there could be a local network for each size as long as the produced local features, Y_j, have the same dimensions among all local networks. Thus ContextNet is not only capable of handling non-IID data, but also a variable number of inputs and inputs of different dimensions and types.

7.4 Experiments

As the start of the LSST survey approaches, simulations are being produced by a variety of teams and collaborations in order to test and calibrate all components of the data management (DM) pipeline. This includes the current star galaxy classifier within the DM stack. It is this simulated data on which we test, and compare to a classifier based on the DM stack. With respect to the domain science, this is the most important comparison since the DM based classifier is what will be used if no superior method is produced. The DM classifier has been inherited and modified from the one used in SDSS and is of the form discussed in Section 7.2. The use of simulated data, while not ideal, is necessary since ground truth on real astronomical data is not possible to obtain.

The architecture of our model is as follows: The local network takes in a cutout of dimension (28, 28) and the layers are Conv(Filters=64, kernel=(3, 3)$\rightarrow$ Elu $\rightarrow$ Conv(Filters=128, kernel=(3, 3) $\rightarrow$ Elu $\rightarrow$ Flatten $\rightarrow$ Dense(20) $\rightarrow$ Elu. We chose only 20 local features because cutouts of galaxies are not complex images, but essentially noisy, elliptical objects positioned in the center of the image (as illustrated in Figure 7.1). The global network takes in the concatenation of the local features from 1000 objects in the exposure, and processes

Table 7.2: Overall comparison of ContextNet with the size magnitude based classifier in the DM stack. ContextNet does significantly better on accuracy and precision and worse on recall. Achieving a high precision sample of stars is important for down stream processing tasks.

Model	Recall	Precision	Accuracy
ContextNet	0.96	0.88	0.93
DM Classifier	0.92	0.82	0.85

with layers Dense(1000) $\rightarrow$ Elu $\rightarrow$ Dense(1000) $\rightarrow$ Elu $\rightarrow$ Dense(1000) $\rightarrow$ Elu. We chose 1000 objects as the minimum number per exposure, which is approximate for LSST. Each exposure will have anywhere from 1200 - 2200 detected sources and anything less than 1000 most likely indicates that something has gone catastrophically wrong with the instruments. The prediction network takes in the local features for a single object and the global features and has the following architecture: Dense(100) $\rightarrow$ Elu $\rightarrow$ Dense(100) $\rightarrow$ Elu $\rightarrow$ Dense(1) $\rightarrow$ Sigmoid. The final output is the probability that the object is a galaxy, and we use binary cross entropy to train.

Our training set consists of 5000 exposures each containing 1000 sources. The test set consists of 1000 exposures and each contain anywhere from 1200-2200 objects. The results are presented in Table 7.2.

As can be seen from Table7.2 and Figure 7.5, ContextNet not only achieves much better overall performance it also achieves even better performance for dim objects. Dim objects are especially important for LSST and future surveys as they image deeper into sky capturing dimmer objects than have previously been measured.

Even though it is not ideal to use simulated data, one of its advantages is that we can use the parameters that defined the objects in the simulation to better understand the predictions being made. This is especially important for deep models where interpretability is hard and

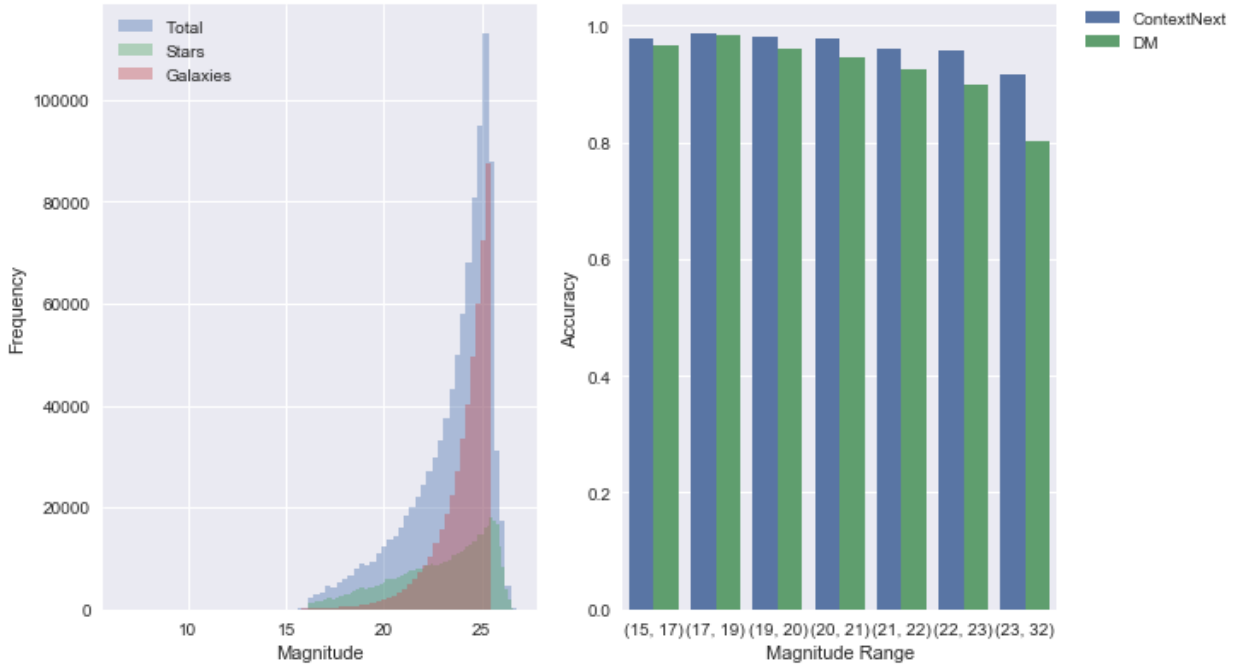

Figure 7.5: Left: The magnitude distribution of the data set used for testing. The color blue is the distribution for all objects. Red is the distribution for galaxies only and green is for stars only. As we can see galaxies tend to be dimmer than stars. Right: Accuracy for the two models binned by magnitude. ContextNet in blue and the DM based classifier in Red. As we can see the two models are competitive for brighter objects, but ContextNet starts to achieve much better performance for dimmer objects. Performance in this dim region is becoming increasingly important as ground-based astronomical surveys image deeper into the sky.

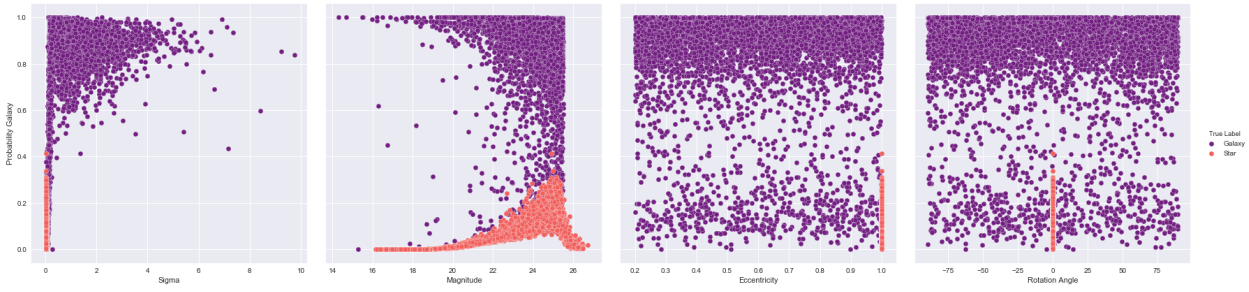

Figure 7.6: In these four plots galaxies are colored purple and stars are pink. The y-axis is the probability of classifying the object as a galaxy. From left to right: the first figure compares the probability of detecting a galaxy with the size of each object, the second with the magnitude, the third with the eccentricity and the last with the amount it was rotated from -90 to 90. Using the simulation parameters can help us interpret the predictions of the model.

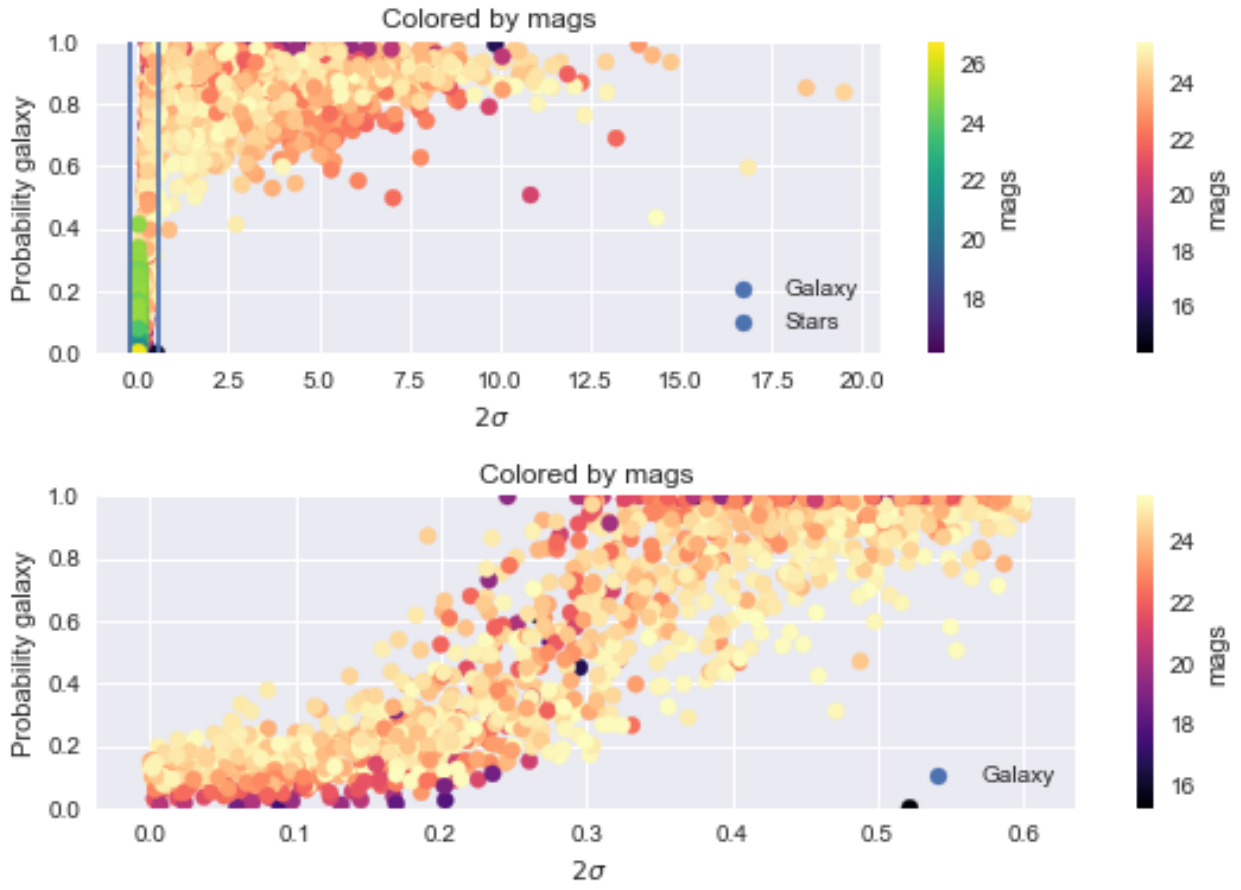

Figure 7.7: The top panel shows the size on each object on the x-axis and the probability of classifying the object as a galaxy on the y-axis; each object is colored by its magnitude. The yellow-red colors are galaxies and the blue-green colors are stars. The bottom panel shows a detail of the leftmost region in the top panel, focusing on smaller objects and with the stars removed.

often not possible. It is also necessary to develop a strong understanding our our model to convince domain scientists to adopt this approach. The main parameters used in the simulation to define stars and galaxies are their brightness or magnitude, size, eccentricity and the amount it is rotated. Stars are modeled as perfectly round objects with negligible size. Galaxies on the other had have distributions over each of these parameters. Figure 7.6 shows the parameters plotted against the probability of being predicted a galaxy by ContextNet. Intuitively the size of the object is the biggest signal for being predicted a galaxy. This is expected when considering that state of the art models for this problem only

use the size of the object to make a prediction. We can also see that the amount the object is rotated or stretched (eccentricity) seems to play a very little role in the prediction. This is somewhat surprising considering that galaxies are the only objects that are rotated and stretched. However this is due to the fact that the PSF can cause stars to appear rotated or stretched making this a much weaker signal. From these plots we can see that the magnitude of the object also appears to be a strong signal for the prediction of ContextNet. This is also a reasonable result given that stars and galaxies have very different magnitude distributions, as seen in Figure 7.5.

We examine the relationship between predictions and magnitudes further in Figure 7.7. Here we plot the probability of being classified as a galaxy on the y-axis and the size of each object on the x-axis. We color each object by its magnitude. The top figure shows all objects and the bottom plot zooms in on just the smaller objects. From this we can see an interesting relationship where small bright galaxies can be classified correctly as galaxies or incorrectly as stars. The incorrect classification of these small bright objects makes sense physically given that small bright galaxies do look very much like stars. But the fact that not all of these objects are classified incorrectly tells us that the model is making classifications based on more than just the size and magnitude of the object.

7.5 Conclusion

In this chapter we presented ContextNet, a framework for composing neural network models to make predictions for non-IID data as well as being able to take in a variable number of inputs and possibly different types of data. We showed that our model achieved better performance for an important problem in astronomy, obtaining superior performance compared to a model that LSST intends to use. Our model does particularly better for dim objects, which are becoming increasingly important. We also are able to partially interpret our re-

sults by relating parameters of the simulated objects with predictions made by the model. This is a necessary analysis when trying to convince domain scientists to adopt new models. In addition we discussed why standard computer vision techniques fail at this problem and need to be extended for contextual classification. We intend to extend this model by not just having one type of local network, but incorporating several for various cutout sizes. We also plan to extend this work to estimate properties of the stars and galaxies in addition to classifying them.

Chapter 8

Conclusion

> Don't adventures ever have an end? I suppose not. Someone else always has to carry on the story.
>
> Bilbo Baggins

This thesis covered three projects spanning several interconnected areas of machine learning and statistics.

8.1 Variational Methods for BOED

We showed how variational methods can be highly effective for evaluating and optimizing an intractable objective function commonly used in Bayesian optimal experimental design. We proposed a novel variational form that was capable of amortizing across designs allowing us to train a single model that was capable of estimating the EIG across the entire feasible region of designs. We showed how design amortization leads to substantially improved computational efficiency compared to prior work. In addition our use of flexible modeling building components from deep learning, in particular set-invariant architectures and conditional normalizing flows allowed our form to more accurately approximate complex

distributions leading to improved accuracy of our EIG estimates compared to prior work.

Going beyond evaluation we showed how our variational form was highly effective at optimizing the EIG over the space of feasible designs and facilitates the use of a broad variety of optimization algorithms, which empowers experimenters to use variational methods for BOED in a wide variety of practical settings. This includes two stage algorithms which first trained an amortized variational posterior and then uses it to make approximations to the EIG during optimization. This included the use of the coordinate-exchange algorithm, which is a commonly used algorithm in OED able to be used with discrete design variables, which are commonly encountered in practical applications of OED. A second two stage algorithm we studied was Bayesian optimization, which is a incredibly popular optimization algorithm when working with expensive to evaluate objective functions and is robust to noise in its estimation. In addition Bayesian optimization is well studied in multi-objective and multi-fidelity settings as well as work work on how it can be used in discrete settings or with gradient information. In addition there are number of excellent software libraries implementing Bayesian optimization that can be immediately used by experimenters in the context of BOED. We showed how training a design amortized variational posterior can significantly improve computational performance for two stage algorithms making them competitive with gradient based methods.

In addition we also generalized the stochastic gradient algorithm proposed in Foster et al. [2020] for simultaneously optimizing the variational parameters and design variables. We generalized this algorithm in the case when our variational posterior is differentiable to the design variables, this results in significantly improved computational performance by allowing us to optimize many designs in parallel using just a single variational posterior. In addition we show that this dependence on the design variables also significantly improves both evaluation and optimization performance.

We believe that this is a promising area for future work and that the work in this thesis provides an excellent foundation to build off of. One area that we believe is particularly interesting and under-studied is how to optimize experimental designs when the experimenter is concerned with functions over their latent parameters. Essentially gerneralized the EIG as follows:

$$EIG(d) = \mathbb{E}_{p(y|d)} \left[H[p(f(\theta))] - H[p(f(\theta)|y, d)] \right] \tag{8.1}$$

where $f(\cdot)$ is some arbitrary function over our latent variables. We believe that the posterior estimator is particularly suited to handle this generalization and requires almost no adjustments from what was presented in this thesis. The other estimators are most likely not as easy to adapt, including nested Monte Carlo. The elegance at which the posterior estimator handles this case can be a significant win for experimenters in all areas of science and engineering who wish to precisely define how they want their experiments to be optimized in terms of functions over their latent variables. In particular this more general formulation allows experimenters to specify and optimize designs for exactly the latent variables they care about while ignoring nuisance parameters when considering what designs are optimal. It is often the case that not all model parameters are cared about equally by experimenters and this more general formulation of the EIG would give them greater flexibility in defining different classes of model parameters when designing experiments.

Further work on variational forms is another promising area for future work. In particular variational forms that accommodate hierarchical models or models with mixed variable types would be very useful.

We also presented two applied projects in the field of astronomy, one focused on optimizing spectroscopic follow-up for astronomical survey and the second was on building a classifica-

tion model for star galaxy classification from ground-based observations.

8.2 Optimizing Spectroscopic Follow-up

As discussed large photometric surveys provide valuable data about the universe, but need to be augmented with more precise spectroscopic information on a subset of the objects. In our work the goal was to use spectroscopic resources to provide accurate labels for the observed object. These spectroscopically labeled objects define a training dataset to train a classification model for the rest of the data. The goal then is to select objects for spectroscopic labeling that will gives us the most informative training dataset. This question broadly falls under the field of active learning, which is a specific form of optimal experimental design. With an amazing team of international scientists we developed and investigated the efficacy of active learning techniques in this domain under realistic constraints. We found that active learning can be a successful strategy for approaching this problem and also produced a software library that can be used by astronomers over the course of their survey.

8.3 ContextNet

In this work we addressed a fundamental challenge in all astronomical data analysis pipelines, classifying objects as stars or galaxies. As discussed in space-based observations this is a relatively easy problem with stars appearing as point sources of light while galaxies have a spatial extent. However terrestrial observations are confounded by the atmospheric effects which acts to convolving all incoming sources of light with a point spread function. Further due to atmospheric turbulence this point spread function is constantly changing in both space and time making the problem of star galaxy classification difficult. We proposed a deep learning model that compensated for this confounding factor by structuring our model

to produce both a local representation for each object in the image and a global representation for the image as a whole. This structure is closely related to how we built our models to satisfy permutation-invariance in our variational BOED work. Structuring our model in this was allowed us to produce a classification model that could deal with the confounding effects of the atmosphere and achieve state of the art performance.

Bibliography

M. Balandat, B. Karrer, D. R. Jiang, S. Daulton, B. Letham, A. G. Wilson, and E. Bakshy. BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. In *Advances in Neural Information Processing Systems 33*, 2020. URL http://arxiv.org/abs/1910.06403.

M. F. Balcan, A. Beygelzimer, and J. Langford. Agnostic active learning. *Journal of Computer and System Sciences*, 75(1):78 – 89, 2009.

D. Barber and F. V. Agakov. The im algorithm: A variational approach to information maximization. In *NIPS*, pages 201–208, 2003. URL http://papers.nips.cc/paper/2410-information-maximization-in-noisy-channels-a-variational-approach.

G. Bazin, N. Palanque-Delabrouille, J. Rich, V. Ruhlmann-Kleider, E. Aubourg, L. Le Guillou, P. Astier, C. Balland, and et al. The core-collapse rate from the Supernova Legacy Survey. *Astronomy and Astrophysics*, 499(3):653–660, June 2009a. doi: 10.1051/0004-6361/200911847.

G. Bazin et al. "the core-collapse rate from snls". *A&A*, 499:653–660, 2009b. doi: 10.1051/0004-6361/200911847.

J. M. Bernardo and A. F. Smith. *Bayesian theory*, volume 405. John Wiley & Sons, 2009.

M. Betoule, R. Kessler, J. Guy, J. Mosher, D. Hardin, R. Biswas, P. Astier, P. El-Hage, and et al. Improved cosmological constraints from a joint analysis of the SDSS-II and SNLS supernova samples. *A&A*, 568:A22, Aug. 2014. doi: 10.1051/0004-6361/201423413.

E. Bingham, J. P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, R. Singh, P. A. Szerlip, P. Horsfall, and N. D. Goodman. Pyro: Deep universal probabilistic programming. *J. Mach. Learn. Res.*, 20:28:1–28:6, 2019. URL http://jmlr.org/papers/v20/18-403.html.

D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

B. Bloem-Reddy and Y. W. Teh. Probabilistic symmetries and invariant neural networks. *J. Mach. Learn. Res.*, 21:90–1, 2020.

M. Bolte. Modern observational techniques, 2015. URL http://www.ucolick.org/~bolte/AY257/s_n.pdf.

J. Bosch, R. Armstrong, S. Bickerton, H. Furusawa, et al. The Hyper Suprime-Cam software pipeline. *Publications of the Astronomical Society of Japan*, 70:S5, Jan. 2018. doi: 10.1093/pasj/psx080.

L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

L. Breiman. Random forests. *Machine Learning*, 45(1):5 – 32, 2001.

E. Brochu, V. M. Cora, and N. de Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning, 2010. URL `https://arxiv.org/abs/1012.2599`.

H. Campbell et al. Cosmology with Photometrically Classified Type Ia Supernovae from the SDSS-II Supernova Survey. *ApJ*, 763:88, Feb. 2013. doi: 10.1088/0004-637X/763/2/88.

K. Chaloner and I. Verdinelli. Bayesian experimental design: A review. *Statistical Science*, pages 273–304, 1995.

T. Charnock and A. Moss. Deep Recurrent Neural Networks for Supernovae Classification. *ApJ*, 837:L28, Mar. 2017. doi: 10.3847/2041-8213/aa603d.

M. J. Childress et al. OzDES multifibre spectroscopy for the Dark Energy Survey: Three year results and first data release. *ArXiv e-prints*, Aug. 2017.

D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4(1):129–145, 1996.

A. Conley et al. Supernova Constraints and Systematic Uncertainties from the First Three Years of the Supernova Legacy Survey. *ApJS*, 192:1–+, Jan. 2011. doi: 10.1088/0067-0049/192/1/1.

M. Dai, S. Kuhlmann, Y. Wang, and E. Kovacs. Photometric classification and redshift estimation of LSST Supernovae. *ArXiv e-prints*, Jan. 2017.

D. DeBarr and H. Wechsler. Spam detection using clustering, random forests, and active learning. In *Sixth Conference on Email and Anti-Spam. Mountain View, California*, pages 1–6. Citeseer, 2009.

R. DeLoach. *Applications of modern experiment design to wind tunnel testing at NASA Langley Research Center*. 1997. doi: 10.2514/6.1998-713. URL `https://arc.aiaa.org/doi/abs/10.2514/6.1998-713`.

P. DIACONIS and B. SKYRMS. *Ten Great Ideas about Chance*. Princeton University Press, 2018. ISBN 9780691174167. URL `http://www.jstor.org/stable/j.ctvc77m33`.

L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using real NVP. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL `https://openreview.net/forum?id=HkpbnH9lx`.

C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios. Neural spline flows. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/7ac71d433f282034e088473244df8c02-Paper.pdf.

C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios. nflows: normalizing flows in PyTorch, Nov. 2020. URL https://doi.org/10.5281/zenodo.4296287.

D. Eriksson, M. Pearce, J. Gardner, R. D. Turner, and M. Poloczek. Scalable global optimization via local bayesian optimization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/6c990b7aca7bc7058f5e98ea909e924b-Paper.pdf.

R. J. Foley and K. Mandel. Classifying Supernovae Using Only Galaxy Data. *ApJ*, 778:167, Dec. 2013. doi: 10.1088/0004-637X/778/2/167.

F. Förster, J. C. Maureira, J. San Martín, M. Hamuy, J. Martínez, P. Huijse, G. Cabrera, L. Galbany, and et al. The High Cadence Transient Survey (HITS). I. Survey Design and Supernova Shock Breakout Constraints. *The Astrophysical Journal*, 832(2):155, Dec. 2016. doi: 10.3847/0004-637X/832/2/155.

A. Foster, M. Jankowiak, E. Bingham, P. Horsfall, Y. W. Teh, T. Rainforth, and N. Goodman. Variational bayesian optimal experimental design. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/d55cbf210f175f4a37916eafe6c04f0d-Paper.pdf.

A. Foster, M. Jankowiak, M. O'Meara, Y. W. Teh, and T. Rainforth. A unified stochastic gradient approach to designing bayesian-optimal experiments. In *International Conference on Artificial Intelligence and Statistics*, pages 2959–2969. PMLR, 2020.

A. Foster, D. R. Ivanova, I. Malik, and T. Rainforth. Deep adaptive design: Amortizing sequential bayesian experimental design. In *International Conference on Machine Learning*, pages 3384–3395. PMLR, 2021.

P. I. Frazier. A tutorial on bayesian optimization, 2018. URL https://arxiv.org/abs/1807.02811.

Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2-3):133–168, 1997.

G. Gamow. The Evolution of the Universe. *Nature*, 162:680–682, Oct. 1948. doi: 10.1038/162680a0.

J. A. Garmilla. *Star Galaxy Separation in Hyper Suprime-Cam and Mapping the Milky Way with Star Counts*. PhD thesis, Astrophysical Sciences Department, Princeton University, 2016.

R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition*, 2014.

A. Goobar and B. Leibundgut. Supernova Cosmology: Legacy and Future. *Annual Review of Nuclear and Particle Science*, 61:251–279, Nov. 2011. doi: 10.1146/annurev-nucl-102010-130434.

P. Goos and B. Jones. *Optimal design of experiments: a case study approach*. John Wiley & Sons, 2011.

K. D. Gupta, R. Pampana, R. Vilalta, E. E. O. Ishida, and R. S. de Souza. Automated supernova ia classification using adaptive learning techniques. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, Dec 2016.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Y. D. Hezaveh, L. P. Levasseur, and P. J. Marshall. Fast automated analysis of strong gravitational lenses with convolutional neural networks. *Nature*, 548:555–557, Aug. 2017. doi: 10.1038/nature23463.

W. Hillebrandt and J. C. Niemeyer. Type ia supernova explosion models. *Annual Review of Astronomy and Astrophysics*, 38(1):191–230, 2000. doi: 10.1146/annurev.astro.38.1.191. URL `https://doi.org/10.1146/annurev.astro.38.1.191`.

C. Hirata and U. Seljak. Shear calibration biases in weak-lensing surveys. *Monthly Notices of the Royal Astronomy*, 343:459–480, Aug. 2003. doi: 10.1046/j.1365-8711.2003.06683.x.

R. Hlozek, M. Kunz, B. Bassett, M. Smith, J. Newling, M. Varughese, R. Kessler, J. P. Bernstein, H. Campbell, B. Dilday, B. Falck, J. Frieman, S. Kuhlmann, H. Lampeitl, J. Marriner, R. C. Nichol, A. G. Riess, M. Sako, and D. P. Schneider. Photometric Supernova Cosmology with BEAMS and SDSS-II. *ApJ*, 752:79, June 2012. doi: 10.1088/0004-637X/752/2/79.

S. C. H. Hoi, R. Jin, J. Zhu, and M. R. Lyu. Semi-supervised svm batch mode active learning for image retrieval. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, June 2008. doi: 10.1109/CVPR.2008.4587350.

N. Houlsby, F. Huszár, Z. Ghahramani, and M. Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

B. Hoyle, K. Paech, M. M. Rau, S. Seitz, and J. Weller. Tuning target selection algorithms to improve galaxy redshift estimates. *MNRAS*, 458:4498–4511, June 2016. doi: 10.1093/mnras/stw563.

E. E. O. Ishida. Machine learning and the future of supernova cosmology. *Nature Astronomy*, 3:680–682, Aug. 2019. doi: 10.1038/s41550-019-0860-6.

E. E. O. Ishida and R. S. de Souza. Kernel PCA for Type Ia supernovae photometric classification. *MNRAS*, 430:509–532, Mar. 2013. doi: 10.1093/mnras/sts650.

E. E. O. Ishida, R. Beck, S. González-Gaitán, R. S. de Souza, A. Krone-Martins, J. W. Barrett, N. Kennamer, and e. Vilalta. Optimizing spectroscopic follow-up strategies for supernova photometric classification with active learning. *Monthly Notices of the Royal Astronomical Society*, 483(1):2–18, Feb. 2019. doi: 10.1093/mnras/sty3015.

D. R. Ivanova, A. Foster, S. Kleinegesse, M. U. Gutmann, and T. Rainforth. Implicit deep adaptive design: Policy-based experimental design without likelihoods. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 25785–25798. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper/2021/file/d811406316b669ad3d370d78b51b1d2e-Paper.pdf`.

M. Jarvis, E. Sheldon, J. Zuntz, T. Kacprzak, et al. The DES Science Verification weak lensing shear catalogues. *Monthly Notices of the Royal Astronomy*, 460:2245–2281, Aug. 2016. doi: 10.1093/mnras/stw990.

B. D. Johnson and A. P. S. Crotts. Photometric Identification of Type Ia Supernovae at Moderate Redshift. *AJ*, 132:756–768, Aug. 2006. doi: 10.1086/503528.

D. O. Jones, D. M. Scolnic, A. G. Riess, R. Kessler, A. Rest, R. P. Kirshner, E. Berger, C. A. Ortega, R. J. Foley, R. Chornock, P. J. Challis, W. S. Burgett, K. C. Chambers, P. W. Draper, H. Flewelling, M. E. Huber, N. Kaiser, R.-P. Kudritzki, N. Metcalfe, R. J. Wainscoat, and C. Waters. Measuring the Properties of Dark Energy with Photometrically Classified Pan-STARRS Supernovae. I. Systematic Uncertainty from Core-collapse Supernova Contamination. *ApJ*, 843:6, July 2017. doi: 10.3847/1538-4357/aa767b.

M. Jurić, J. Kantor, K. Lim, R. H. Lupton, et al. The LSST Data Management System. *ArXiv e-prints*, Dec. 2015.

S. M. Kahn, N. Kurita, K. Gilmore, M. Nordby, P. O'Connor, R. Schindler, J. Oliver, R. Van Berg, S. Olivier, V. Riot, P. Antilogus, T. Schalk, M. Huffer, G. Bowden, J. Singal, and M. Foss. Design and development of the 3.2 gigapixel camera for the Large Synoptic Survey Telescope. In I. S. McLean, S. K. Ramsay, and H. Takami, editors, *Ground-based and Airborne Instrumentation for Astronomy III*, volume 7735 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, page 0, 2010. doi: 10.1117/12.857920.

N. V. Karpenka, F. Feroz, and M. P. Hobson. A simple and robust method for automated photometric classification of supernovae using neural networks. *MNRAS*, 429:1278–1285, Feb. 2013. doi: 10.1093/mnras/sts412.

N. Kennamer, D. Kirkby, A. Ihler, and F. J. Sanchez-Lopez. ContextNet: Deep learning for star galaxy classification. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2582–2590. PMLR, 10–15 Jul 2018. URL `https://proceedings.mlr.press/v80/kennamer18a.html`.

N. Kennamer, E. E. Ishida, S. Gonzalez-Gaitan, R. S. de Souza, A. Ihler, K. Ponder, R. Vilalta, A. Möller, D. O. Jones, M. Dai, et al. Active learning with resspect: Resource allocation for extragalactic astronomical transients. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 3115–3124. IEEE, 2020.

N. Kennamer, S. Walton, and A. Ihler. Design amortization for bayesian optimal experimental design, 2022. URL `https://arxiv-export1.library.cornell.edu/abs/2210.03283?context=stat.ML`.

R. Kessler, B. Bassett, P. Belov, V. Bhatnagar, H. Campbell, A. Conley, J. A. Frieman, A. Glazov, and et al. Results from the Supernova Photometric Classification Challenge. *Publications of the Astronomical Society of Pacific*, 122:1415–1431, Dec. 2010. doi: 10.1086/657607.

D. P. Kingma and M. Welling. Auto-encoding variational bayes, 2013. URL `https://arxiv.org/abs/1312.6114`.

A. Kirsch, J. van Amersfoort, and Y. Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. In *Advances in Neural Information Processing Systems*, pages 7026–7037, 2019.

I. Kobyzev, S. J. Prince, and M. A. Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE transactions on pattern analysis and machine intelligence*, 43 (11):3964–3979, 2020.

J. Kranjc, J. Smailović, V. Podpečan, M. Grčar, M. Žnidaršič, and N. Lavrač. Active learning for sentiment analysis on data streams: Methodology and workflow implementation in the clowdflows platform. *Information Processing & Management*, 51(2):187–203, 2015.

A. Krause. *Optimizing sensing*. PhD thesis, Carnegie Mellon University, 2008.

A. Krause and D. Golovin. Submodular function maximization. In *Tractability*, 2014.

A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9(Feb):235–284, 2008.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL `https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf`.

S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

N. V. Kuznetsova and B. M. Connolly. A Probabilistic Approach to Classifying Supernovae Using Photometric Information. *ApJ*, 659:530–540, Apr. 2007. doi: 10.1086/511814.

Y. Liu. Active learning with support vector machine applied to gene expression data for cancer classification. *Journal of chemical information and computer sciences*, 44(6):1936–1941, 2004.

M. Lochner, J. D. McEwen, H. V. Peiris, O. Lahav, and M. K. Winter. Photometric Supernova Classification with Machine Learning. *The Astrophysical Journal, Supplement*, 225: 31, Aug. 2016a. doi: 10.3847/0067-0049/225/2/31.

M. Lochner, J. D. McEwen, H. V. Peiris, O. Lahav, and M. K. Winter. Photometric Supernova Classification with Machine Learning. *ApJS*, 225:31, Aug. 2016b. doi: 10.3847/0067-0049/225/2/31.

I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=Bkg6RiCqY7`.

LSST Science Collaboration, P. A. Abell, J. Allison, S. F. Anderson, J. R. Andrew, J. R. P. Angel, L. Armus, D. Arnett, S. J. Asztalos, T. S. Axelrod, et al. LSST Science Book, Version 2.0. *ArXiv e-prints*, Dec. 2009.

P. V. Luong, S. Gupta, D. Nguyen, S. Rana, and S. Venkatesh. Bayesian optimization with discrete variables. In *Australasian Conference on Artificial Intelligence*, 2019.

R. Lupton, J. E. Gunn, Z. Ivezić, G. R. Knapp, and S. Kent. The SDSS Imaging Pipelines. In F. R. Harnden, Jr., F. A. Primini, and H. E. Payne, editors, *Astronomical Data Analysis Software and Systems X*, volume 238 of *Astronomical Society of the Pacific Conference Series*, page 269, 2001.

D. MacKay, D. Kay, and C. U. Press. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003. ISBN 9780521642989. URL `https://books.google.fr/books?id=AKuMj4PN_EMC`.

K. Madsen, H. B. Nielsen, and O. Tingleff. Methods for non-linear least squares problems (2nd ed.), 2004.

R. Mandelbaum, C. M. Hirata, A. Leauthaud, R. J. Massey, and J. Rhodes. Precision simulation of ground-based lensing data using observations from space. *Monthly Notices of the Royal Astronomy*, 420:1518–1540, Feb. 2012. doi: 10.1111/j.1365-2966.2011.20138.x.

D. Masters, P. Capak, D. Stern, O. Ilbert, M. Salvato, S. Schmidt, G. Longo, J. Rhodes, and et al. Mapping the Galaxy Color-Redshift Relation: Optimal Photometric Redshift Calibration Strategies for Cosmology Surveys. *ApJ*, 813:53, Nov. 2015. doi: 10.1088/0004-637X/813/1/53.

P. McCullagh and J. Nelder. *Generalized Linear Models, Second Edition*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis, 1989. ISBN 9780412317606. URL https://books.google.com/books?id=h9kFH2_FfBkC.

R. K. Meyer and C. J. Nachtsheim. The coordinate-exchange algorithm for constructing exact optimal experimental designs. *Technometrics*, 37(1):60–69, 1995a. ISSN 00401706. URL http://www.jstor.org/stable/1269153.

R. K. Meyer and C. J. Nachtsheim. The coordinate-exchange algorithm for constructing exact optimal experimental designs. *Technometrics*, 37(1):60–69, 1995b.

A. Möller and T. de Boissière. SuperNNova: an open-source framework for Bayesian, neural network-based supernova classification. *Monthly Notices of the Royal Astronomical Society*, 491(3):4277–4293, Jan. 2020. doi: 10.1093/mnras/stz3312.

A. Möller, V. Ruhlmann-Kleider, C. Leloup, J. Neveu, N. Palanque-Delabrouille, J. Rich, R. Carlberg, C. Lidman, and C. Pritchet. Photometric classification of type Ia supernovae in the SuperNova Legacy Survey with supervised learning. *J. Cosmology Astropart. Phys.*, 12:008, Dec. 2016. doi: 10.1088/1475-7516/2016/12/008.

D. C. Montgomery. *Design and analysis of experiments*. John wiley & sons, 2017.

J. I. Myung, D. R. Cavagnaro, and M. A. Pitt. A tutorial on adaptive design optimization. *Journal of mathematical psychology*, 57(3-4):53–67, 2013.

B. Naul, J. S. Bloom, F. Pérez, and S. van der Walt. A recurrent neural network for classification of unevenly sampled variable stars. *Nature Astronomy*, 2:151–155, Feb. 2018. doi: 10.1038/s41550-017-0321-z.

G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294, 1978.

J. Newling, M. Varughese, B. Bassett, H. Campbell, R. Hlozek, M. Kunz, H. Lampeitl, B. Martin, R. Nichol, D. Parkinson, and M. Smith. Statistical classification techniques for photometric supernova typing. *MNRAS*, 414:1987–2004, July 2011. doi: 10.1111/j.1365-2966.2011.18514.x.

R. Nowak. Noisy generalized binary search. In *Advances in Neural Information Processing Systems*, pages 1366–1374, 2009.

R. D. Nowak. The geometry of generalized binary search. *IEEE Transactions on Information Theory*, 57(12):7893–7906, 2011.

A. B. Owen. *Monte Carlo theory, methods and examples*. 2013.

G. Papamakarios, E. T. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *J. Mach. Learn. Res.*, 22(57):1–64, 2021.

S. Perlmutter, G. Aldering, G. Goldhaber, R. A. Knop, P. Nugent, P. G. Castro, S. Deustua, S. Fabbro, A. Goobar, and et al. Measurements of Ω and Λ from 42 High-Redshift Supernovae. *ApJ*, 517:565–586, June 1999. doi: 10.1086/307221.

K. Perrett et al. Real-time Analysis and Selection Biases in the Supernova Legacy Survey. *AJ*, 140:518–532, Aug. 2010. doi: 10.1088/0004-6256/140/2/518.

M. M. Phillips. The absolute magnitudes of Type IA supernovae. *ApJ*, 413:L105–L108, Aug. 1993.

Planck Collaboration, R. Adam, P. A. R. Ade, N. Aghanim, Y. Akrami, M. I. R. Alves, F. Argüeso, M. Arnaud, F. Arroja, M. Ashdown, and et al. Planck 2015 results. I. Overview of products and scientific results. *A&A*, 594:A1, Sept. 2016. doi: 10.1051/0004-6361/201527101.

D. Poznanski, A. Gal-Yam, D. Maoz, A. V. Filippenko, D. C. Leonard, and T. Matheson. Not Color-Blind: Using Multiband Photometry to Classify Supernovae. *PASP*, 114:833–845, Aug. 2002. doi: 10.1086/341741.

D. Poznanski, D. Maoz, and A. Gal-Yam. Bayesian Single-Epoch Photometric Classification of Supernovae. *AJ*, 134:1285–1297, Sept. 2007. doi: 10.1086/520956.

T. Rainforth, R. Cornish, H. Yang, A. Warrington, and F. Wood. On nesting monte carlo estimators. In *International Conference on Machine Learning*, pages 4267–4276. PMLR, 2018.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005. ISBN 026218253X.

J. Regier, A. Miller, J. McAuliffe, R. Adams, M. Hoffman, D. Lang, D. Schlegel, and M. Prabhat. Celeste: Variational inference for a generative model of astronomical images. In F. Bach and D. Blei, editors, *Proc. International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2095–2103, Lille, France, 07–09 Jul 2015.

E. A. Revsbech, R. Trotta, and D. A. van Dyk. STACCATO: A Novel Solution to Supernova Photometric Classification with Biased Training Sets. *ArXiv e-prints*, June 2017.

D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In E. P. Xing and T. Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1278–1286, Bejing, China, 22–24 Jun 2014. PMLR. URL `https://proceedings.mlr.press/v32/rezende14.html`.

J. W. Richards, D. Homrighausen, P. E. Freeman, C. M. Schafer, and D. Poznanski. Semi-supervised learning for photometric supernova classification. *MNRAS*, 419:1121–1135, Jan. 2012a. doi: 10.1111/j.1365-2966.2011.19768.x.

J. W. Richards, D. L. Starr, H. Brink, A. A. Miller, J. S. Bloom, N. R. Butler, J. B. James, J. P. Long, and J. Rice. Active Learning to Overcome Sample Selection Bias: Application to Photometric Variable Star Classification. *ApJ*, 744:192, Jan. 2012b. doi: 10.1088/0004-637X/744/2/192.

A. G. Riess, A. V. Filippenko, P. Challis, A. Clocchiatti, A. Diercks, P. M. Garnavich, R. L. Gilliland, C. J. Hogan, and et al. Observational Evidence from Supernovae for an Accelerating Universe and a Cosmological Constant. *AJ*, 116:1009–1038, Sept. 1998. doi: 10.1086/300499.

C. P. Robert and G. Casella. *Monte Carlo statistical methods*, volume 2. Springer, 1999.

S. A. Rodney and J. L. Tonry. Fuzzy Supernova Templates. I. Classification. *ApJ*, 707: 1064–1079, Dec. 2009. doi: 10.1088/0004-637X/707/2/1064.

B. T. P. Rowe, M. Jarvis, R. Mandelbaum, G. M. Bernstein, et al. GALSIM: The modular galaxy image simulation toolkit. *Astronomy and Computing*, 10:121–150, Apr. 2015. doi: 10.1016/j.ascom.2015.02.002.

E. G. Ryan, C. C. Drovandi, J. M. McGree, and A. N. Pettitt. A review of modern computational algorithms for bayesian optimal design. *International Statistical Review*, 84(1): 128–154, 2016.

M. Sako, B. Bassett, A. Becker, D. Cinabro, F. DeJongh, D. L. Depoy, B. Dilday, M. Doi, and et al. The Sloan Digital Sky Survey-II Supernova Survey: Search Algorithm and Follow-up Observations. *AJ*, 135:348-373, Jan. 2008. doi: 10.1088/0004-6256/135/1/348.

J. Schulman, N. Heess, T. Weber, and P. Abbeel. Gradient estimation using stochastic computation graphs. *Advances in Neural Information Processing Systems*, 28, 2015.

P. Sebastiani and H. P. Wynn. Maximum entropy sampling and optimal bayesian experimental design. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(1):145–157, 2000.

B. Settles. *Active Learning*. Morgan & Claypool, 2012.

H. S. Seung, H. Sompolinsky, and N. Tishby. Statistical mechanics of learning from examples. *Physical review A*, 45(8):6056, 1992.

K. SMITH. ON THE STANDARD DEVIATIONS OF ADJUSTED AND INTERPOLATED VALUES OF AN OBSERVED POLYNOMIAL FUNCTION AND ITS CONSTANTS AND THE GUIDANCE THEY GIVE TOWARDS A PROPER CHOICE OF THE DISTRIBUTION OF OBSERVATIONS. *Biometrika*, 12(1-2):1–85, 11 1918. ISSN 0006-3444. doi: 10.1093/biomet/12.1-2.1. URL `https://doi.org/10.1093/biomet/12.1-2.1`.

T. Solorio, O. Fuentes, R. Terlevich, and E. Terlevich. An active instance-based machine learning method for stellar population studies. *MNRAS*, 363:543–554, Oct. 2005. doi: 10.1111/j.1365-2966.2005.09456.x.

D. N. Spergel, R. Bean, O. Doré, M. R. Nolta, C. L. Bennett, J. Dunkley, G. Hinshaw, N. Jarosik, E. Komatsu, L. Page, H. V. Peiris, L. Verde, M. Halpern, R. S. Hill, A. Kogut, M. Limon, S. S. Meyer, N. Odegard, G. S. Tucker, J. L. Weiland, E. Wollack, and E. L. Wright. Three-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Implications for Cosmology. *ApJS*, 170:377–408, June 2007. doi: 10.1086/513700.

M. Sullivan et al. Photometric Selection of High-Redshift Type Ia Supernova Candidates. *AJ*, 131:960–972, Feb. 2006. doi: 10.1086/499302.

G. Taguchi. *System of Experimental Design: Engineering Methods to Optimize Quality and Minimize Costs*. Number v. 1 in System of Experimental Design: Engineering Methods to Optimize Quality and Minimize Costs. UNIPUB/Kraus International Publications, 1987. ISBN 9780941243001. URL `https://books.google.com/books?id=-PVQAAAAMAAJ`.

C. A. Thompson, M. E. Califf, and R. J. Mooney. Active learning for natural language parsing and information extraction. In *ICML*, pages 406–414. Citeseer, 1999.

R. Tripp. "a two-parameter luminosity correction for type ia supernovae". *A&A*, 331:815–820, Mar. 1998.

M. M. Varughese, R. von Sachs, M. Stephanou, and B. A. Bassett. Non-parametric transient classification using adaptive wavelets. *MNRAS*, 453:2848–2861, Nov. 2015. doi: 10.1093/mnras/stv1816.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf`.

R. Vilalta, E. E. O. Ishida, R. Beck, R. Sutrisno, R. S. de Souza, and A. Mahabal. Photometric redshift estimation: An active learning approach. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, Dec 2017.

B. T. Vincent and T. Rainforth. The darc toolbox: automated, flexible, and efficient delayed and risky choice experiments using bayesian adaptive design. *PsyArXiv*, 2017.

M. J. Wainwright, M. I. Jordan, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.

Y. Wang, E. Gjergo, and S. Kuhlmann. Analytic photometric redshift estimator for Type Ia supernovae from the Large Synoptic Survey Telescope. *MNRAS*, 451:1955–1963, Aug. 2015. doi: 10.1093/mnras/stv1090.

J. Wu, M. Poloczek, A. G. Wilson, and P. Frazier. Bayesian optimization with gradients. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper/2017/file/64a08e5f1e6c39faeb90108c430eb120-Paper.pdf`.

X. Xia, P. Protopapas, and F. Doshi-Velez. *Cost-Sensitive Batch Mode Active Learning: Designing Astronomical Observation by Optimizing Telescope Time and Telescope Choice*, pages 477–485. 2016. doi: 10.1137/1.9781611974348.54.

R. W. Yeung. A new outlook on shannon's information measures. *IEEE transactions on information theory*, 37(3):466–474, 1991.

M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola. Deep sets. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper/2017/file/f22e4747da1aa27e363d86d40ff442fe-Paper.pdf`.