Better Input, Better Output: Improving photometric redshifts by enhancing training data and optimizing observations

John Bryce Kalmbach

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2019

Reading Committee:

Miguel Morales, Chair

Andrew J. Connolly

Paul A. Wiggins

Program Authorized to Offer Degree:
Physics

University of Washington

Abstract

Better Input, Better Output: Improving photometric redshifts by enhancing training data and optimizing observations

John Bryce Kalmbach

Chair of the Supervisory Committee:
Professor Miguel Morales
Department of Physics

We are at the beginning of an era of large scale survey astronomy where we will soon measure photometry for billions of galaxies. In order to effectively use these galaxies for dark energy measurements we require measurements of the distances to these galaxies. Spectroscopic redshifts are not feasible for more than a small fraction of these galaxies and thus our primary distance measurements will rely on photometric redshift methods. This thesis highlights three challenges in photometric redshift estimation and techniques we developed to tackle these challenges:

Using Information Theory to Optimize Bandpasses for Photometric Redshifts:

We apply ideas from information theory to create a method for the design of optimal filters for photometric redshift estimation. We show the method applied to a series of simple example filters in order to motivate an intuition for how photometric redshift estimators respond to the properties of photometric passbands. We then design a realistic set of six filters covering optical wavelengths that optimize photometric redshifts for $z <= 2$. We create a simulated catalog for these optimal filters and use our filters with a photometric redshift estimation code to compare to the filters for the Large Synoptic Survey Telescope (LSST) which have key features in common with our optimal filters.

Expanding Template Sets for Template Based Photo-Z Algorithms:

Measuring the physical properties of galaxies such as redshift frequently requires the use of Spectral Energy Distributions (SEDs). SED template sets are, however, often small in number and cover limited portions of photometric color space. Here we present a new method to estimate SEDs as a function of color from a small training set of template SEDs. We first cover the mathematical background behind the technique before demonstrating our ability to reconstruct spectra based upon colors and then compare to other common interpolation and extrapolation methods. When the photometric filters and spectra overlap we show reduction of error in the estimated spectra of over 65% compared to the more commonly used techniques. We also show an expansion of the method to wavelengths beyond the range of the photometric filters. Finally, we demonstrate the usefulness of our technique by generating 50 additional SED templates from an original set of 10 and applying the new set to photometric redshift estimation. We are able to reduce the photometric redshifts standard deviation by at least 22.0% and the outlier rejected bias by over 86.2% compared to original set for $z \leq 3$.

Color Space Data Augmentation for Photometric Redshifts:

When training sets for machine learning methods are not representative of the test set then there can be errors in the resulting estimates. In photometric redshifts this can happen when the color space of the spectroscopic data does not match the observed galaxy color space for an empirical photometric redshift estimation method. We first show how a lack of data in a region of color space of the training data affects photometric redshift estimation and then develop three different methods to add in synthetic training data to the missing area to mitigate the errors. Our best performing method lowers the photo-z bias by 51% and reduces the outlier fraction by 9.6% in the test data that lies in the missing area of color

space compared to an unrepresentative training catalog.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

ANN: Artificial Neural Network

CMNN: Color Matched Nearest Neighbor

ESP: Estimating Spectra from Photometry

GAN: Generative Adversarial Network

GP: Gaussian Process

GPR: Gaussian Process Regression

IG: Information Gain

IQR: Interquartile Range

LSST: Large Synoptic Survey Telescope

PCA: Principal Component Analysis

SDSS: Sloan Digital Sky Survey

SED: Spectral Energy Distribution

ACKNOWLEDGMENTS

I first want to thank my wife, Kristin, for her love and support throughout this effort. I couldn't have done this without her positive encouragement and selfless understanding.

I also want to thank my mother, Maria Victoria, for helping me follow my dreams since the very beginning. She has been my number one cheerleader since I was a little boy looking up at the night sky in Alaska and wondering what was out there.

Finally, I want to thank my advisor, Andrew Connolly, for the opportunity to work with him. I have learned so much working with Andy and am grateful for his mentorship.

DEDICATION

This dissertation is dedicated to my late father, Fritz Kalmbach, who taught me the importance of education.

Chapter 1

INTRODUCTION

In September 1912, Vesto Slipher at Lowell Observatory observed the Andromeda galaxy with a spectrograph and based upon the shift of spectral lines made the first measurements of the radial velocity of a galaxy (Slipher 1913). Slipher continued making measurements of radial velocities and had over a dozen velocities for "sprial nebulae" by 1915 (Slipher 1915). Over a decade later, Edwin Hubble connected radial velocities derived from Slipher's techniques to his own measurements of the distances to galaxies from observations of Cepheid variables and novae and showed that the further away a galaxy is then the faster it moves away from us (Hubble 1929). This milestone result provided the first evidence that the universe is expanding and implies that space itself is expanding. One consequence of this is that the wavelength of light traveling through space is stretched and results in the effect known as cosmological redshift, $z = \frac{\Delta\lambda}{\lambda}$ and provides a direct relationship between the observed spectrum of a galaxy and the cosmological distance of that galaxy. This effect is what Slipher measured with his spectroscopic observations of galaxies and is still a very important tool in the study of cosmology. If one can measure the spectral energy distribution (SED) of a galaxy then by matching the observed wavelengths of strong features like the emission and absorption lines present within the SED to those recorded in laboratory measurements here on Earth the redshift of a galaxy can be found to a high degree of accuracy.

The ability to use the spectrum of the galaxy to determine the distance to that galaxy removes the need to resolve indiviual stars in a galaxy to measure distance and allows us to get distance estimates for the furthest galaxies in the universe. Since the light from galaxies takes time to reach Earth proportional to the distance it has to travel we can also relate

information on distances to understand the evolution of the universe over time. Baryon Acoustic Oscillations (BAO) and weak graviational lensing are two cosmological probes that help us learn about dark energy and the expansion history of the universe. Both benefit from large galaxy surveys that cover large areas of the sky and provide observations of as many galaxies as possible. Large galaxy surveys like the catalog that the Large Synoptic Survey Telescope (LSST) (Ivezić et al. 2008) are an essential tool to enable this kind of science. But the LSST is an imaging telescope and will not provide spectra for the billions of observed galaxies in its catalog and the measurements required for spectroscopic redshift (spec-z) are difficult and time consuming. Accurate spec-z measurements require multiple strong emission line measurements in the spectrum and this gets much harder as galaxies get fainter. To get accurate spec-z measurements for galaxies at the LSST 'gold sample' depth of $i = 25.3$ for dark energy measurements would require 100 hours of observing time on an 8-10 meter diameter telescope and would still only produce a high-confidence redshift 60-75% of the time (Newman et al. 2015).

Therefore, the need exists for methods that are able to use the wide, broadband filters of photometric surveys to provide estimates of the redshift of galaxies. This relies on using the photometric observations as a way to understand the general shape of the underlying SED of a galaxy and translate this information into a redshift estimate. This class of methods are known appropriately as photometric redshift (photo-z) methods. While less accurate than spec-z methods, the photo-z methods are applicable to the entire catalog of galaxies in a large, photometric survey with the information already at hand.

1.1 Photometric Redshift Estimation Techniques

The differences between two photometric filters is referred to as a color and when a galaxy has a greater flux in the filter centered at a longer wavelength we refer to it as redder. As a galaxy is redshifted the flux from bluer parts of a spectrum move to longer, redder wavelengths and the colors of a galaxy gets redder. The two most prominent features from the SED of a galaxy that affect the colors of galaxies in the optical wavelengths are the sharp

break at 4000 Å and the Lyman break above 912 Å (see Figure 1.1). Each of the breaks are areas in the spectrum of a galaxy where there is a much larger amount of flux on the redder side of the break. When two different photometric bandpasses appear on either side of one of the breaks there is a large difference in the flux of the galaxy in the two filters and a strong red color to the galaxy when using these two bandpasses. As these breaks are redshifted the signal travels to redder areas of the spectrum and this strong red signal moves to different bandpasses. By identifying in which bandpasses this signal is present and the wavelengths they cover we can get an estimate for the redshift of a galaxy. The 4000 Å break provides information up to $z \sim 1.4$ while the Lyman break appears in optical photometry at $z \sim 2.5$ (LSST Science Collaboration et al. 2009). There are two main types of photo-z techniques that fit the colors of galaxies in different ways, template fitting methods and empirical relationship methods.

1.1.1 Template Fitting Photo-Z

Baum (1962) developed the first photo-z technique by creating SEDs for observed elliptical galaxies out of a fit to nine photometric bands. The general shape of this SED provided an estimate for the observed wavelength of the 4000 Å break for each galaxy and thus a redshift value for each galaxy. This is still the idea for template based photo-z fitting in general. Template fitting methods use a set of rest frame spectra that cover a range of galaxy types. The spectra are redshifted across a grid of redshifts and the colors are calculated in the photometric filters of a given catalog. Then the observed color of a galaxy is compared to template colors resulting in a redshift based upon the redshift of the best fitting template. Koo (1985) used synthetic model galaxy SEDs to create redshift contours through a two-dimensional color space and predict observed redshifts based upon the location in the color space. Loh and Spillar (1986) then used a set of empirical templates for different galaxy types at various redshifts. They then fit observed fluxes from six optical bands by minimizing the chi-square of observations to the fluxes calculated for the fiducial spectra.

The choice of template libraries varies and can include empirical spectra such as the com-

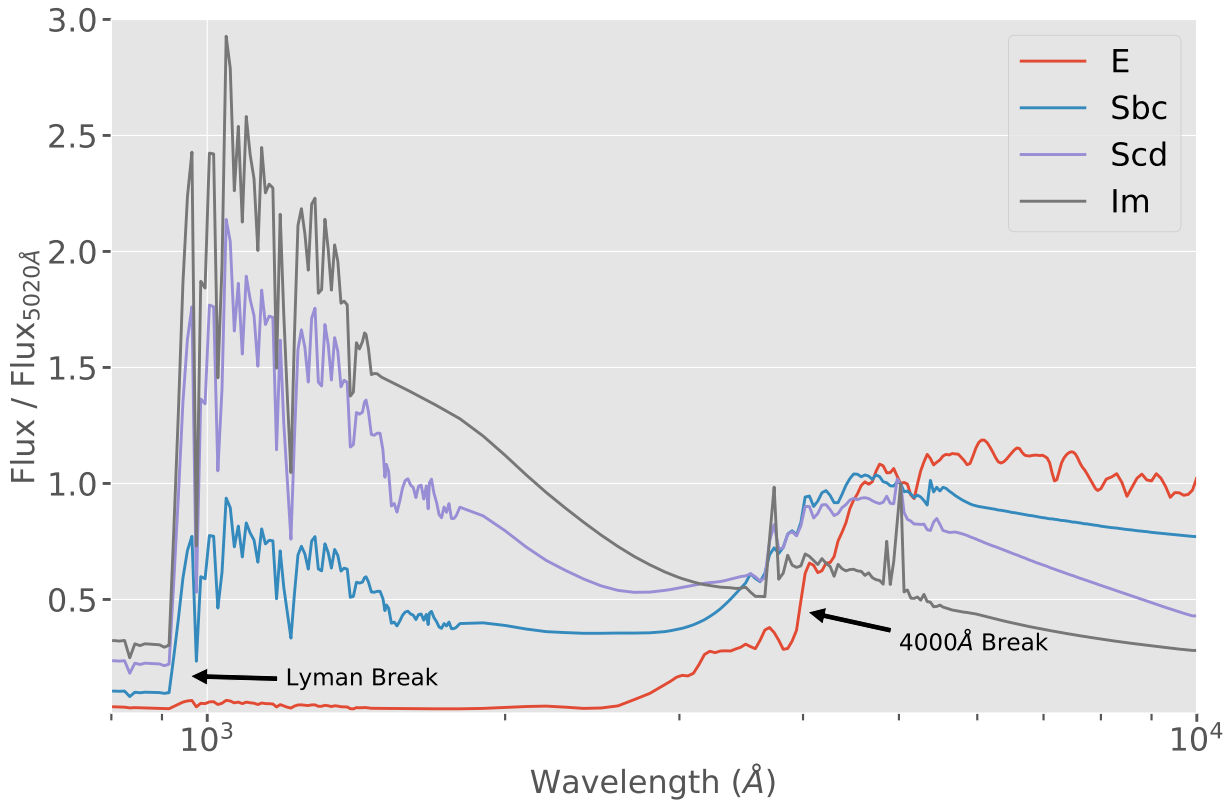

Figure 1.1: The Coleman, Wu, and Weedman (1980) templates extended into the UV and IR by Bolzonella et al. (2000) using the GISSEL code (Bruzual and Charlot 1993, 2003). The templates correspond to four types of galaxy: elliptical (E), bc-type spiral (Sbc), cd-type spiral (Scd) and irregular (Im). Notice the Lyman break below 1000 Å and the sharp break around 4000 Å which is especially obvious in the elliptical template.

monly used templates of Coleman, Wu and Weedman (1980) (CWW) or synthetic libraries such as the GISSEL models of Bruzual and Charlot (1993, 2003) or a combination of the two. Synthetic libraries are popular since sets of templates for a variety of combinations of star formation histories, metallicities and other parameters can be created with detailed physics taken into account. It is common to start with empirical spectra such as the CWW templates and then supplement them with synthetic templates to cover the full color space present in observations. Sawicki et al. (1997) used the GISSEL library to provide bluer templates along with the CWW templates. Another option is to use synthetic libraries to extend the CWW templates into the UV and IR as done in the templates used by Bolzonella et al. (2000) and shown in Figure 1.1.

Straightforward chi-square minimization is still used in some popular codes like *LePhare* (Arnouts et al. 1999; Ilbert et al. 2006) and *hyperz* (Bolzonella et al. 2000) but more recently attempts to combine template fitting with other information have been developed. For instance, the Bayesian Photometric Redshift (*BPZ*) code of Benitez (2000) uses Bayesian methods to incorporate priors into a template fitting framework. *EAZY* (Brammer et al. 2008) also incorporates priors in addition to allowing the best fit template to actually be a linear combination of the original template set.

There are two main problems that arise when using template fitting as mentioned in Benitez (2000). The first is color-redshift degeneracy between templates at very different redshifts. This mainly occurs when low redshift galaxies with colors that are the result of the 4000 Å break are matched to a template with similar colors at high redshift where the main feature in the optical range in the Lyman break. The reverse situation of high redshift galaxy and low redshift template also occurs. Bayesian priors in redshift and magnitude can, however, be used to help alleviate this issue. This problem is a problem with photo-z in general but the second issue is specific to the template fitting method. When using templates there are regions in color space where observed galaxy colors exist but no templates are near so matching to the nearest templates likely introduces errors. This challenge is targeted by the work in Chapter 3.

1.1.2 Empirical Photo-Z

The second class of photometric redshift methods are empirical methods that use the photometry for a set of galaxies with spectroscopic redshifts as a training set to map out a relationship between the fluxes or colors of these galaxies and the redshifts. This was pioneered with the work of Connolly et al. (1995) where magnitudes in four photometric passbands were used to fit a quadratic function to the redshift of the galaxies. They also did further analysis and identified that the main feature that changed the colors of the low redshift galaxies ($z \lesssim 0.6$) as a function of redshift was indeed the 4000 Å break in the spectra.

More recently advances in data science and larger spectroscopic training sets have led to a proliferation of techniques that use more advanced statistical methods or machine learning to relate redshift and photometry. For example, using the development of new matrix inversion techniques and the Sloan Digital Sky Survey (SDSS) data Way et al. (2009) used Gaussian process regression (GPR) to fit the photometry and some additional observational features of galaxies to redshift. Almosallam et al. (2016) developed the code *GPz* to apply GPR to galaxy photometry while also including the heteroscedastic errors of the photometry. One of the main features of GPR is that the output predictions have a prediction of the variance for each point. Including the photometric errors allows *GPz* to better model the variance of its outputs and therefore Almosallam et al. (2016) suggest the predicted variance functions as a quality metric on the estimated redshifts. If users need a dataset with high confidence in the photometric redshifts they can cut on the predicted variance to select a high grade sample.

Random forests are a popular machine learning technique based upon the use of decision trees. Decision trees take the input data of a set of objects and attempt to split the data along one of the input parameters that maximizes the classification of the data according to a chosen metric. This technique can be extended to regression by using the training values of the continuous output variable as the output classes. Random forests are an extension of decision trees that use bootstrapping to generate multiple datasets from subsets of the data

and fit decision trees for all of these bootstrapped samples while also randomly selecting subsets of the input features for the split at each level of the tree. The final prediction for a random forest regression is the average predicted value across the trees in the random forest. Carliles et al. (2010) first used random forests to estimate redshifts using SDSS data. They also used the ensemble of the predictions for the data from the random forest to generate per-object uncertainties on the photo-z estimates. *TPZ* (Carrasco Kind and Brunner 2013) is a publicly available python code for photo-z that uses prediction trees and random forests as well. The additional development present in *TPZ* is that instead of a simple error estimate for each source, *TPZ* is able to provide a full probability density function (PDF) for each photo-z.

Finally, artificial neural networks are on the cutting edge of machine learning technology today and have been applied to photometric redshift estimation for over a decade since the work of Firth et al. (2003). Neural networks consist of layers of interconnected nodes which contain real number values and the connections define the weights for each node as the values are "transmitted" to a node in another layer. The first layer is the input layer where the nodes consist of the input values of the data as well as a bias node and the final layer is the output layer which in this case usually consists of a single node containing a value for the redshift. In between these are "hidden" layers of nodes connected to the layers before and after it in the network architecture. The first publicly released artificial neural network code for photo-z was *ANNz* (Collister and Lahav 2004). This was followed up more recently by another code, *ANNz2* (Sadeh et al. 2016), which is able to also provide PDF estimates for photo-z. The latest developments from this area of photo-z methodology are efforts to use postage stamp images of the galaxies themselves in the photometric filters as input to deep neural networks (Hoyle 2016; Pasquet et al. 2019). This requires a massive increase in computational power but is made possible through the development of neural network architectures increasingly being able to use Graphical Processing Units (GPUs) and even Tensor Processing Units (TPUs) designed specifically for high performance with neural networks. The tradeoff in computational resources is countered by the fact that the images in

addition to having all the photometric magnitude information for the galaxies also contains any additional features that would be selected for an input layer and leaves none of these features out.

Shortcomings of empirical based methods are related to the properties of the training data that these methods require. These methods depend upon a training sample of spectroscopic redshifts that cover the color-magnitude space of the observed galaxies. If the spectroscopic training set does not properly match the observations then this introduces bias in the results. Furthermore, methods like artificial neural networks are very good at fitting to available data but poor at extrapolating beyond the limits of the training data (Firth et al. 2003; Collister and Lahav 2004). The need for well sampled training data makes empirical based methods a poor choice for very high redshift and faint galaxies where spectroscopic data sets are sparse.

1.2 Cosmology Motivations for improved Photometric Redshifts

Photometric redshifts are our way to add a third dimension to the location of galaxies in large scale photometric surveys and thus essential to mapping out the large scale structure of the universe around us. Measurements of cosmological parameters that rely upon large numbers of galaxies only available through photometric surveys require high precision photo-z measurements. Errors and biases in photo-z will inevitably manifest themselves in the resulting cosmological parameters derived by these methods.

1.2.1 Baryon Acoustic Oscillations

Measuring Baryon Acoustic Oscillations (BAO) is one way to derive cosmological parameters like the Hubble parameter H(z) or angular diameter distance $D_A(z)$ as a function of redshift but requires large survey areas to get measurements to the level needed to successfully detect the BAO signal (Seo and Eisenstein 2003). Eisenstein et al. (2005) were able to successfully detect the BAO peak for the first time in data from the Sloan Digital Sky Survey using spectroscopic redshifts. Attempts to detect and measure from SDSS with photometric redshifts followed but were of significantly less confidence (Blake et al. 2007; Padmanabhan

et al. 2007).

One of the major challenges for measuring cosmological parameters from the BAO signal in photometric surveys is in fact the precision of the photo-z values. Errors in redshift effectively broaden observations of actual clustering and thus lower any clustering signal measured. For this reason, measuring the BAO signal is much harder with photo-z and improvements to photo-z precision in a given survey can pay off in large improvements for BAO measurements. In fact, Seo and Eisenstein (2003) calculated that errors in $D_A(z)$ from photo-z scale as $(\frac{\sigma_z}{(1+z)})^{\frac{1}{2}}$ and thus improving photo-z scatter from the 4% level in (1+z) to the 1% level improves angular diameter distance measurement errors by a factor of 2. While a four times improvement in photo-z error is a lot the fact of the matter is that any improvements to photo-z precision directly lead to an increase in the ability to gather cosmological information from measuring baryon acoustic oscillations.

1.2.2 Weak Lensing

Gravitational lensing occurs when light from galaxies behind foreground objects is gravitationally deflected back towards the observer. In the weak lensing regime this manifests itself in distortions of the measured shape of background galaxies. These effects allow observers to map out dark matter in the foreground regions as well as measure dark energy parameters. The lensing signal visible in the shapes of galaxies is very small and using as many galaxies as possible is essential to increase the signal-to-noise of these measurements (LSST Science Collaboration et al. 2009). As a result, photometric galaxy surveys with photo-z measurements are essential since they provide many more galaxies with redshift measurements than is possible with spectroscopic redshifts.

The higher photometric redshift uncertainties, however, lead to issues when measuring weak lensing. For instance, measurements of dark energy parameters through weak lensing tomography require grouping galaxies into redshift bins. Photo-z errors put galaxies into the wrong bins and thus introduce a bias into the lensing signals for each bin. Ma et al. (2006) found that for redshift bins of $\delta z = 0.1$ the photo-z bias (where bias is the expected

$z_{true} - z_{photo}$ for a given true redshift) and uncertainty both had to be better than 0.003 - 0.01 in each bin to prevent degradation of the dark energy parameter errors by more than a factor of 1.5. More concerning was that this calculation assumed a much smaller sky area than a survey like LSST and the authors noted that for larger, deeper surveys the requirements become more stringent. This example provides insight into why photo-z improvements in both bias and error metrics are significant needs for weak lensing to perform high precision cosmology in the LSST era.

1.3 Outline

The work in this thesis is focused on using statistics and machine learning to improve the outputs of photo-z methods that already exist. Chapter 2 presents a mathematical formalism and computational method for designing filter sets that optimally measure photometric redshifts. We use this to present a hypothetical six filter set that improves upon LSST photo-z performance. In Chapter 3 we use principal component analysis and Gaussian process regression to expand template sets that are used for photo-z estimation. Data augmentation methods are used in Chapter 4 to enhance performance of photo-z training sets. Finally, we conclude in Chapter 5 and discuss future work that can be done.

Chapter 2

USING INFORMATION THEORY TO OPTIMIZE BANDPASSES FOR PHOTOMETRIC REDSHIFTS

2.1 Introduction

In a seminal work, Shannon (1948) introduced the concept of information theory. While originally concerned with the information content of messages sent along a channel with limited bandwidth and other signal processing problems, applications of information theory now extend to a multitude of fields including astronomy, finance (Ormos and Zibriczky 2015), and genomics (Adami 2004). Information theoretic concepts are now used in astronomy across a wide range of problems. For instance, Weir et al. (1995) worked with decision trees for star/galaxy classification and used the information entropy to inform the class impurity at each branching. In Seehars et al. (2014) the authors utilized information theory to judge the information gain on parameter posteriors from a series of Cosmic Microwave Background experiments. They were also able to separate the information gained from improvements in statistical error to that gained from new data changing the posterior distributions . Finally, Cincotta et al. (1995) proposed the use of Shannon entropy to find the period of astronomical light curves and Graham et al. (2013) extended this to use conditional entropy. When comparing this conditional entropy algorithm to a wide variety of other period finding methods including Lomb-Scargle, Graham et al. (2013) found that the conditional entropy algorithm was the best when looking at performance with regard to period recovery and computation time. In this chapter, we apply information theory to a combination of survey design and photometric redshift estimation.

Photometric redshift estimation uses multiple observations of extragalactic sources, spread across a range of filters or passbands, to derive an approximate redshift for a given source

(Baum 1962; Koo 1985). The accuracy of these redshift estimates is dependent on the position of breaks or features within a source spectrum relative to the passbands of the photometric filters. For example, the 4000 Å break transitions out of the LSST y-band at a redshift of $z \sim 1.5$ resulting in an increase in the uncertainty of the estimated redshifts until the Lyman break enters at $z \sim 2.5$. In principle, the location and shape of a set of filters could be designed to track specific features within a galaxy spectrum and thereby improve the photometric redshift (at least over a narrow range of redshifts). This work attempts to find a principled way to define the photometric redshift performance of optical filters using information theory and, more specifically, information gain, and thereby derive a set of filters that are optimal for a specific set of survey objectives. The information gain method we outline here can be extended in the future to other areas of astronomy where color can be used to classify objects. Here our classes are photometric redshift bins but could easily be used to classify types of stars instead.

We start in §2.2 with a brief primer on information theory before applying the concept to 3 basic examples in the following sections. In §2.7 we apply the technique to design optimal filter sets and in §2.8 we compare photometric redshifts for a simulated catalog using the proposed filter sets versus LSST filters. We discuss our work and future directions for it in Section 2.9 and conclude in Section 2.10.

2.2 Introduction to Information Theory

2.2.1 Entropy

Consider an event Y with a set of n possible outcomes that each occur with probabilities p_1, p_2, p_3, ..., p_n and $\sum_{i=1}^{n} p(y_i) = 1$. How can we measure the amount of choice or uncertainty present in the selection of an outcome? Shannon (1948) concluded that the uncertainty in the observed outcome is given by the entropy (H) of this set of possible outcomes where the entropy is defined as:

$$H(y) = -\sum_{i} p(y_i) \log_2[p(y_i)] \tag{2.1}$$

Some properties that become apparent from Equation 2.1 are that the maximum entropy occurs when all outcomes are equally probable and that entropy becomes zero when all probabilities go to zero except one. Figure 1 shows the entropy when we have two possible choices A and B and how the entropy changes as the probability of getting outcome A changes. When using base 2 in the logarithm then entropy is measured in bits and the entropy represents the average number of binary digits required to encode a set of outcomes from Y.

For instance, imagine we are observing an event that has two possible outcomes that we label A and B. If $p(A) = p(B) = 0.5$ then the entropy calculation tells us that the best representation we can derive will encode $H(A) + H(B) = -2 * 0.5 \log_2(0.5) = 1$ bit on average. Therefore, simply using $A = 0$ and $B = 1$ when reporting a string of results is an optimal encoding since there is a one-to-one relationship with the length of the encoded information and the number of results. However, if we had a situation where $p(A) > p(B)$ we would have an entropy less than 1. According to information theory then the best encoding scheme could encode the results with less than 1 bit on average. To say this in the reverse way, this means that we can represent a string of results with a number of bits smaller than the length of the results string. Unfortunately, knowing the amount of information in the distribution doesn't tell us how to optimally encode information, but a possible method would be to encode strings of consecutive A results with a single bit. This would mean each bit of information on average would represent more than one result.

2.2.2 Conditional Entropy and Information Gain

Lindley (1956) was the first to extend information theory to quantify the information gained from a measurement by measuring how much an experiment reduced entropy. For instance, imagine a community wants to screen its members for an illness and we know it targets primarily individuals over 40. If we only have a list of the members of the community we can only assign the same probability of illness to all members and can do no better than randomly reaching out to individuals in the population. But if we know the ages of the population

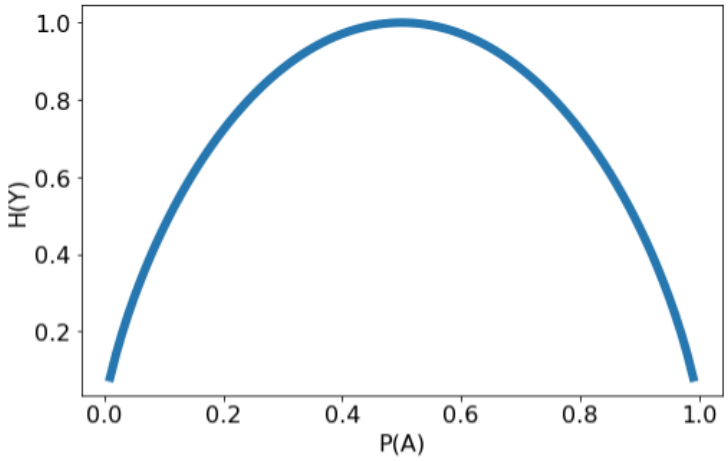

Figure 2.1: Entropy of a system with two outcomes A and B as the probability of getting outcome A changes.

we have more information about whom we should target. Using information theory we can actually measure the information gained when the additional information, in this case the ages of the population, is known. To do so we need to know that conditional entropy is the amount of entropy in an observation of Y when the value of $X = x_j$ is a known quantity. It is defined mathematically in a similar way to entropy:

$$H(Y|X = x_j) = -\sum_i P(y_i|x_j) \log_2[p(y_i|x_j)] \tag{2.2}$$

In our example, Y is the probability of illness in the overall community and X is the age. If we know the overall distribution of X we can calculate the average conditional entropy for the system:

$$H(Y|X) \equiv \sum_{i,j} P(x_j) H(y_i|x_j) \tag{2.3}$$

For the example presented here, we have a different probability for an illness at different ages and this gives us additional information that refines the probability of illness to be more precise for each individual. Therefore, the average conditional entropy will be smaller than the overall entropy since we have less uncertainty in estimates of who might be ill. The

actual information gain (IG) can be found by subtracting the average conditional entropy from the original entropy:

$$IG(Y|X) = H(Y) - H(Y|X) \tag{2.4}$$

To put numbers into our example let's give the overall probability of the illness to be 22%, but for the 60% of the population under 40 the probability becomes only 10%, while for the other 40% of the population it is 40%. Without information on the ages then we have .76 bits of entropy in our estimates of illness ($.22 * \log_2(.22) + .78 * \log_2(.78) = .76$). Adding in the age information gives us an average conditional entropy of $(.6 * (.1 * \log_2(.1) + .9 * \log_2(.9))) + (.4 * (.4 * \log_2(.4) + .6 * \log_2(.6))) = .67$ bits. Therefore, the information gain becomes $.76 - .67 = .09$ bits of information gained when we incorporate age information. In the extreme that an illness hit everyone over 40 and nobody under 40 then age information would provide us with a perfect understanding of who had the disease and who didn't. In this case, average conditional entropy would fall to 0 and we would have information gain equal to the total original entropy. This shows that the more information gain we can derive from a measurement of X then the more this measurement reduces our uncertainty in another property Y.

2.2.3 Application to Astronomical Observation

Often in astronomy, we employ a particular observation (be it photometric, spectroscopic, or other) in order to learn about particular properties of the object we are observing. In the formalism expressed above, our observation (say the magnitude through a particular photometric filter) is given by X, where X represents a continuous distribution of observed values. The intended classification of the object (be it star/quasar classification, galaxy type, photometric redshift, etc.) is represented by the values Y, which may or may not include prior probabilistic information. Given a suitably realistic spectroscopic model of our sample, we can calculate the information gain expected from a given filter set, and use this quantitative measure to optimize our choice of filters for the task. In the following sections,

we will explore the properties of information gain as applied to increasingly more realistic astronomical measurements.

2.3 Toy Example 1: Simple Galaxy Classification

Imagine for the time being that all galaxies have spectral energy distributions (SEDs) which fall precisely in one of two classes: we'll call them "red" and "blue" (see Fig. 2.2, upper panel). We'll denote this spectral type by the label Y, which can take on the values $Y \in \{y_r, y_b\}$. Furthermore, imagine that any galaxy has an even chance of being either red or blue. Mathematically stated, this means that $P(y_r) = P(y_b) = 0.5$. From Equation 2.1 we can quickly compute the entropy $H(Y) = 1$.

Now suppose that an astronomer would like to choose a pair of filters, the magnitude difference (i.e. color) of which will give maximal discrimination between the two types of galaxies. Heuristically, it is clear that placement of one filter toward the left, and another toward the right accomplishes this: the difference between the filter magnitudes gives a positive (red) color for spectrum y_r, and a negative (blue) color for spectrum y_b, leading to an ability to easily distinguish between the two spectra.

This conclusion can be reached in a quantitative fashion by computing the information gain for color measurements through the two filters at various locations as shown in the lower panel of Figure 2.2. To construct this surface, we assume trapezoidal filters (see the upper panel of Figure 2.2) of total width 100 nanometers, containing sloped wings of width 25 nanometers, and numerically compute magnitudes through each filter. We also include a realistic CCD response function that accounts for the curved edges noticeable in the blue end filter and the very top of the red filter. We normalize the spectra to $i = 22.0$ and include a sky background normalized at $i = 20.47$. Finally, we assume a single visit of an LSST-like telescope to calculate the magnitudes and the signal to noise ratio (SNR) of each filter. Subtracting the two magnitude values for each spectrum gives us the respective colors with a defined set of filters. We use the SNR to calculate the expected uncertainty around each color measurement. This will give us a Gaussian distribution for the colors of each

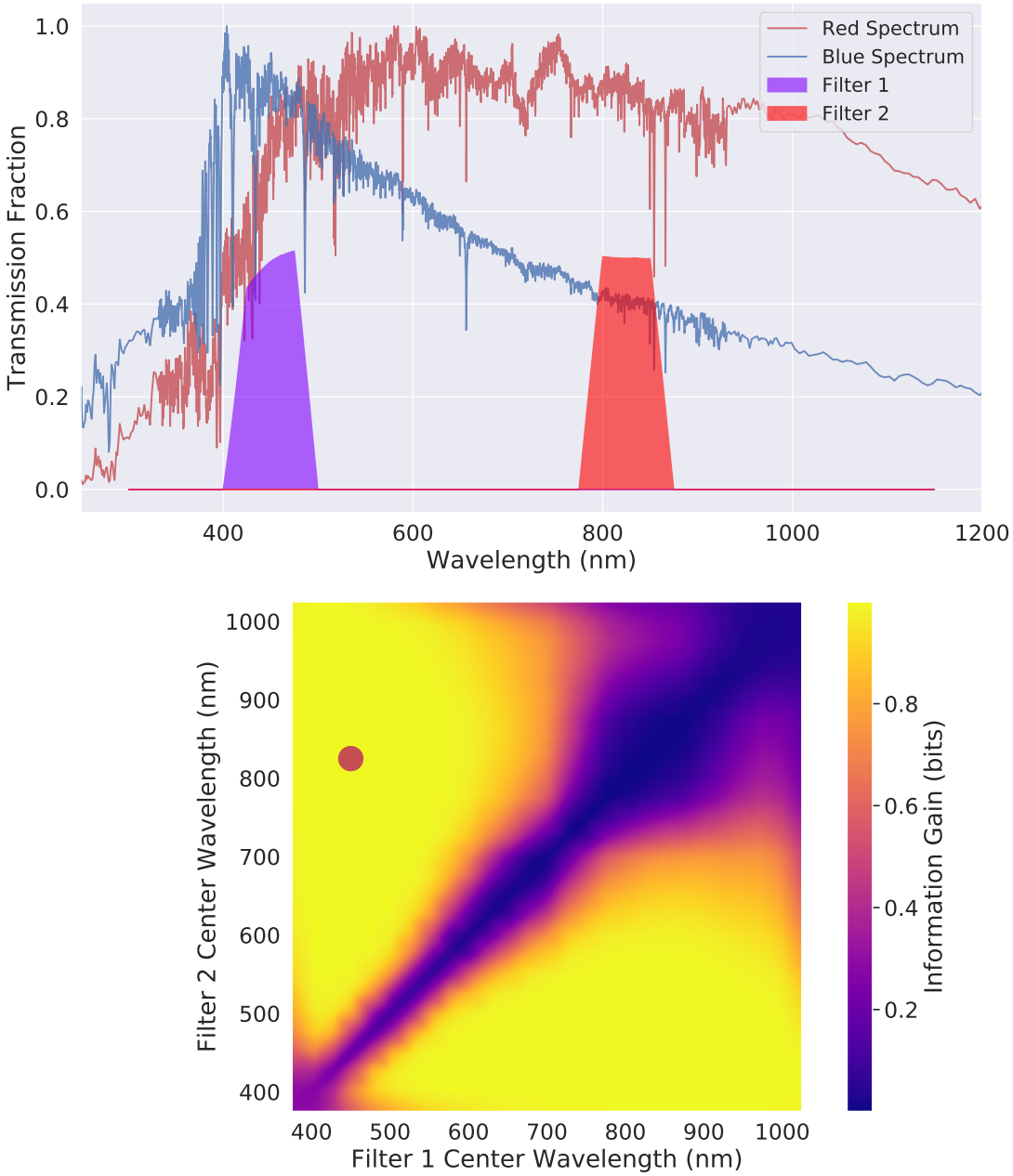

Figure 2.2: Top: Optimal filters for differentiating between equally probable red and blue spectra. Bottom: The information gain as a function of central wavelength of each filter. Notice that when the filters are nearly identical the information gain tends towards 0. The maximum information gain filters shown in the top panel are located at the red point with an information gain of over 0.99 bits.

spectrum in a given filter set. The conditional entropy is then calculated by measuring how much the two color distributions overlap. What we hope to see is that the filter locations that maximize information gain are those that move the two colors distributions as far apart as possible.

When we look at Figure 2.2 we see exactly that. The top panel shows the maximal information gain filters are located with peaks around 450 and 825 nanometers. The bottom panel is a plot of the information gain as a function of the center wavelength of each filter and displays that we can almost perfectly distinguish one spectrum from the other since our information gain is greater than 0.99 bits out of a possible 1.0. The nearly zero information gain along the diagonal makes sense since this is where the filters lie on top of one another and produce the same measured magnitude on average. This gives us an average observed color value of 0 for each spectrum since the color is the subtraction of the magnitude in one filter from the other. Information gain is near but not completely zero along this axis since the width of the error distribution in the color measurement is different for each spectrum. Therefore, we do have a little bit of information to help label an observation. For instance, if we make an observation with identical filters and get a color value of 0.02 mags and this turns out to be a 3σ measurement for the red spectrum but 5σ for the blue spectrum this provides a small amount of information that increases the probability of this being a red spectrum measurement.

This case showed the basics of the information gain theory with an easy problem. Discriminating between two spectra is something we can easily do without resorting to information gain but finding the filters that help discriminate between galaxies at different redshifts is a more realistic and interesting problem. In the next two sections we move on to two simple examples of optimizing filters for photometric redshift estimation.

2.4 Toy Example 2: Measuring the redshift of a sigmoid spectrum

We can use the same formalism as above to address the question of filter choice for determination of photometric redshifts. In this case the observable Y is the redshift of the galaxy.

Because Y cannot represent a continuous distribution within the information gain framework (note the sums in Equation 2.2 above and see Section 2.6 below for more information), we must bin the result. In practice this is not a problem: using a sufficiently large number of bins will allow the redshift result to be recovered to any reasonable accuracy.

For the sake of descriptive simplicity, we'll begin with a toy model using very simple spectra. Imagine now that every galaxy in the universe has a spectrum given by a sigmoid function:

$$S(\lambda; \lambda_0) = \frac{1}{1 + \exp(\lambda - \lambda_0)} \tag{2.5}$$

This is very close to a step function with $S(\lambda) = 0$ for $\lambda \ll \lambda_0$, and $S(\lambda) = 1$ for $\lambda \gg \lambda_0$. With $\lambda_0 = 364.6\ nm$, this shape mimics the balmer-limit break observed in the spectra of galaxies, from which photometric redshift determination gains significant leverage. Imagine furthermore that these galaxies are located at various redshifts, with a probability distribution given by

$$P(z) \propto z^2 \exp[-(z/z_0)^2]. \tag{2.6}$$

with z_0 set so that the median z is 0.6 (typical of ground-based surveys such as DES (Pogosian et al. 2005)). If we break the redshift range into 40 bins from $0.0 < z \leq 2.$ (giving a bin width $\Delta z = 0.05$) then the information contained in the redshift of a galaxy can be computed to be $H(Z) \approx 4.4$ using Equation 2.1 and the prior. This can be interpreted as saying that on average, 4.4 bits of information are needed to specify the redshift of a particular galaxy in the distribution. Because there are $40 \approx 2^{5.3}$ bins, one might wonder why a full 5.3 bits per galaxy would not be needed to specify the redshift: the reason for this is due to the prior information contained within the probability distribution (Eqn. 2.6), which allows more compact representation of the data.

If we perform a maximization of the information gain (Eqn. 2.4) using the color observed through two filters as in Section 2.3, we obtain the information gain surface shown in the lower panel of Figure 2.3. The location of the optimal filters are much more constrained than

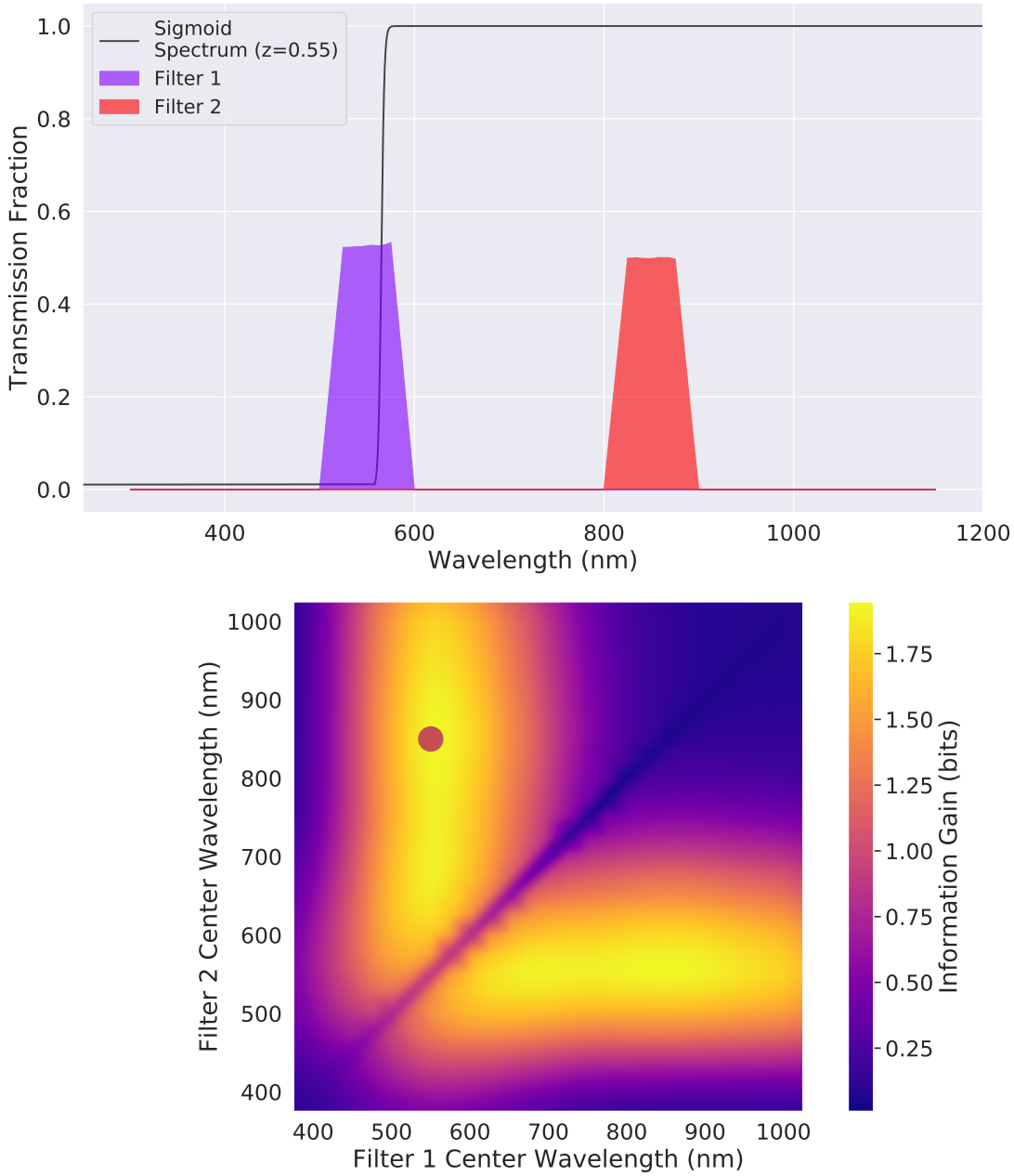

Figure 2.3: Top: Optimal filters for differentiating between the sigmoid spectrum at different redshifts up to z = 2. The sigmoid spectrum is shown at a redshift of 0.55 near the peak of the redshift prior function. Bottom: The information gain as a function of central wavelength of each filter. The maximum information gain filters shown in the top panel are located at the red point with an information gain of ~ 1.95 bits out of a possible 4.4.

in the binary choice example in Section 2.3. Because the redshift distribution peaks near $z = 0.55$, filter combinations where the leftmost filter is centered near 600.0 nm lead to the greatest information gain, as seen in the upper panel of Fig. 2.3. This is because as the break in the spectrum passes through the left filter the magnitude changes at each redshift making the colors in this redshift range very different from one another since the magnitude of the righthand filter is not changing at all. The broadness of the region of maximal information gain shows that there is a large region of the parameter space in which the filter locations lead to nearly maximal information. As long as one filter is located to measure the spectral break near the peak of the redshift prior then the other filter can be shifted left or right over a range of > 200 nm without significantly reducing the information gain.

Quantitatively, the maximal information gain using two filters is ~ 1.95, out of a total information of roughly 4.40. That is, in this simple model, photometric redshifts based on a single color can recover 44% of the redshift information. Most of the lost information exists because we cannot easily differentiate between redshifts close to one another when the break is outside the filters.

2.5 Toy Example 3: Measuring the redshift of a single galaxy

Though the sigmoid spectrum explored in Section 2.4 gives some interesting insight, realistic spectra have many more features in addition to the Balmer break. In this section, we explore a similar example using a single red synthetic galaxy spectrum.

Figure 2.4 shows the equivalent of Figure 2.3 for this more realistic spectrum. The spectrum is that of the red galaxy from Section 2.3 with a strong Balmer break around 400 nm. If the redshift information is coming primarily from this break, we'd expect the optimal filter locations and associated information gain to be similar to that seen in the sigmoid spectrum of Section 2.4.

As before, the information gain surface in Figure 2.4 shows a region of low information gain in the places where the two filters largely overlap. Also like the previous example, the Balmer break is the main factor that determines the locations of the optimal filters. The

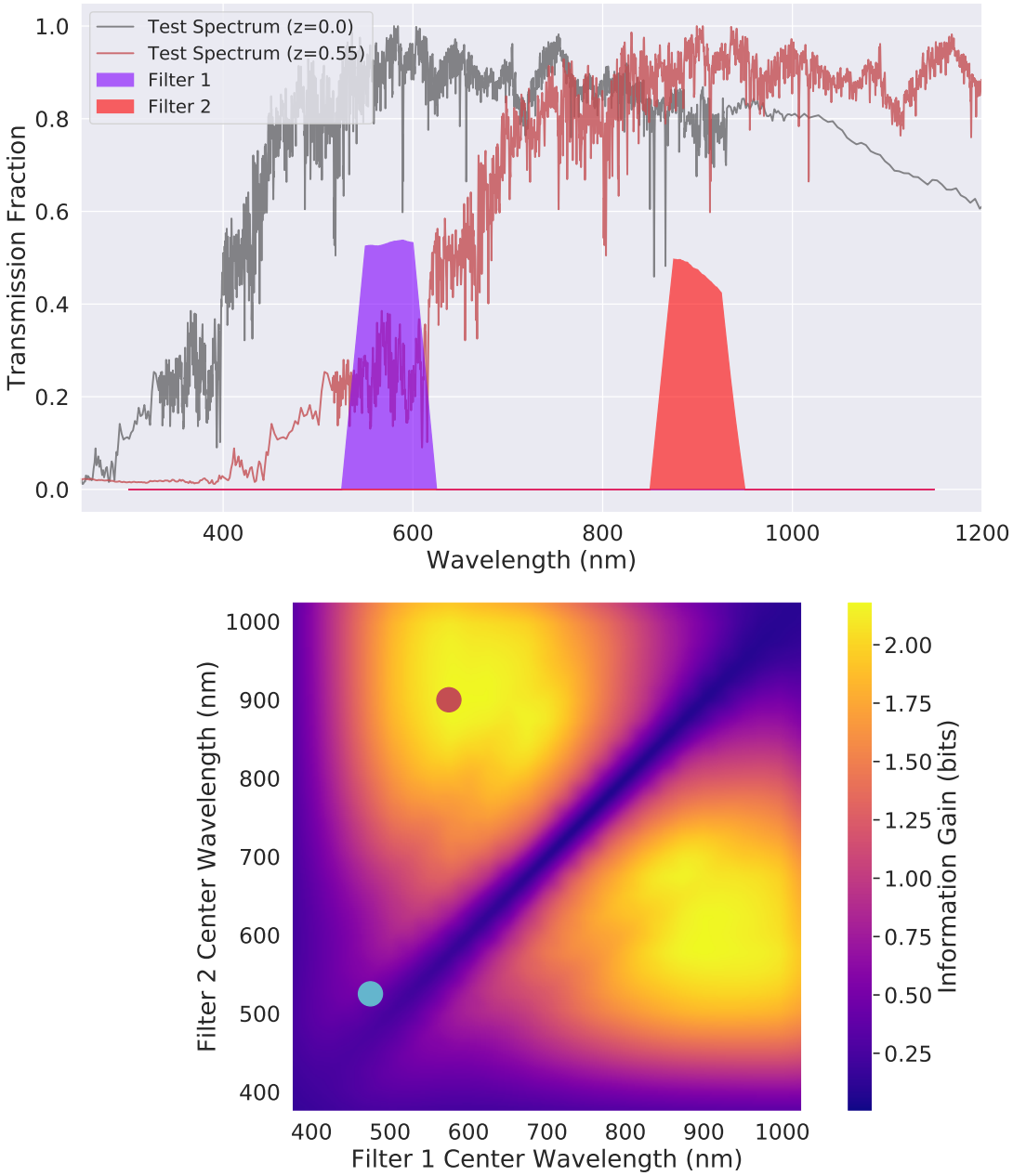

Figure 2.4: Top: Optimal filters for differentiating between the red galaxy spectrum at different redshifts up to z = 2.5. Bottom: The information gain as a function of central wavelength of each filter. The maximum information gain filters shown in the top panel are located at the red point with an information gain of ~ 2.19 bits out of a possible 4.4. The blue point shows the location of the alternate set of filters used in Figure 2.5.

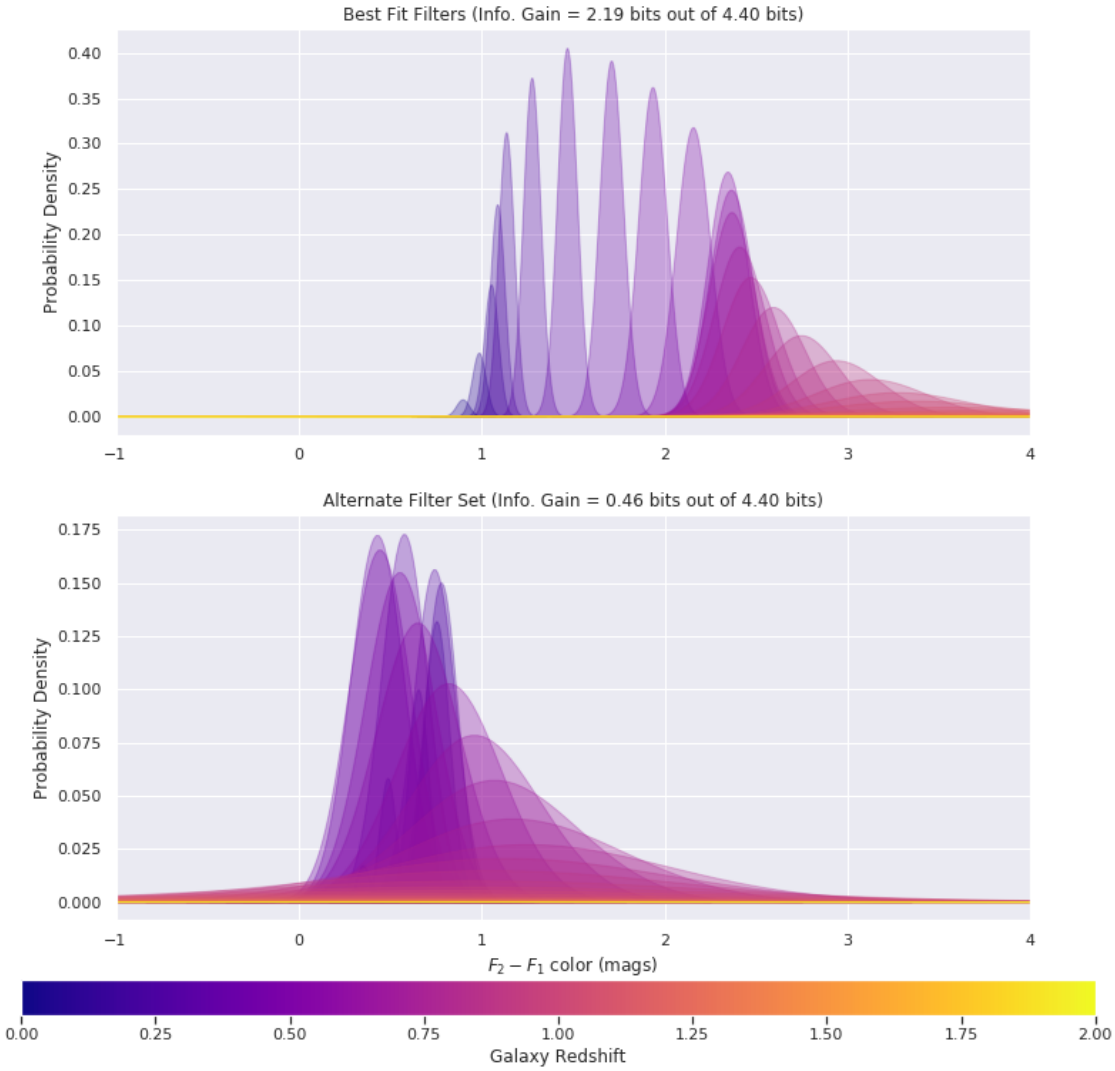

Figure 2.5: Top: The distribution of colors for the red galaxy spectrum at a series of redshifts using the optimal filter scheme. This figure includes the prior on redshift that more strongly weights intermediate redshifts over low and high redshifts. Notice how redshifts near the peak of the prior ($z \sim 0.55$) have the least overlap in their possible color values. Bottom: The distribution of colors using a filter scheme that produces 21% of the optimal information gain (this corresponds to the blue point in Figure 2.4). Here the distributions pile on top of one another and a given color could be the result of the spectrum at a large number of possible redshifts.

bluer filter is once again located in the range where the break is passing through the filter when the spectrum is redshifted to the peak redshift of the prior distribution. The maximal information gain in this case is slightly higher to what we saw previously: ~ 2.19 out of 4.40. This increase is mainly due to the broadness of the Balmer jump (about 100 nm wide here compared to a negligible width previously) which allows a wider range of redshifts to benefit from the change in magnitude of the blue filter as the break passes through it.

To see exactly what is the difference that leads to better information gain in one set of filters versus another we explain the results shown in Figure 2.5. Here the top panel shows the probability distributions of observed colors of the spectrum at each redshift using the optimal filters from Figure 2.4 weighted by the prior probability at that redshift. These probability distributions are centered at the mean color for the template and the width is a result of photometric uncertainties and is affected by the design of the filters and survey. In the bottom panel we see the same distributions for colors derived using a set of filters that only produced 0.46 bits of information gain and are marked in the lower panel of Figure 2.4 by the blue dot. Notice how much more overlap there is in the distributions for colors at each redshift in the bottom panel. On top where we have higher information gain we can be much more confident that a galaxy measured with a certain color will have a given redshift especially in the redshifts around the peak of the prior distribution at $z \sim 0.55$.

The simple examples shown here help connect the information theory presented to practical results in astronomical terms. However, to fully apply information theory to larger template sets and higher numbers of filters we need to further develop our mathematical approach and build the computational tools that will allow us to perform larger experiments.

2.6 Calculating Information Gain in Practice

To calculate the information gain in more complicated scenarios we needed to develop code that could quickly calculate information gain (IG) based upon multiple colors and redshifts of multiple SEDs. For this purpose we created a python code called *SIGgi* (Kalmbach 2018) (where SIG stands for Spectral Information Gain). In practice we calculate IG starting

from calculating the average conditional entropy $H(Y|X)$ where we rewrite it by combining Equations 2.2 and 2.3 along with the identity $P(x_i, y_i) = P(y_i|x_i)P(x_i)$ to get:

$$H(Y|X) = -\sum_{i,j} p(x_j, y_i) \log_2[\frac{p(x_j, y_i)}{p(x_j)}] \tag{2.7}$$

But in our case X is the vector of colors of the SED and is continuous. Therefore, we allow continuous observations by using the Kullback-Leibler (KL) divergence (Kullback and Leibler 1951):

$$D_{KL}(P||Q) = \sum_k p(y_k) \log_2(p(y_k)/q(y_k)) \tag{2.8}$$

If $p(y)$ and $q(y)$ are continuous probability distributions and normalized to 1 across the entirety of y then the KL divergence is:

$$D_{KL}(p||q) = \int p(y) \log_2[p(y)/q(y)]\ dy \tag{2.9}$$

Now we can see that $H(Y|X)$ can be written in terms of the KL divergence as:

$$H(Y|X) = -D_{KL}(p(y_i, x)||p(x)) \tag{2.10}$$

where y remains a discrete variable and thus requires that we continue to bin redshift, but now x is expressed as a continuous observable. Finally we combine this with Equation 2.7 to get:

$$H(Y|X) = -\sum_i \int p(y_i, x) \log_2[\frac{p(y_i, x)}{p(x)}]\ dx \tag{2.11}$$

where i is a particular redshift bin.

So, to compute the conditional entropy we must be able to determine the joint probability $P(y_i, x) = P(y_i)P(x|y_i)$, where each y_i represents a discrete unknown property (e.g. binned redshift), and x is a continuous observable (e.g. photometric colors). $P(y_i)$ is simply the prior distribution of the unknown property, and $P(x|y_i)$ expresses the distribution of observables for a particular input. We have a model to predict this conditional distribution $P(x|y_i)$ of an observation x given a value y_i (for example, we can compute the colors of a galaxy given its redshift). In the case of a single spectrum this distribution $P(x|y_i)$ is assumed

to be normally distributed about a single value. This is because the colors of a galaxy spectrum at a single redshift will have a mean observed value and Gaussian uncertainties in each color. The width of the Gaussian in each dimension will depend on the photometric uncertainty of the measurements in the filters for that color. In the case of multiple spectra the color distribution for a single redshift will be the sum of the normal distributions for each individual spectrum at that redshift.

Because the calculation of conditional entropy via Equation 2.11 involves an integral over a potentially high-dimensional space with very fine resolution, a straightforward numerical integration based on a grid of values becomes too costly to use in practice. For example, a single color for a collection of galaxy spectra in LSST filters may spread over up to two and a half magnitudes (see Figure 2.6). To assure sufficient sampling of the distribution of colors, this requires on order 100 grid divisions per dimension, which leads to on order 10^{10} grid points in five dimensions for a naive implementation. In practice, even this resolution can produce artifacts due to insufficient sampling of the distributions.

However, the calculation of this integral can be done by probabilistically sampling from the color distributions. We start from Equation 2.11 in combination with the approximation $\int p(x)q(x)\,dx = \lim_{N\to\inf} \frac{1}{N} \sum_{x_i\sim p(x)} q(x_i)$ and calculate the following in our code:

$$H(Y|X) = -\frac{1}{N} \sum_{i} \sum_{x_j\sim p(y_i,x_j)}^{n_i} \log_2[\frac{p(y_i,x_j)}{p(x_j)}] \tag{2.12}$$

To evaluate Equation 2.12, we draw $N = 10^6$ points from the prior distribution for redshift $p(y_i)$. This gives us n_i points that fall into each redshift bin that we then use to calculate the inner sum for that bin. For each point in the redshift bin we randomly pick an SED with a uniform probability (a simplification we discuss modifying in future work in Chapter 5). We then draw a vector of colors from the multivariate Gaussian distribution that models the observed photometric color and uncertainties for the redshifted SED. We save all these color points so that we have a representation of the complete color space for that redshift based upon the available galaxy SEDs.

To calculate the sum over the logarithm we need to find the values for $p(y_i, x_j)$ and $p(x_j)$

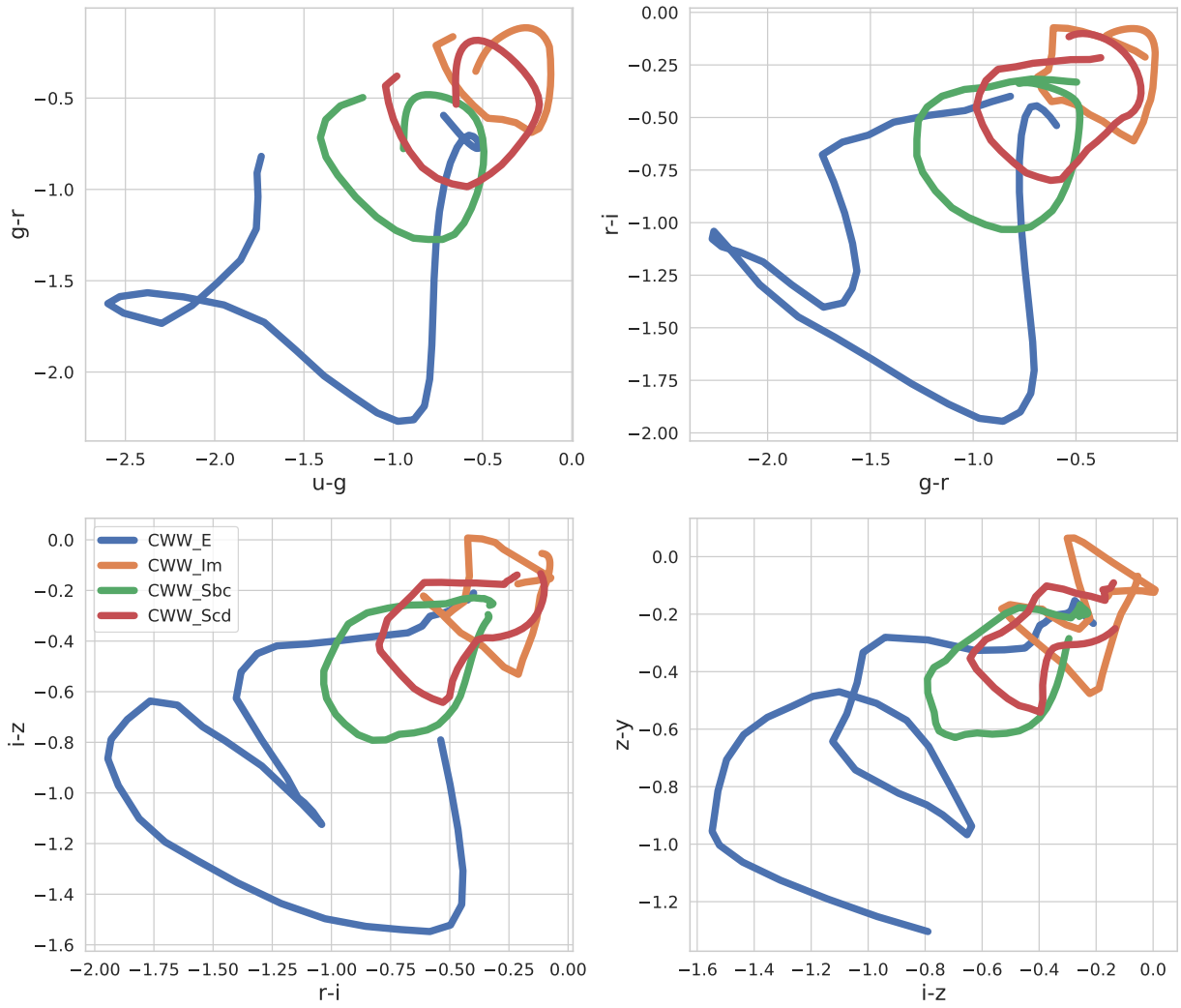

Figure 2.6: Color-color plot of the 4 Coleman et al. (1980) templates in the LSST filters.

where x_j is a point in color space. The value for $p(x_j)$ will be the sum of values measured at x_j from the multivariate Gaussians that are the probability density functions for the colors for each SED at each redshift. To get $p(y_i, x_j)$ this calculation includes only the redshifted SEDs at the specific redshift y_i. We use this technique to sum over the points in each redshift bin and then sum over the values for each redshift bin before normalizing by $\frac{1}{N}$ to get a final answer for $H(Y|X)$. This value for conditional entropy is then subtracted from the full entropy to get the information gain.

2.7 A Realistic Sample

The real world is not nearly as clean as the simple situations discussed above. Rather than observing galaxies of a single spectral type, we observe many different galaxies with different intrinsic spectral characteristics at many different redshifts. Rather than having a single color associated with each redshift, we have a broad distribution of colors associated with each redshift.

To study this, we need a representative sample of spectra which evenly samples the expected space of observations. Since we are interested in the estimation of photometric redshifts we use the commonly utilized template sets from (Coleman et al. 1980) (CWW) and (Calzetti et al. 1994) (Kinney-Calzetti Atlas) supplemented by Arnouts et al. (1999) at UV and IR wavelengths with the GISSEL code (Bruzual and Charlot 1993, 2003). The colors and photometric uncertainties for the Gaussian distributions in color space are calculated based upon normalizing all the SEDs to $i = 25$ and using a sky background set at $i = 20.47$ with an LSST-like telescope over a 10 year LSST-like survey. The sky background is modeled with a sky SED provided in the LSST Sims throughputs package (Connolly et al. 2014). We choose $i = 25$ for our normalization since this approaches the faint limit on the definition of LSST "gold sample" galaxies for photometric redshifts (LSST Science Collaboration et al. 2009). We also use a prior function on redshift that we derive from the photo-z training catalog we describe in Section 2.8.1. This prior came from fitting a function of a similar form to Equation 2.6 on the 39,952 training set galaxies with $24.75 < i_{mag} < 25.$ and is

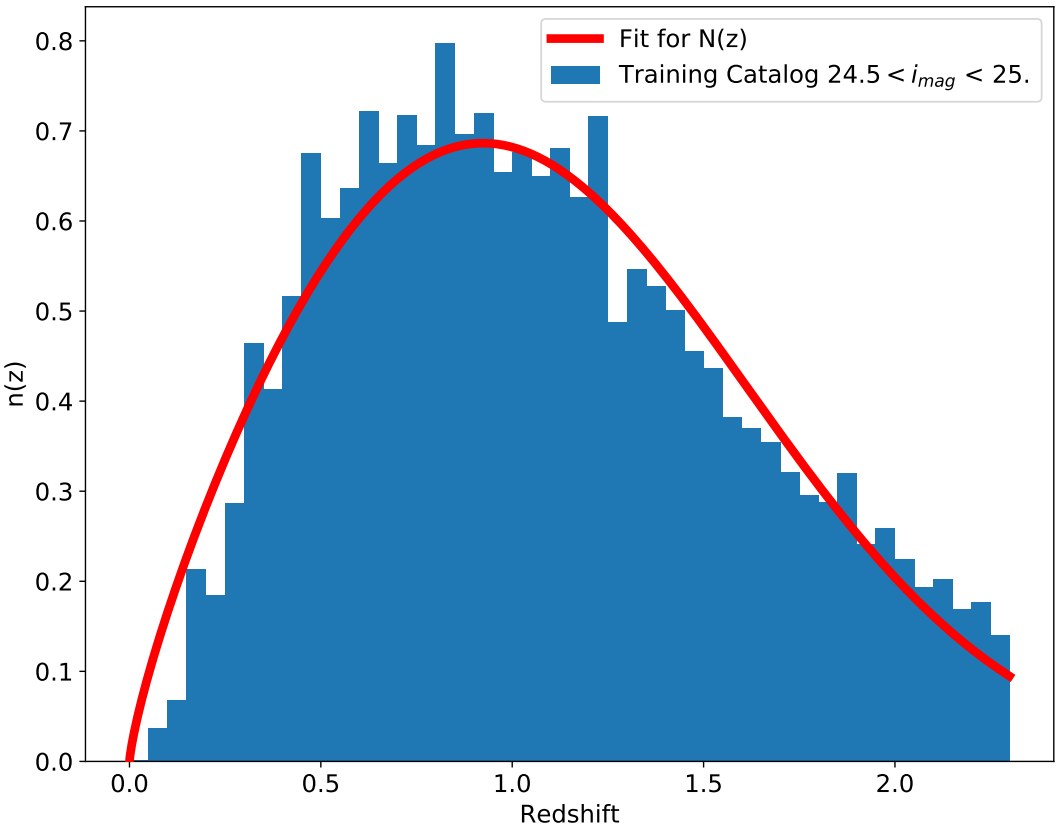

Figure 2.7: Redshift prior derived from training catalog.

designed to match the galaxies expected with the $i = 25$ normalization of the SEDs. The histogram of the galaxies and the prior are shown in Figure 2.7. In our tests, we bin the redshift every 0.05 between $0.0 \leq z \leq 2.3$ giving us 46 total bins. We set the limit at 2.3 because the CWW-Kinney templates blue limit starts to pass into the bluest wavelengths of our filters at this redshift.

Finally, instead of sampling along a grid of set widths and centers as we did in the example problems we use scikit-optimize (Head et al. 2018) to optimize the locations and shapes of

Table 2.1: Number of visits to a field in LSST survey for each filter

Filter	**u**	**g**	**r**	**i**	**z**	**y**
# of visits	56	80	184	184	160	160

our filters. Scikit-optimize is an open source python-based Bayesian optimization package designed to optimize complex spaces such as the high-dimensional information gain space in our problem. We use the Gaussian Process based estimator provided in the code to model the output space and choose locations for optimization. In each run, we initialize the space with 10 points before running the optimization and allow the optimization to run in parallel, updating after running a set number of points independently each time.

2.7.1 Adding a new filter to LSST

In our first experiment we used the LSST filters as a set of fixed filters and wanted to find the optimal additional filter in the optical range that would benefit photometric redshifts the most. For our simulation we gave this filter an equal number of visits as proposed for the LSST y filter and kept the same number of visits for the other filters in effect extending the baseline LSST survey shown in Table 2.1. We allowed the four corners of a trapezoidal filter to move independently in the wavelength range between 300 and 1100 nm. This gave us an optimization with four degrees of freedom that allowed the location and width of the filter to vary as well as the slope of the wings of the filter on each side.

The resulting filter is shown in Figure 2.8 with and without the accompanying LSST filters. The best filter is a large filter with wide wings at the blue and red ends. This filter raises the information gain only slightly from 2.71 bits for the LSST filters alone to 2.83 bits.

This wide filter is centered around the Balmer break at the peak redshift of the prior distribution at $z \sim 0.92$. This is obvious in the top panel of Figure 2.8 where the test SEDs are shown redshifted to this peak value. This seems to confirm what we saw in the examples

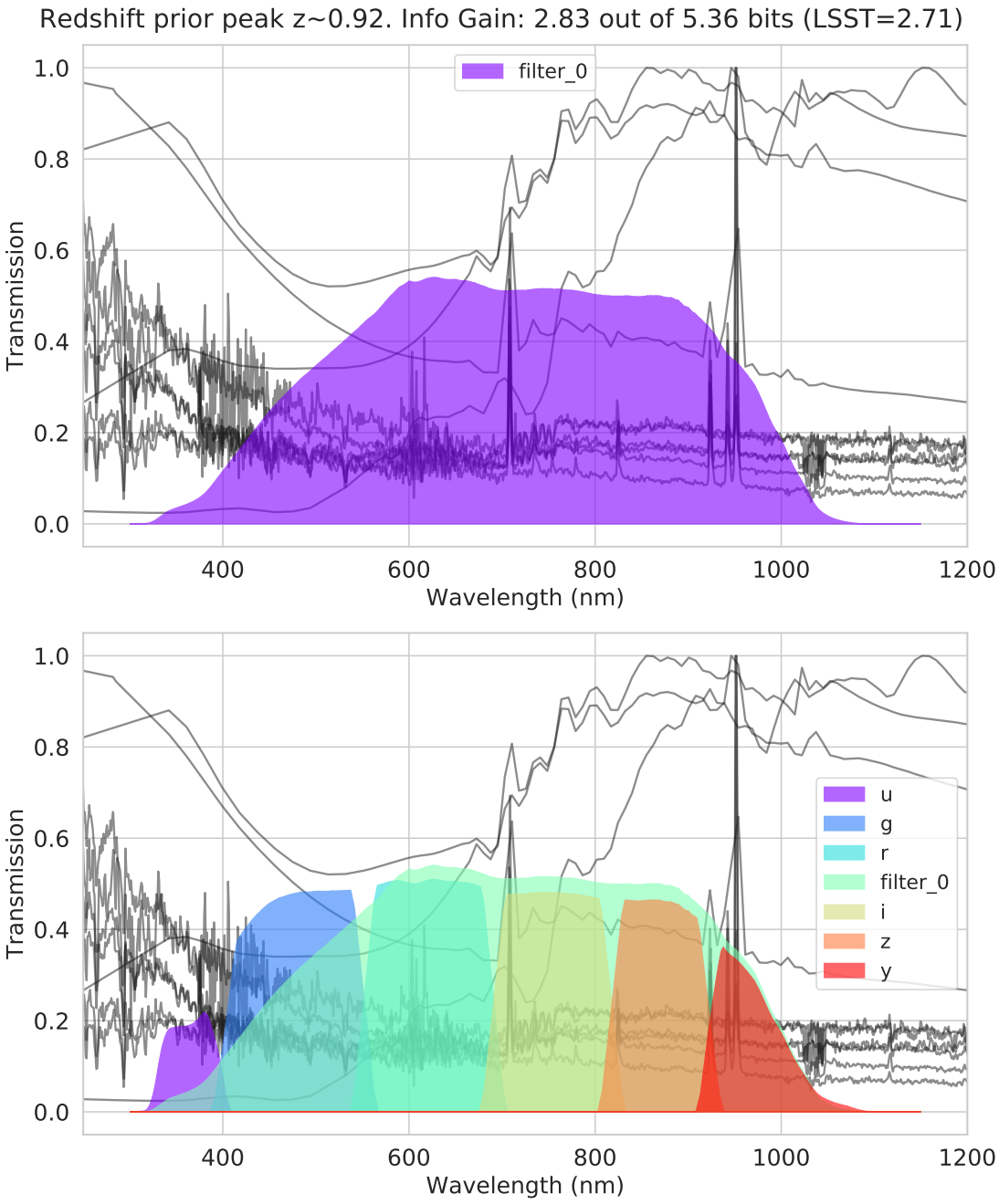

Figure 2.8: Top: The best additional filter added to LSST filters is a wide filter overlapping all the original LSST filters when the template flux normalized to LSST $i = 25$. The CWW-Kinney templates are shown in the background redshifted to the peak of the prior distribution ($z \sim 0.92$). Bottom: The additional filter with the LSST filters provided for comparison.

of the Sections 2.4 and 2.5. Another thing to notice is that the additional information gain provided by a seventh filter to the LSST in the optical range is only a 4% improvement. This indicates that it is difficult to improve the LSST filters by adding wide filters in the optical range.

Effect of changing prior

To verify that the Balmer break is the primary source of information we reran the optimization with a different prior to see how the location of the seventh filter changed. We used the prior from simple examples earlier that peaks at $z \sim 0.55$. The outcome is shown in Figure 2.9 and confirms the shift in the filter location toward the location of the Balmer break at the new peak of the redshift prior. Thus, we observe that filters will constrain redshift the best if they can optimally constrain the location of the Balmer break as it moves across the optical wavelength range.

2.7.2 Six filter survey: Properties of optimal filter sets for photometric redshifts

The locations and shapes of photometric filters affect the colors observed for stars and galaxies. Photometric redshifts rely upon the design of filter systems that will pick up the spectral features for galaxies in the relevant redshift range of a survey. The colors produced by a photometric system are also important for estimating stellar properties (Lenz et al. 1998) and quasar selection (Peters et al. 2015). Here we investigate optimal shapes and locations for photometric redshifts but plan to extend this evaluation to other astronomical problems that require colors in future work.

In this test we decided to run 10 sets of filter optimization allowing the width and locations of six filters to vary for a total of 12 degrees of freedom. In each test we set a different ratio from 0.1 - 1.0 for the top-to-bottom width of a trapezoidal filter. Figure 2.10 compares the allowable shapes for the filters with the most triangular filter having a ratio of 0.1 on the left compared to the most rectangular on the right with a ratio of top width to bottom width of 1.0. We then found the best information gain for each filter shape at the end of the

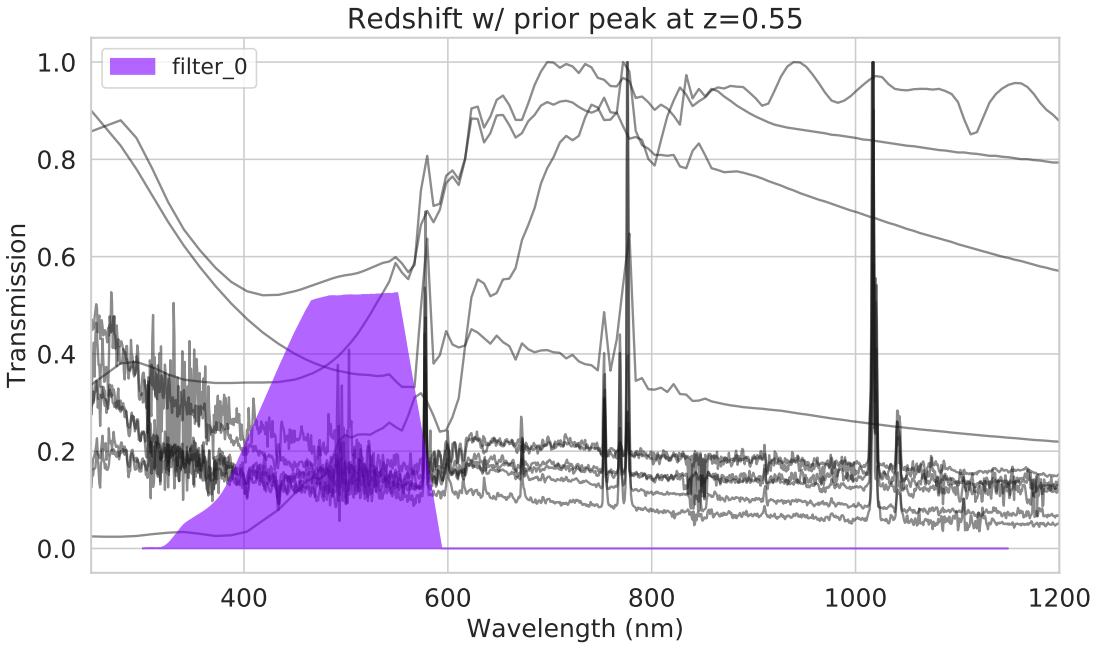

Figure 2.9: The best additional filter when using a redshift prior distribution that peaks at $z \sim 0.55$. The optimal filter is shifted further towards the blue end of the optical range to get information from the Balmer break around the peak redshift of the redshift prior. The SED templates are shown in the background redshifted to $z = 0.55$.

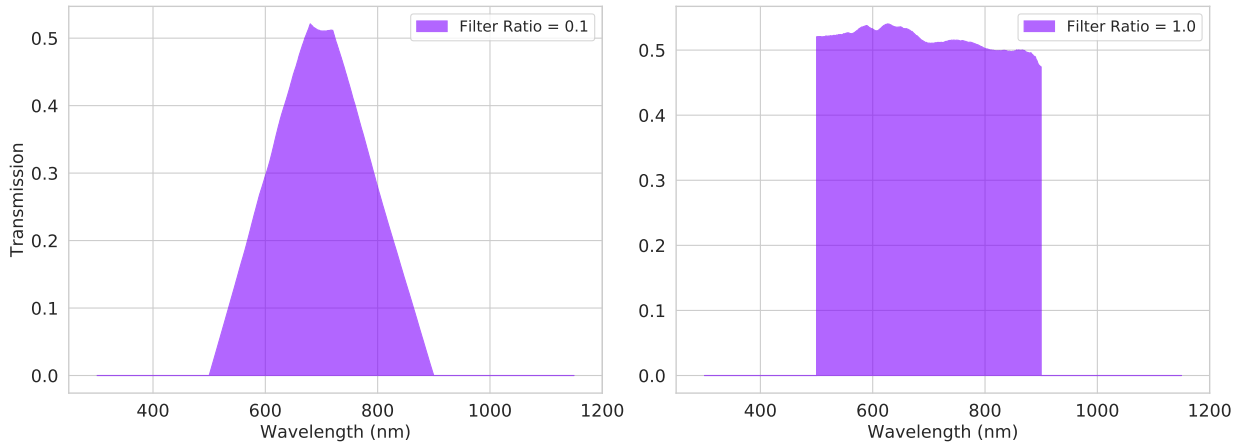

Figure 2.10: A comparison of the allowable filter shapes. Left: A filter with a ratio of top width to bottom width of 0.1. Filters with lower ratios are more triangular. Right: A filter with a top-to-bottom width ratio of 1.0. Filters with higher ratios are more rectangular or "top hat" like.

optimization run and compared the best values for each width ratio. The results are shown in Figure 2.11 and discussed in the sections below. Our best information gain for 6 filters was 2.91 bits which is an increase of 0.2 bits compared to the 2.71 bits of information gain when using the LSST filters.

Filter Shape

The results in Figure 2.11 clearly show a general trend that increasing the slope of the wings of a trapezoidal filter leads to better information gain up to a ratio of 0.9 where the trend flattens out. This overall trend makes sense since the information gain is related to the width of the distributions as we saw in Figure 2.5. There, the better information gain came when the possible distribution of colors for a given redshift was narrower. This means that the width of the color distribution is affected by the signal to noise of the magnitude measurement in each filter. Allowing a wider top of the filter increases the overall transmission for filters

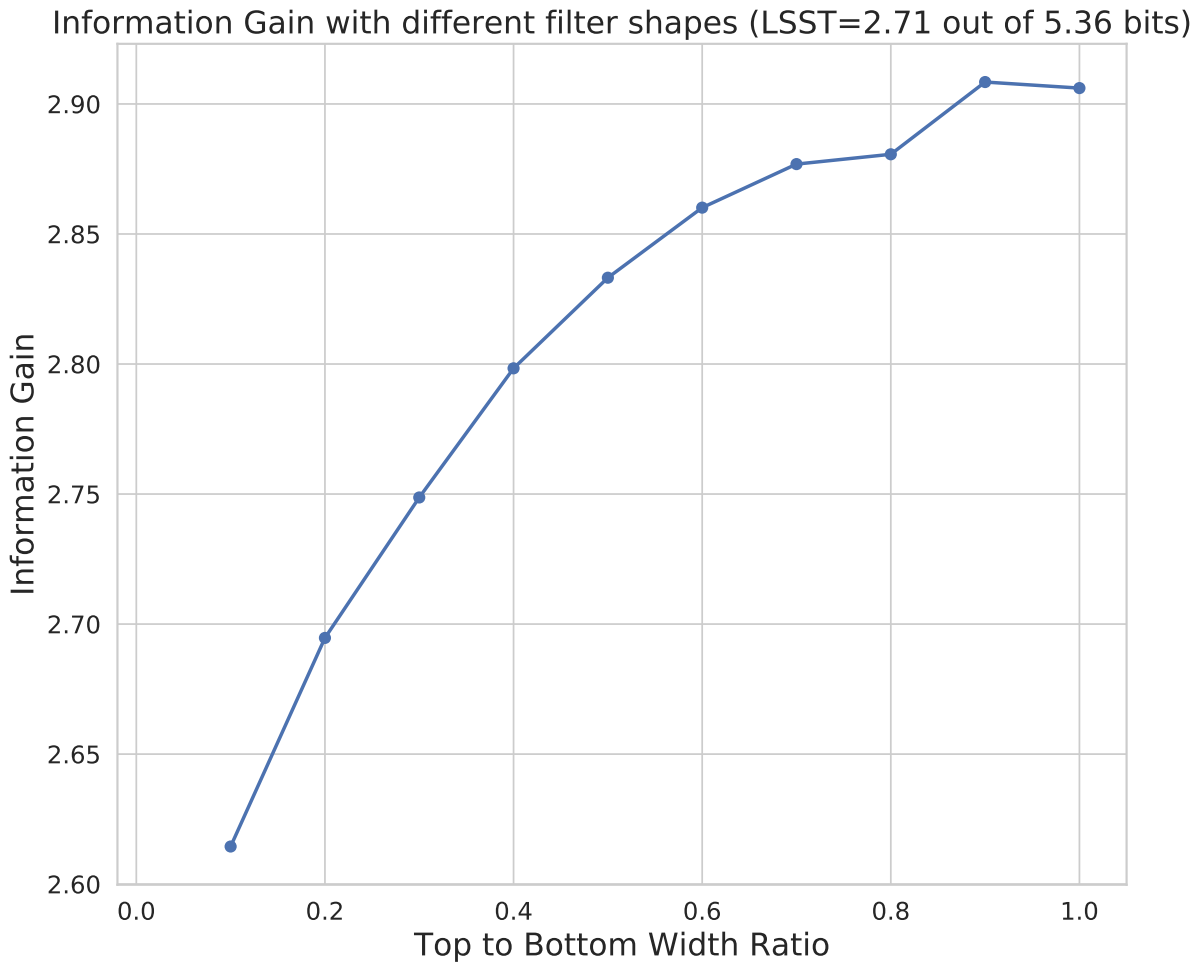

Figure 2.11: The best information gain for a set of trapezoidal filters as a function of the ratio of the width for the top of the filter transmission curve to the bottom width.

of a similar width and thus increases the signal to noise of the flux measurement in that filter for a given spectrum at a given brightness.

Wider tops to the filters also avoids gaps in between filters without the need for a lot of overlap in the wings of each filter. Gaps in the filters allow strong features to fall between filters and wastes information that would otherwise be available. Preventing gaps is necessary to avoid this, but as we will explain in § 2.7.2 some overlap is beneficial but there is a limit. Narrow filter wings avoid the information loss caused by filter gaps while minimizing the extra amount of filter overlap that provides redundant information.

Filter Overlap

Filter sets with overlap perform better since overlapping adds information as to where in a filter strong features appear. If filters do not have any overlap then it is much harder to distinguish at what redshift exactly a strong feature in the spectrum passes from one filter to another. But, if there is a small amount of overlap, then the redshift at which the feature is in the overlap area creates a distinct color value from redshifts slightly higher or lower where it will be in only one filter or another. In our optimal filter set every filter overlaps with its neighbors.

However, too much overlap creates redundant information and stops being beneficial. We set up an extreme situation with a top-to-bottom ratio of 0.9 just like the optimum filters but with an overlap of half of each filter width and shown in Figure 2.12. In this setup every wavelength has coverage in two filters. When we calculate the information gain for this situation it has fallen compared to the optimal filter situation above from 2.91 bits to 2.51 demonstrating that complete overlap in every possible wavelength is not ideal.

2.8 Simulated photometric redshift estimation

To test our new filter sets we created a simulated data set with each new filter set and performed photometric redshift estimation. What we wanted to see was how the improvement in the information gain over the LSST filters translated to photometric redshift performance

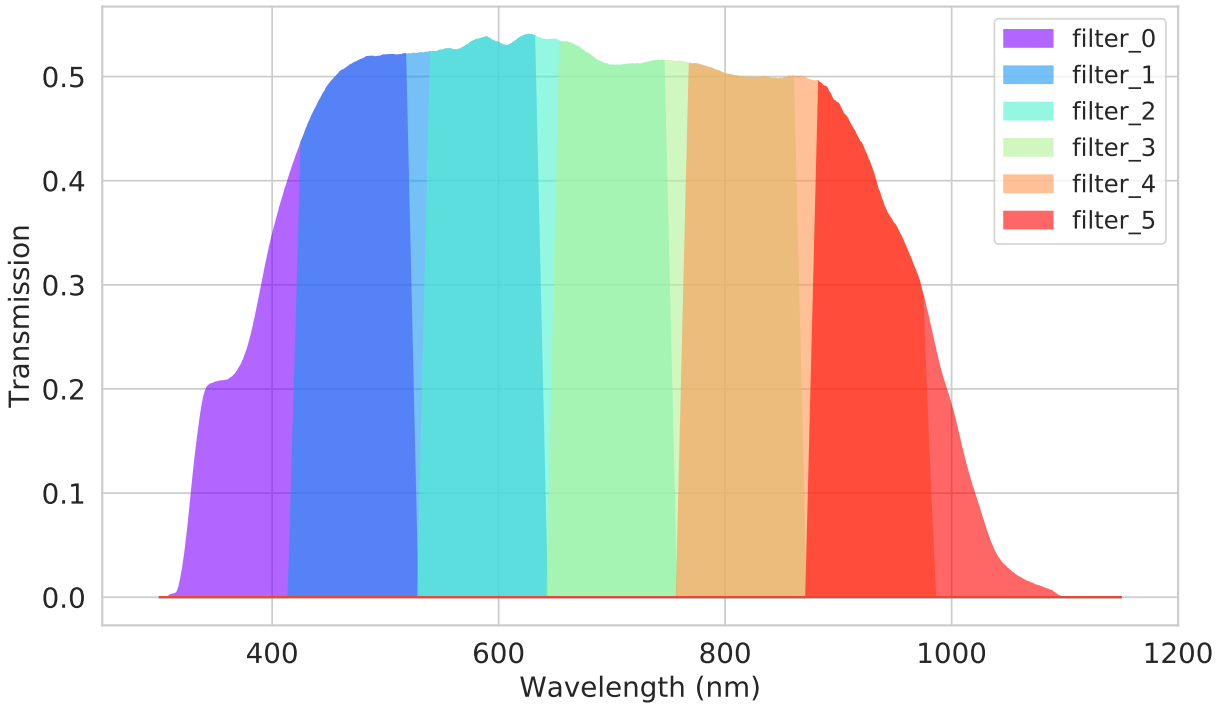

Figure 2.12: 6 filters with 50% overlap of each adjacent filter. The information gain for this situation is only 2.51 bits out of 5.36 possible compared to the 2.91 bits gained with the ideal filter set that has the same top-to-bottom ratio of 0.9.

on a simulated catalog.

2.8.1 Simulated Catalog

We generated simulated catalogs of a circular area on the sky with a radius of 0.8 degrees using the LSST Catalog Simulations (CatSim) code (Connolly et al. 2014). We generated three different catalogs, one each for the different filter sets we want to compare: LSST only, 6 new filters, and LSST+1 filter designs. The LSST CatSim code generates galaxy photometry using templates from Bruzual and Charlot (2003). We ran the code with the different filter sets over the same simulated footprint over a simulated 10 year survey with the same survey properties as given in Graham et al. (2018). Where we included a seventh filter we gave it 160 visits to match the number in the LSST y filter. Following the same procedure as Graham et al. (2018) we included fainter galaxies and used the simulated magnitude errors to apply random normal scatter to the catalog before making a cut at LSST $i < 25$. Then we made a final cut and only kept objects with a redshift <= 2.3 since this was the range of our redshifts when optimizing the filters in Section 2.7. This final cut was then split to give us 50,000 test objects and a training catalog with 245,106 objects in our simulated catalogs.

2.8.2 Calculating Photometric Redshifts

We used the Color Matched Nearest Neighbors (CMNN) redshift estimation code of Graham et al. (2018) on our simulated data. The CMNN photo-z code calculates the Mahalanobis distance in color space between each test galaxy and galaxies in the training catalog. The photo-z value for the test galaxy is then the redshift of nearest neighbor in the training catalog. We ran the CMNN photo-z code on each of our 3 simulated catalogs and compared the results. Figure 2.13 shows the density plots comparing the input catalog redshifts to the photometric redshifts. Between the 2 LSST based filter schemes there does not appear to be much difference in the density plots. The 6 new filters do seem to improve the results at redshifts below a true redshift of 0.6 where there is a clear degeneracy between redshifts of around 0.6 and 0.2 in the LSST filters visible in a cross like feature in the density plots.

Beyond this the density plots once again look similar to the LSST except there does seem to be more scatter at redshifts greater than 1.5 where the Balmer break leaves the optical range. Since we have previously shown that the information gain is strongly affected by the Balmer break this is not surprising.

To get a more informative look at the errors we use the photometric redshift error defined in Graham et al. (2018). The photometric redshift error is defined as $\Delta z_{1+z} = (z_{true} - z_{phot})/(1 + z_{phot})$ and we use this to calculate four error statistics. To analyze the error we plot the bias and standard deviation of the Δz_{1+z} values as a function of the true redshift. We also plot a robust standard deviation which is the standard deviation of the interquartile range of the errors multiplied by 1.349 to create a value comparable to a standard deviation. Finally we plot the fraction of outliers with Δz_{1+z} values greater than 0.15. We use 12 bins in the redshift range from 0 to 2.3 and plot the values in Figure 2.14. In Figure 2.15 we compare the differences to the LSST values for each new filter set. The dashed black line is set where the values are equal to the LSST so that above this line the new filter set is worse and below the line the new filters perform better than LSST.

The new 6 filters obtained through information gain optimization do offer more improvement which is also consistent with the greater 7% information gain improvement compared to adding a seventh filter. Overall they improve the standard deviation by 3.1% and the outlier fraction by 7.1%. These gains are driven by improvements for the redshifts below 1.5 and traded for performance losses at higher redshifts. This was noted in the density plots and is consistent with information gain focusing on the presence of the Balmer break in the optical range.

Looking at the figures comparing the redshift errors we see that adding a new optical filter to the LSST filters offers only a slight improvement. There is a slightly lower overall standard deviation around the $z \sim 0.9$ peak of the prior distribution we used and over the whole test set we have a 0.7% improvement in standard deviation. The biggest gain seems to be a 1.2% improvement in overall outlier fraction but gains are modest in general. This is consistent with the small 4% information gain we found when going from 6 LSST filters

to 7.

2.9 Discussion

We were only able to provide small improvements over the LSST filters both in terms of information gain and actual photometric redshift estimation. This is because the LSST filters already have similar features to those we identified as optimal for photometric redshift filters. The LSST filters have no gaps between them and each filter has a small degree of overlap with the adjacent filters. The filters are also nearly top hat in shape with slight wings on each side which is consistent with our findings of optimal filter sets. Our results from adding a seventh filter in the current LSST wavelength range show that another filter in optical wavelengths is not a good way to improve photometric redshifts with LSST. In fact, it seems that since the Balmer break is so important as we have shown then following it into the infrared regime is essential to improve LSST photometric redshifts. This is in line with previous work highlighting the potential photometric redshift improvement from combining LSST observations with infrared data from future space telescope missions (Jain et al. 2015; Rhodes et al. 2017; Graham et al. 2019).

The practical application of the optimal filters in our work shows that our method has merit and can be used to design observations tailored to photometric redshift estimation. However, limiting ourselves to redshifts up to $z = 2.3$ we are not able to provide insight into how filters can improve the Lyman vs Balmer break degeneracy. This is a large problem in photometric redshift estimation and exploring filter design with our method at a larger redshift range could provide interesting insights to this. Gathering templates that apply to the blue end of the optical range beyond $z = 2.3$ and into the redshifts where this is applicable will help us explore the question in the future.

2.10 Conclusion

We have introduced a new technique to apply information theory to the design of filters in order to optimize photometric redshifts. We showed its theory and provided insight into

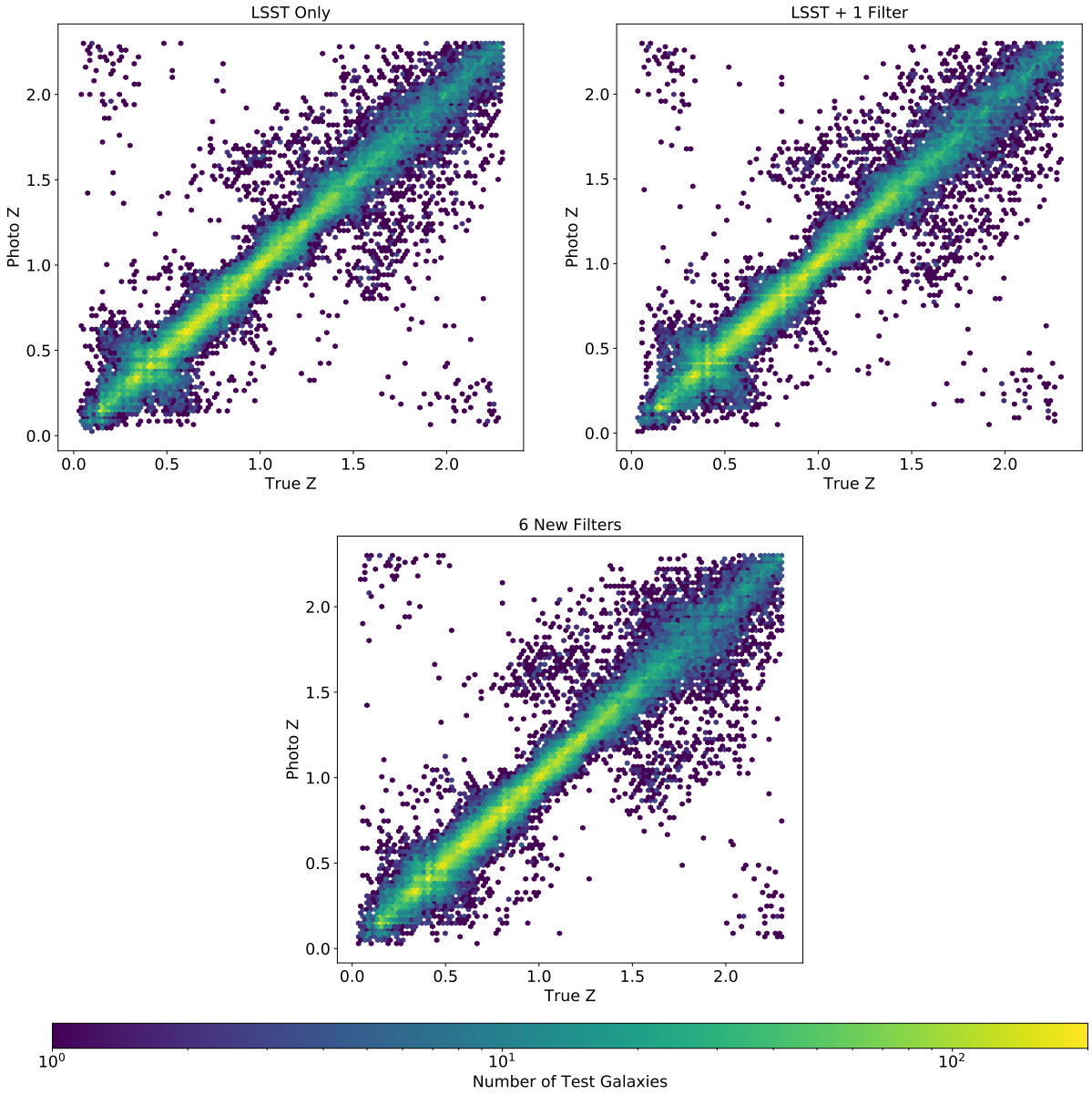

Figure 2.13: Density plots for the results from photometric redshift estimation on the simulated catalogs with the CMNN photo-z code and the different filter sets. Top Left: LSST Filters Only. Top Right: LSST + 1 new filter. Bottom: 6 New Optimized Filters.

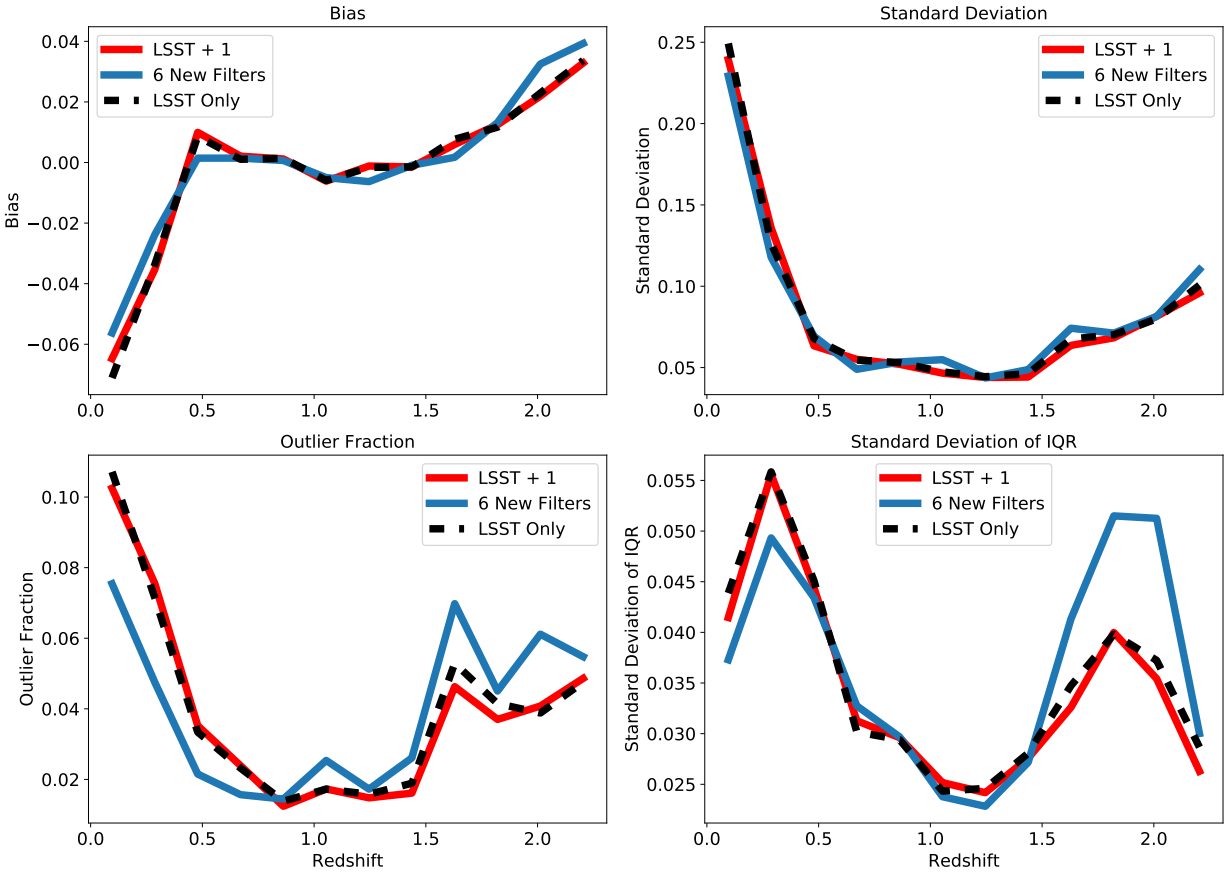

Figure 2.14: Comparing the photometric redshift errors of the 3 different filter sets. As expected from the density plots adding a new filter to LSST does not change much and the 6 new filters reduce outliers at low redshift but trade this for performance at higher redshifts.

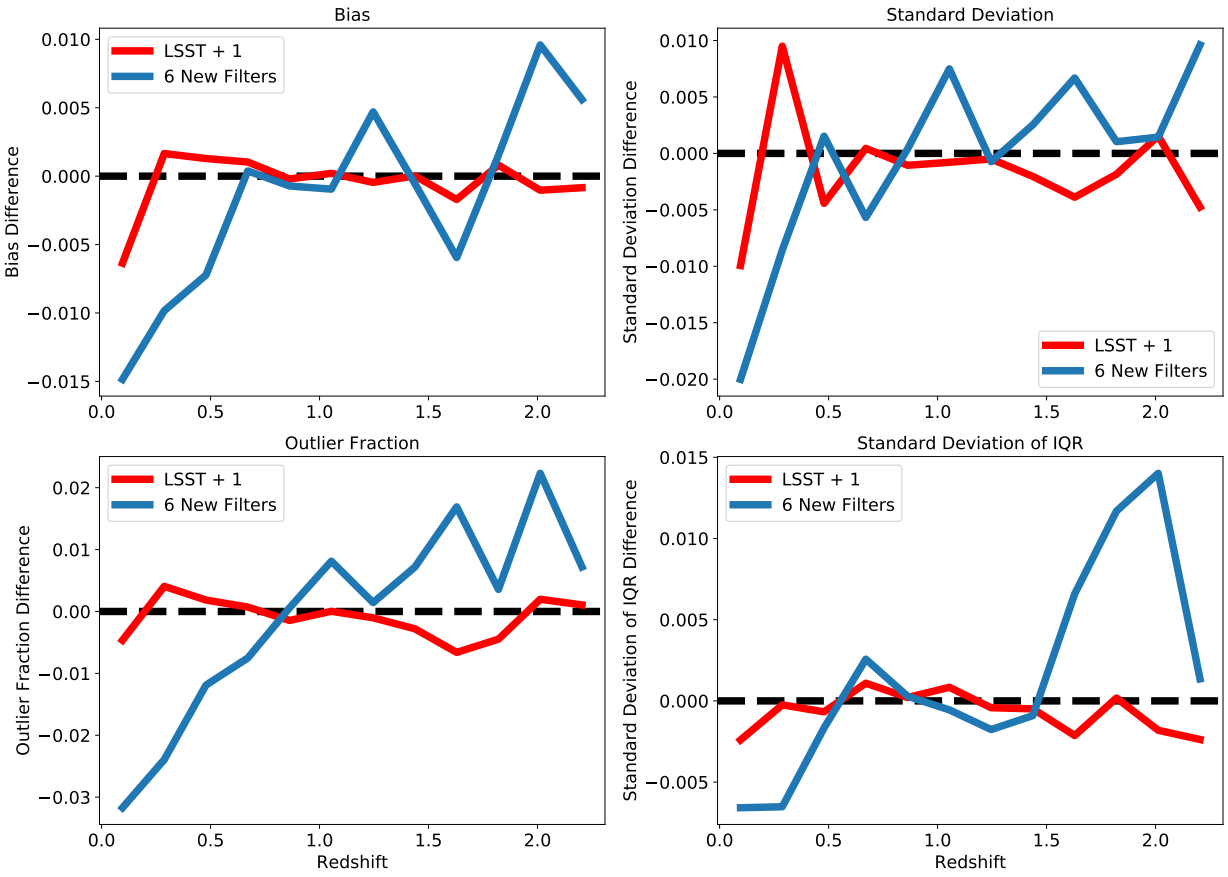

Figure 2.15: Comparing the differences in photometric redshift errors of the 2 new filter sets to performance with the LSST filters only. The black line indicates errors are the same as the LSST filters. Below the black line means improvement over LSST while above the black line indicates worse performance. The added filter seems to slightly improve bias and overall standard deviation around the peak of the redshift prior at 0.9 and helps reduce outliers overall. As noted above the 6 new filters outperform LSST at lower redshifts in return for worse performance at higher redshifts.

the use with three simple examples before using it in a practical situation. We created an optimal set of six filters to cover the optical wavelengths in an ideal manner for photometric redshifts. This application revealed insight into general attributes of an ideal filter set for photometric redshifts. Ideal filters will have narrow wings and be near top hat in shape. They will also have a small amount of overlap. We showed that the main information source for photometric redshift estimation is the Balmer break and optimal filters will focus on maximizing information gain from this source.

We also applied the two different filter sets to a simulated catalog of photometric data and compared the photometric redshift estimation results to the LSST filters. We showed that a set of six filters optimized using information gain could improve the standard deviation of the errors associated with photometric redshifts by 3% overall at redshifts up to 2.3 over the LSST filters and outliers up to 7% but improved performance at lower redshifts was traded for worse results than LSST at higher redshifts. The LSST filters perform near the optimal set and have features we identify as optimal for photo-z filters such as overlap with neighboring filters and a nearly rectangular shape. We also discuss future directions that will improve our technique and will be possible with improvements to the code we used in this work. This python code, *SIGgi*, is publicly available at `https://github.com/jbkalmbach/siggi` and is pip installable.

Chapter 3

EXPANDING TEMPLATE SETS FOR TEMPLATE BASED PHOTO-Z ALGORITHMS

3.1 Introduction

Modeling Spectral Energy Distributions (SEDs) is at the heart of many methods used to study the properties of galaxies. One example is the use of SEDs to estimate galaxy redshifts from photometry rather than spectra (Baum 1962; Koo 1985). Photometric redshift estimation works by establishing a relationship between the colors of galaxies observed in a limited set of filters to the colors derived from a full SED of known redshift. Finding the right SED to compare to the catalog colors can, however, be a challenge. Observed template sets are small because of the need to measure high signal-to-noise spectra across a large wavelength range and into the UV leads to the use of a limited number of local galaxies. For example, the commonly used Coleman et al. (1980) template set is only four templates. Synthetic spectra from models like Bruzual and Charlot (2003) can provide large template sets but have their own problems modeling the UV wavelengths that are important to photometric redshifts as these wavelengths are redshifted into the optical range (Sawicki et al. 1997; Blanton and Roweis 2007). This means that photometric redshift estimation requires matching a continuous distribution of galaxy colors to a finite set of colors from the template SEDs. When there are large gaps between template colors this leads to uncertainty and erroneous matching can lead to catastrophic outliers in the redshift estimates of a catalog. It would therefore be useful to have an interpolation scheme that, given an arbitrary set of color values, could produce a realistic SED corresponding to those colors. Linear interpolation is sometimes used to expand SED sets for photometric redshift estimation as in Gorecki et al. (2014), but this method is not guaranteed to produce realistic SEDs.

Another use for galaxy SEDs is the generation of mock catalogs from galaxy modeling codes. When these codes output a mock catalog they use a set of synthetic SEDs to estimate the colors of the simulated galaxies, but are only able to output SEDs using a grid of values for physical properties (e.g., temperature, metallicity, age) that should realistically be continuous. The ability to create a smooth and reasonable interpolation of SEDs could lead to more realistic color statistics for these mock catalogs.

In this chapter, we introduce a method for SED interpolation that aims at realistically producing SEDs across continuous regions of color space. We will show that our method reproduces the colors of SEDs better than other interpolation methods such as nearest neighbor or linear interpolation. In Sections 3.2 and 3.3 we explain Principal Component Analysis and Gaussian processes and how we apply them to generate an interpolation scheme in color space for SEDs. In Section 3.4 we present a demonstration of our technique where we generate SEDs at given points in color space and compare the results to other interpolation methods. In Section 3.5 we will use photometric redshift estimation as an example of the benefits of our method. Finally, we conclude in Section 3.6.

3.2 Creation of a basis set using Principal Component Analysis

SED fitting techniques attempt to map observed galaxy colors to intrinsic galaxy properties. For instance, photometric redshift estimation attempts to match galaxy colors to their redshifts. As we will show, the sparseness of the template set and its incomplete coverage in color space can create errors in redshift estimation. In order to improve SED fitting techniques we therefore need to be able to interpolate and extrapolate in color space to create a continuous mapping between colors and SEDs. To begin solving this problem, we choose to create a new basis in Section 3.4.1 from the template SEDs using Principal Component Analysis (PCA). Our goal is to use PCA to create new spectra that match what we expect to find in a region of color space. Since the colors of a galaxy are driven by the features in a spectrum and we expect different basis spectra to contain specific features and we will generate spectra to match desired colors by properly weighting the principal components. Properly weighted

linear combinations of the basis SEDs can then be used to construct galaxy SEDs. These weights will be estimated from the relationship in color space for the PCA coefficients of the original template spectra. Here we describe the creation of the basis set with PCA and in the following section we explain how we estimate new weights for the basis SEDs for a specific location in color space.

3.2.1 Principal component analysis

PCA, also known as the Karhunen-Loève transform, has primarily been applied to SEDs in astronomy for classifications using a small number of coefficients to describe each spectrum (Connolly et al. 1995; Yip et al. 2004). This method identifies the directions of maximal variance in a dataset and in so doing provides a new basis to represent the data along these directions. For our case we call these new bases eigenspectra following the convention of Connolly et al. (1995). Using a linear combination of all these eigenspectra with proper weighting we can reconstruct the original spectra:

$$f_{\lambda i} = \sum_{j=1}^{m} y_{ij} e_{\lambda j} \tag{3.1}$$

where $f_{\lambda i}$ is the flux of the ith original spectrum at wavelength λ, m is the total number of eigenspectra and y_{ij} is the coefficient that is applied to jth eigenspectrum when reconstructing the ith spectrum. If we keep all m eigenspectra we can reconstruct the input model spectra perfectly, but in PCA the eigenvectors are ordered by the variance they describe in the original dataset. If desired, we can truncate the sum in Equation 3.1 at a value $n < m$ where n is the number of components that reproduces the original spectra within some error tolerance.

3.3 Gaussian process Regression for Eigencofficients

Our goal is now to estimate SEDs for points in color space where we want to expand our template set, i.e. generate additional SEDs where the templates do not cover the catalog color space. To do this we want to estimate new PCA coefficients using a regression of the PCA

coefficients on to template SEDs based upon their locations in color space. We use Gaussian processes to do this regression because they perform well in creating smooth, nonlinear functions and are able to handle interpolation and extrapolation. Gaussian Processes have entered into astronomy recently in areas such as photometric redshift estimation (Way et al. 2009; Almosallam et al. 2016; Leistedt and Hogg 2017) and analysis of time series data (Aigrain et al. 2015). We use maximum likelihood estimation to quickly optimize a new regression function for each of the PCA coefficients since the hyperparameters that estimate each coefficient are not expected to be the same.

3.3.1 Gaussian processes

A Gaussian process (GP) is a continuous distribution of random variables where any finite number of points can be described with a joint Gaussian distribution (Rasmussen and Williams 2005). For example, consider the 2-d data vector $\mathbf{y}(\mathbf{x_i}) = (y(x_1), ..., y(x_n))$ which is a single draw of a multivariate Gaussian distribution of dimension n. Since $\mathbf{y}$ is the result of a joint Gaussian distribution at points $x_1, ..., x_n$ it is also the result of a GP that can be sampled continuously along the x axis (Ebden 2015). GPs are described by a mean function, $m(\mathbf{x})$, and a covariance function, $k(\mathbf{x}, \mathbf{x}')$, so that a GP can be abbreviated as $f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$ (Rasmussen and Williams 2005). In general and in our use of GPs, the mean function is assumed to be zero so that the GP can be completely specified by the covariance function that relates data points to one another. Putting this together we can use a GP associated with an observed dataset $\mathbf{y} \sim \mathcal{N}(0, k(\mathbf{x}, \mathbf{x}'))$ to make predictions about the values of other points in the same space as $\mathbf{y}(\mathbf{x})$. This is the procedure known as Gaussian process regression and is better known in some fields as kriging. If the data are noisy we can follow Ebden (2015) and add in a Gaussian noise component along with the covariance function giving

$$K(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}') + \sigma_n^2 \delta(\mathbf{x}, \mathbf{x}') \tag{3.2}$$

where $\delta(x, x')$ is the Kronecker delta. Here we assume that our noise is Gaussian and independent, but if there is covariance in the noise we could include additional terms in the covariance function with the desired form (Rasmussen and Williams 2005). Thus, our problem that we show in practice in Section 3.4.2 can now be described with our observed data sample $\mathbf{y}$ (in this case the PCA coefficients) and a test sample $\mathbf{y}_*$ (our interpolated coefficients) as:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}_* \end{bmatrix} \sim \mathcal{N}(0, \begin{bmatrix} K & K_*^T \\ K_* & K_{**} \end{bmatrix}) \tag{3.3}$$

where K is the covariance matrix of the data, K_* is a matrix containing the covariances of the data and the test points and K_{**} is the covariance matrix between the test points.

Since we are interested in using this for regression we want to find the conditional probability of a set of test points given our observations, $p(\mathbf{y}_*|\mathbf{y})$. Using the equations from Appendices 2 and 3 in Rasmussen and Williams (2005) this turns out to be another Gaussian distribution:

$$\mathbf{y}_*|\mathbf{y} \sim \mathcal{N}(K_* K^{-1}\mathbf{y}, K_{**} - K_* K^{-1} K_*^T) \tag{3.4}$$

The mean of the new Gaussian at $\mathbf{x}_*$ is the best estimate for the value of $\mathbf{y}_*$ while the variance of the new Gaussian at that point provides a measure of the uncertainty in the predicted $\mathbf{y}_*$.

3.3.2 Choice of Kernels and Hyperparameters

We do not assume that the coefficients for each principal component will have the same relationships in color space and as a result we train a separate GP for the eigencoefficients of each principal component. The GPs use the color coordinates of the training spectra as inputs $\mathbf{x}$ and the PCA eigencoefficients as the outputs $y(\mathbf{x})$. In section 3.2 we showed how the eigencoefficients can be used to reconstruct a spectrum using the PCA eigenspectra. Since our goal is to create new SEDs at specific points in color space we use GP regression to estimate new eigencofficients for points in color space. Then we use the coefficients along with the eigenspectra to calculate a new SED. Since the covariances in color space may be different for each of the principal components we use a separate GP regression for each

eigencoefficient. We also tune each GP to find the best hyperparameters for the covariance function. These hyperparameters are the terms in each covariance function that affect the strength of the relationship of the points in the training set to a desired measurement location. For instance, the commonly used squared exponential covariance function uses a scale length to adjust the weighting in the GP based upon Euclidean distance to the training points. In order to find the hyperparameters for a set of data, we maximize the log marginal likelihood function as found in Chapter 5 of Rasmussen and Williams (2005):

$$\log\ p(\mathbf{y}|X,\theta) = -\frac{1}{2}\mathbf{y}^T K^{-1}\mathbf{y} - \frac{1}{2}\log\left|K\right| - \frac{n}{2}\log 2\pi \tag{3.5}$$

and use K to mean the same input data covariance matrix as above while n refers to the size of the training set.

We implement our GP methods using the Python language package, *george*[1] (Ambikasaran et al. 2015) and use for comparison 4 different kernel functions that come in *george* to describe our covariance. All are stationary kernels in which the kernel function does not depend on the values of the input coordinates, but only on the distance between points. These are the squared exponential, $\theta_1 \exp(\frac{-d_i^2}{2\theta_2})$, the exponential, $\theta_1 \exp(-\sqrt{\frac{d_i^2}{\theta_2}})$, the Matern-3/2, $\theta_1(1+\sqrt{\frac{3d_i^2}{\theta_2}})\exp(-\sqrt{\frac{3d_i^2}{\theta_2}})$ and the Matern-5/2, $\theta_1(1+\sqrt{\frac{5d_i^2}{\theta_2}}+\frac{5d_i^2}{3\theta_2})\exp(-\sqrt{\frac{5d_i^2}{\theta_2}})$. θ_1 and θ_2 are the tunable hyperparameters of each covariance model, where the first is used to set the signal variance in the outputs and the second adjusts the distance scale on each model. To set the values of θ_1 and θ_2 for each covariance function we maximize the value of Equation 3.5 using Nelder-Mead optimization implemented using the Scipy library (Virtanen et al. 2016).

3.4 Testing and Results

We are now going to describe how we utilize the PCA and GP methods introduced above in order to fill in the color space of template fitting photometric redshifts. The template fitting photo-z methods we want to improve often use sets of SED templates obtained by

[1] https://github.com/dfm/george

observation as mentioned in §3.1. In this work we attempt to mimic the small sizes of empirical template sets and therefore we create basis sets in our experiments starting from 10 training templates. We then approximate new templates at locations in color space poorly sampled by the original template set. We get new templates by estimating new weights for the basis templates with the Gaussian Process Regression (GPR) and use the weights to create a new template that matches the colors where we want to add templates.

Generating a template that matches the colors we want in the rest frame is, however, not enough. We need to make sure that the template we generate is realistic or we will not be able to generate accurate colors as the template is redshifted. To test this we use a large set of synthetic templates from Bruzual and Charlot (2003) and randomly sample 10 templates from the set to use in the generation of our basis set. Then we randomly pick an additional 50 templates from the larger set and calculate the rest frame colors of these templates. These are the locations in color space we will generate new templates with our PCA and GP method. To verify these are realistic we will then compare how well we regenerate the original template at that location in color space. Our metric for how well we perform is the residual between the flux at each wavelength for the original template and our estimated template.

To measure our performance we did three sets of 500 runs following this test outline. The first test looked at the results where we restricted SED estimation to the optical wavelength range covered by the filters we used for the input colors. The other two tests looked at estimating the SEDs over additional wavelengths above and below the optical range. For each test, we compared our results for SED estimation against two other methods commonly used to expand template sets, nearest neighbor estimation and linear interpolation. In each of the 500 runs within an individual test the results were stored and a new run was started with a new set of training and test SEDs randomly chosen from the SED library. We then averaged over all the results to get an idea of how well we perform with many different mappings of color space.

3.4.1 Creation of training set

As templates we used SEDs from the LSST simulations (Connolly et al. 2014) SED library[2] which are Bruzual and Charlot (2003) (BC03) model SEDs. The full library of 959 spectra samples 4 different star formation histories and span a range of ages from 1.585 Myr to 12.5 Gyr (using the Padova 1994 isochrones). It also includes 6 different metallicities and uses the Chabrier (2003) IMF as described in the BC03 package. Emission lines are not included in these spectra. The top part of Figure 3.1 shows a set of sample SEDs with different ages, metallicities and star formation histories. The spectra cover a wavelength range from 9 nm up to 160 μm, but we only used wavelengths less than 2400 nm where the resolution is best. Here the resolution varies between 0.1 nm for most of the optical range up to 10 nm at longer wavelengths and values in between for other wavelengths. The bottom part of Figure 3.1 shows a more detailed look at the resolution throughout the wavelength range. Since some spectra in the library are nearly identical in shape and the Gaussian processes require non-singular matrices for matrix inversion we trimmed the catalog to 789 spectra by removing spectra that duplicated the flux of another within 0.001% at more than 90% of the wavelength points.

In each run we created a new basis set from 10 spectra randomly picked from the full set of 789 spectra. We used 10 spectra in order to provide a training set comparable in size to photometric redshift training sets like the commonly used Coleman et al. (1980) (CWW) templates. Then we used the PCA method in the decomposition module of scikit-learn (Grisel et al. 2016) to find eigenspectra and eigencoefficients. In each PCA we kept 9 of the 10 components since we found that 9 components in each PCA decomposition explained over 99.9999% of the variance. We show in Figure 3.2 the mean spectrum and the first three eigenspectra for one of our training sets in 3.4.2.

---

[2]We used the version current as of January 2017 which can be downloaded at `https://lsst-web.ncsa.illinois.edu/sim-data/sed_library/seds_170124.tar.gz`.

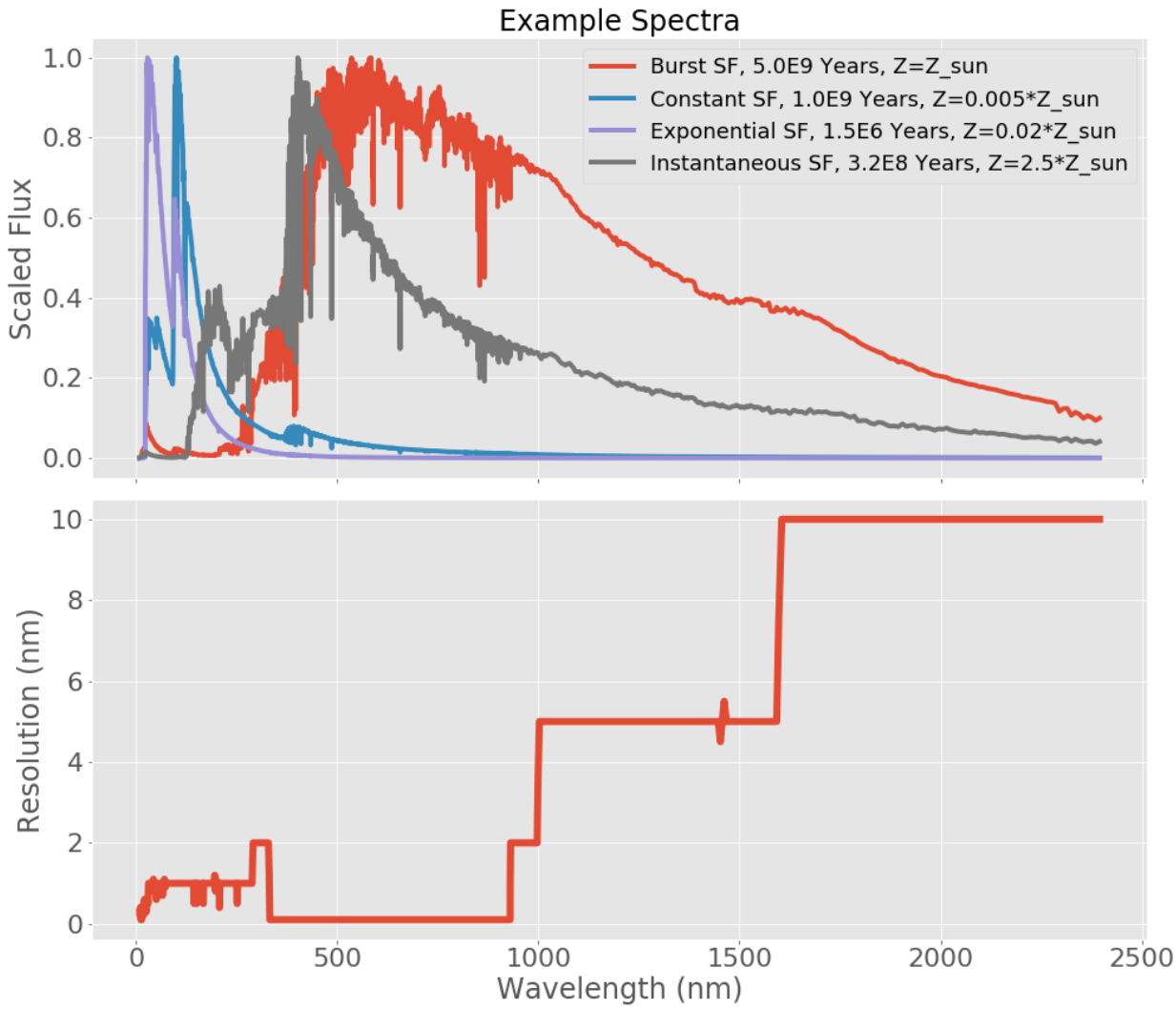

Figure 3.1: Top: Example spectra from LSST simulations SED library used in this work. Bottom: Resolution of the spectra as a function of wavelength.

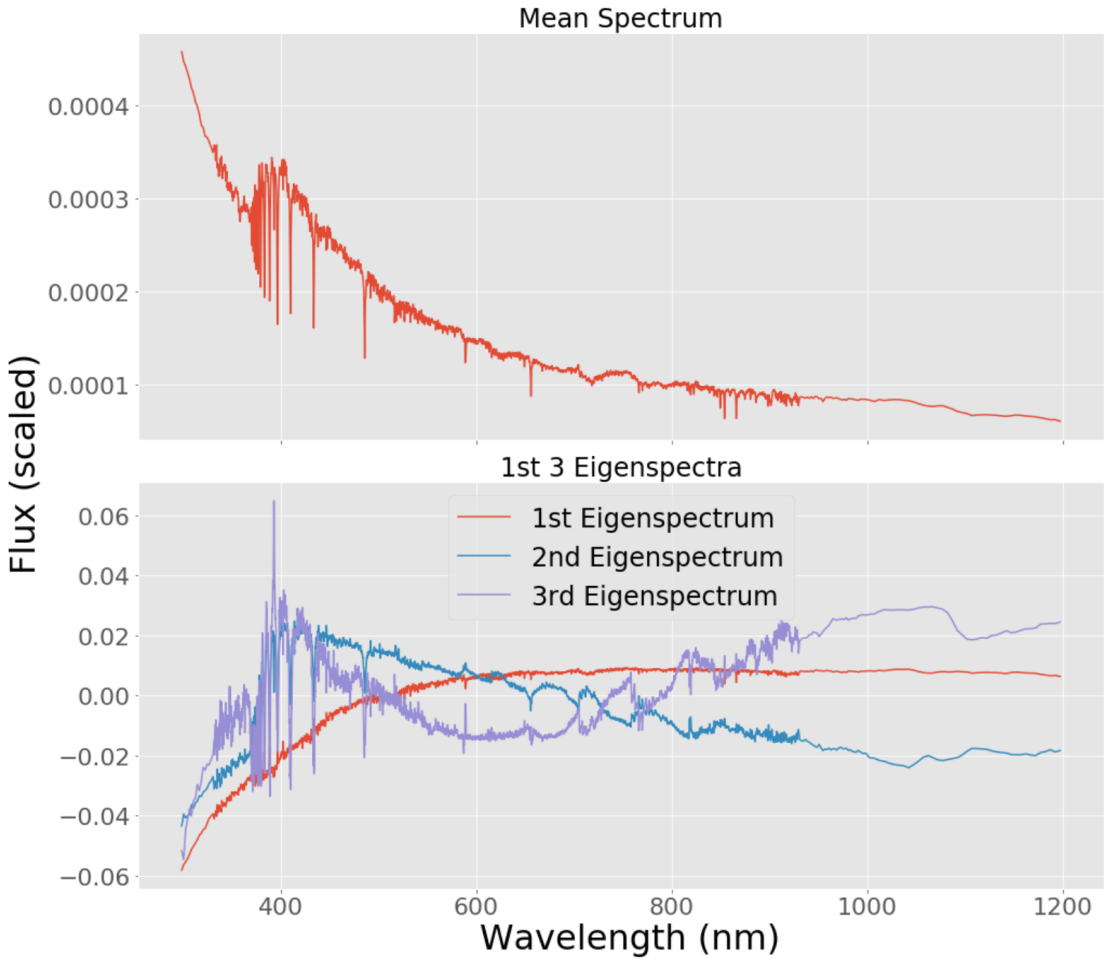

Figure 3.2: Mean spectrum and first three eigenspectra of the PCA performed on a randomly chosen set of 10 BC03 spectra.

3.4.2 Estimating SEDs

In order to estimate new eigencofficients at locations in color space we ran the 500 iterations of each test with a different one of the four kernel functions described in 3.3.2 for our Gaussian processes. In each of the 500 iterations we implement a separate Gaussian process for each set of eigencoefficients we ended up with 9 different Gaussian processes from each training set.

We then used the Gaussian processes and the colors of our test set to predict eigencoefficients for the test spectra and used this information along with the eigenspectra and mean spectrum of the training set to generate estimates of the test SEDs. Since we use Principal Component Analysis for our basis set there is a possibility of negative flux at some points in the estimated SEDs. In these cases we set the flux to 0 where it would otherwise be negative.

To generate our linear estimation comparison set we trained a linear regression in color space with the training SEDs and then used this to estimate the flux values at the colors of our test spectra. For the nearest neighbor comparisons we experimented with different variations where the nearest neighbors were determined by distance in color space between the test colors and the nearest training colors. We tested using a uniform weighting with 1, 2 or 4 neighbors as well as a distance weighted estimate using 2 or 4 neighbors.

This means in each section below we ran 500 iterations with 10 different settings: 4 PCA + GP estimation runs (each with a different covariance kernel function), 5 nearest neighbors runs (for a test point we choose an average spectrum of the 1, 2 or 4 nearest training spectra with and without weighting by the distance to the test point for the options with 2 and 4 spectra), and a set of linearly interpolated spectra.

Optical Wavelengths

The colors we plan on using for this technique are most likely those of an optical survey such as the Large Synoptic Survey Telescope (LSST). Using the latest version of the LSST

Table 3.1: Mean Gaussian Process hyperparameter values

Principal Component	Exponential (θ_1, θ_2)	Squared Exponential (θ_1, θ_2)	Matern-3/2 (θ_1, θ_2)	Matern-5/2 (θ_1, θ_2)
1st	(5.14E-5, 6.16E+1)	(1.65E-4, 9.66)	(9.21E-4, 2.57E+2)	(2.75E-4, 3.60E+1)
2nd	(1.58E-6, 6.09E-1)	(3.66E-5, 2.14)	(6.58E-5, 1.60E+1)	(8.09E-5, 8.94)
3rd	(9.94E-8, 1.23E-1)	(9.93E-6, 1.19)	(3.49E-5, 3.70E+1)	(3.39E-5, 5.71)

Note: Mean Gaussian Process hyperparameter values for 1st three principal components after 500 runs. θ_1 is a scaling factor and θ_2 is a length factor.

Table 3.2: Percentage residual errors in flux in 3.4.2

Error Metric	Wavelengths nm	Exp.	Sq. Exp.	Mat.-3/2	Mat.-5/2	NN	2 NN	Linear
Mean Error	299 - 1200	7.25%	3.32%	3.60%	3.17%	9.70%	9.23%	9.62%
IQ Mean Error	299 - 1200	4.04%	2.09%	2.63%	2.11%	7.39%	6.18%	6.65%

Note: Residuals are errors in flux between true and estimated SEDs.

bandpasses from the LSST simulations software stack (Connolly et al. 2014) [3] we calculated the colors for our training and test SEDs. We followed the procedure outlined above only using the SEDs at wavelengths from 299 - 1200 nm which covers the range of the LSST filters for the PCA stage. The mean maximum likelihood hyperparameters across all 500 runs for the first three eigencoefficients are shown for each of the kernel functions in Table 3.1 where, as in Section 3.3.2, θ_1 is a scaling factor and θ_2 is a length factor. Notice that the θ_1 and θ_2 hyperparameters in each model vary with each PCA component as expected.

The output of each of the 500 runs is a set of 50 estimated SEDs from each estimation method at the locations in color space of 50 original test SEDs. To compare the results we find the absolute difference between the estimated and original SED for each method and then calculate the fractional residuals. Figure 3.3 displays the mean residuals between

[3]We used the versions current as of March 2017 found here: https://github.com/lsst/throughputs

the predicted SEDs and the actual SEDs as a function of wavelength for the two Gaussian Process kernels with the lowest mean residual error across the complete wavelength range, the two nearest neighbor settings with the lowest mean residual error and the linear interpolation. The results show that our GP method with a Matern-5/2 kernel outperforms all other techniques across almost all wavelengths that we used. The same technique with a squared exponential kernel also works well compared to the alternatives. In Figure 3.4 we show the ratios between our method and the comparison methods. The best nearest neighbor technique in this test was the distance weighted 2 nearest neighbors estimate and this is what we include for comparison in Figure 3.4. For most of the spectra our method using the Matern-5/2 kernel has less than 60% of the error as the nearest neighbor method and less than 50% of the error as the linear estimation method. The mean error and median error across the spectrum is shown in Table 3.2. Comparing the different methods we see that the mean percent error was 3.20% and 3.49% for the Matern-5/2 kernel and squared exponential kernel methods, respectively. This is lower than the 9.70% mean error for the nearest neighbor method, the 9.23% for the 2 nearest neighbors method and the 9.62% mean error when using linear estimation. In fact, all four GP kernels outperformed the nearest neighbor and linear methods.

Expanding the wavelength range

It is important for applications involving redshifts or predicting magnitudes in other bands to be able to use our technique to generate accurate SEDs over a larger wavelength range than just the rest frame optical wavelengths. Therefore our second test was using the same training and test SEDs with the same colors and performing the same analysis, but extending the wavelengths to 99 and 2400 nm in our training SEDs for use in the creation of the PCA basis.

Table 3.3 shows that our methods are not performing as well in this test as they did in the first one. The mean error across the full spectrum is now as high as 73.5% for the squared exponential kernel and even the best GP kernel, the exponential is at 43.7%. The traditional

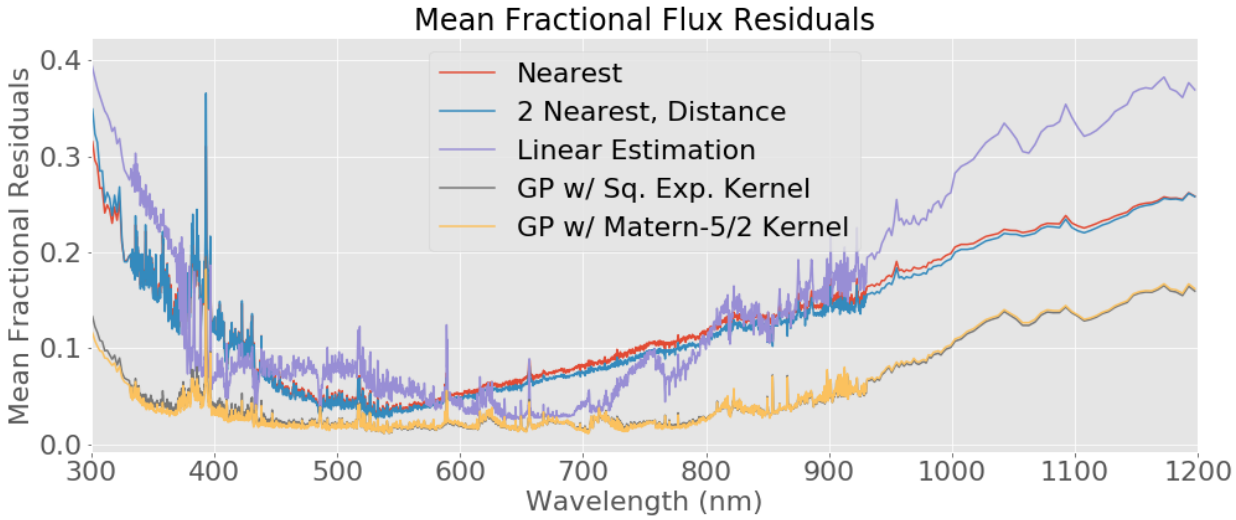

Figure 3.3: Mean fractional residuals between predicted and original spectra over 500 runs. Gaussian processes with a Matern-5/2 kernel outperformed across almost all wavelengths.

Table 3.3: Percentage Residual Errors in flux for 3.4.2

Error Metric	Wavelengths nm	Exp.	Sq. Exp.	Mat.-3/2	Mat.-5/2	NN	2 NN	Linear
Mean Error	99 - 2400	43.7%	73.5%	51.2%	60.1%	27.1%	33.6%	56.5%
Mean Error	299 - 1200	14.7%	19.8%	16.1%	17.6%	14.9%	14.2%	16.1%
IQ Mean Error	99 - 2400	13.4%	15.9%	15.4%	14.6%	13.7%	13.2%	15.9%

Note: Residuals are errors in flux between true and estimated SEDs.

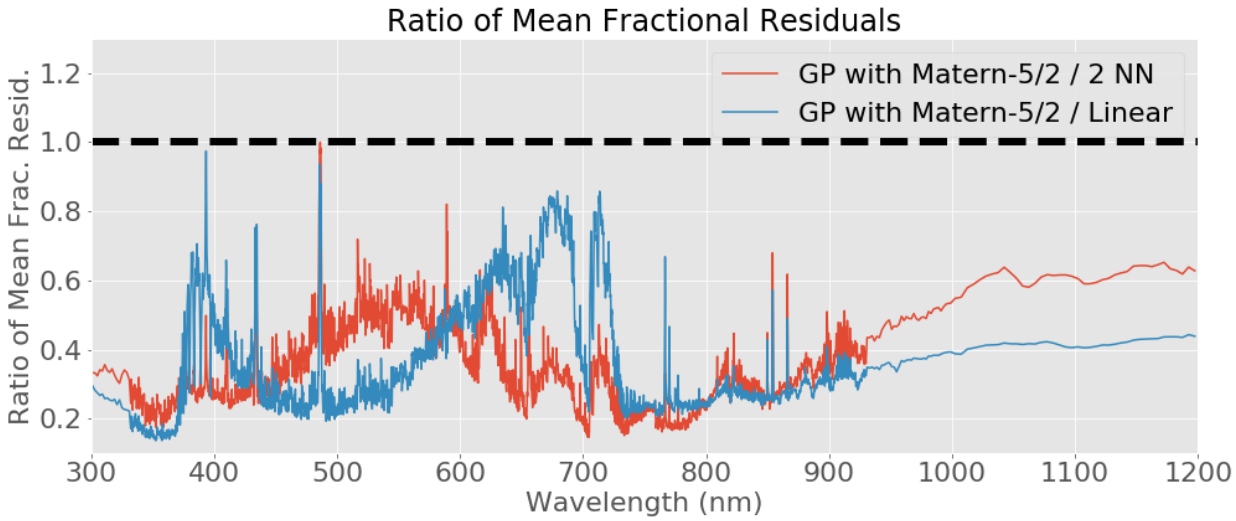

Figure 3.4: Ratio of the fractional residuals from our method compared to that from 2 nearest neighbors with distance weighting and to that from linear interpolation. The line drawn at 1.0 shows the level at which the errors in each method would be the same. Values below this line mean that the Gaussian Process estimated SEDs have a lower mean residual than those of the alternate method. The Matern-5/2 kernel has less error than any other method at almost every wavelength.

methods also show increases in error but now outperform the Gaussian process regression methods. Here the exponential kernel performs better than the other kernels but still is worse than the nearest neighbor and linear methods across most of the spectrum. Looking at the second row of Table 3.3 however, we see that the errors in the region of the spectrum used to train the GPs are actually comparable between the Nearest Neighbor method and our GP method. Since all SED sets in this test are the same as those we used in 3.4.2 it appears that the addition of wavelengths outside the range of our filters are leading to the decrease in performance of our method with most of the error appearing at the blue end of the spectrum. We therefore decided to try and find ways to use information about the rest of the spectrum available to our training set in order to improve our estimates for the test set where we only use the LSST filters.

Using artificial filters in training

Since it seems that the errors are driven by features outside the range of our filters we decided to expand the amount of information we were using in our training set. We added a series of top hat filters outside the range of the LSST *ugrizy* filters we were using. We tried a few simple combinations of 50 nm wide top hat filters close to the existing bands and the best results came from 2 50 nm wide top hat filters on the blue end at 100-150 nm and 200-250 nm as well as 2 on the red end at 1250-1300 nm and 1350-1400 nm.

First, we wanted to test the hypothesis that the relationship of the spectra in the UV and IR regions were not being fully captured by only using the optical filters. We took our full set of BC03 spectra and picked out a test spectrum. We then took all of the other spectra within a radius of 0.1 magnitudes in the 5-dimensional color space provided by the LSST filters. The top plot in Figure 3.5 shows the difference in flux between these spectra and the test spectrum. Then we repeated this process with the same spectrum, but in the 9-dimensional color space of the LSST *ugrizy* plus our top hat filters in the UV and IR regions of the spectrum. Once again we used a radius of 0.1 magnitudes and show the results in the bottom plot of Figure 3.5. Notice that in the comparison of the spectra with similar optical

Table 3.4: Percentage Residual Errors in flux for 3.4.2

Error Metric	Wavelengths nm	Exp.	Sq. Exp.	Mat.-3/2	Mat.-5/2
Mean Error	99 - 2400	32.1%	72.3%	25.6%	44.7%
Mean Error	299 - 1200	12.1%	21.6%	13.1%	16.9%
IQ Mean Error	99 - 2400	11.1%	16.8%	15.1%	14.2%

Note: Residuals are errors in flux between true and estimated SEDs.

colors the SEDs are very similar across the optical and IR wavelengths. In the UV portion of the SED (wavelengths less than the u band filter) there is, however, significant diversity in the spectra. When adding the top hat filters to expand the color space and selecting spectra with common optical and UV colors the large amount of diversity in the UV part of the spectrum is removed (for spectra within the same 0.1 magnitude radius of our test spectrum).

Confident that the top hat filters would help to better fit the relationships between spectra in color space we moved on to using the top hat filters in our GPR. Since we would not have this information when applying our technique to observed colors we only used this to optimize the hyperparameters for the GP on the training set information. We then used these optimized hyperparameters with the same 5 color GPs as before on the test set data. Therefore, we simulated having full spectra in our templates but only the observed color information we would have in a real application as the data for our test set estimates. With this plan we reran our tests of 500 iterations once again and compared the results.

In general, there is an improvement in our estimates of the full spectrum compared to test 2. Figure 3.6 shows our results for the mean fractional residuals using the Matern-3/2 kernel with and without the artificial filters. There is obvious improvement across the whole

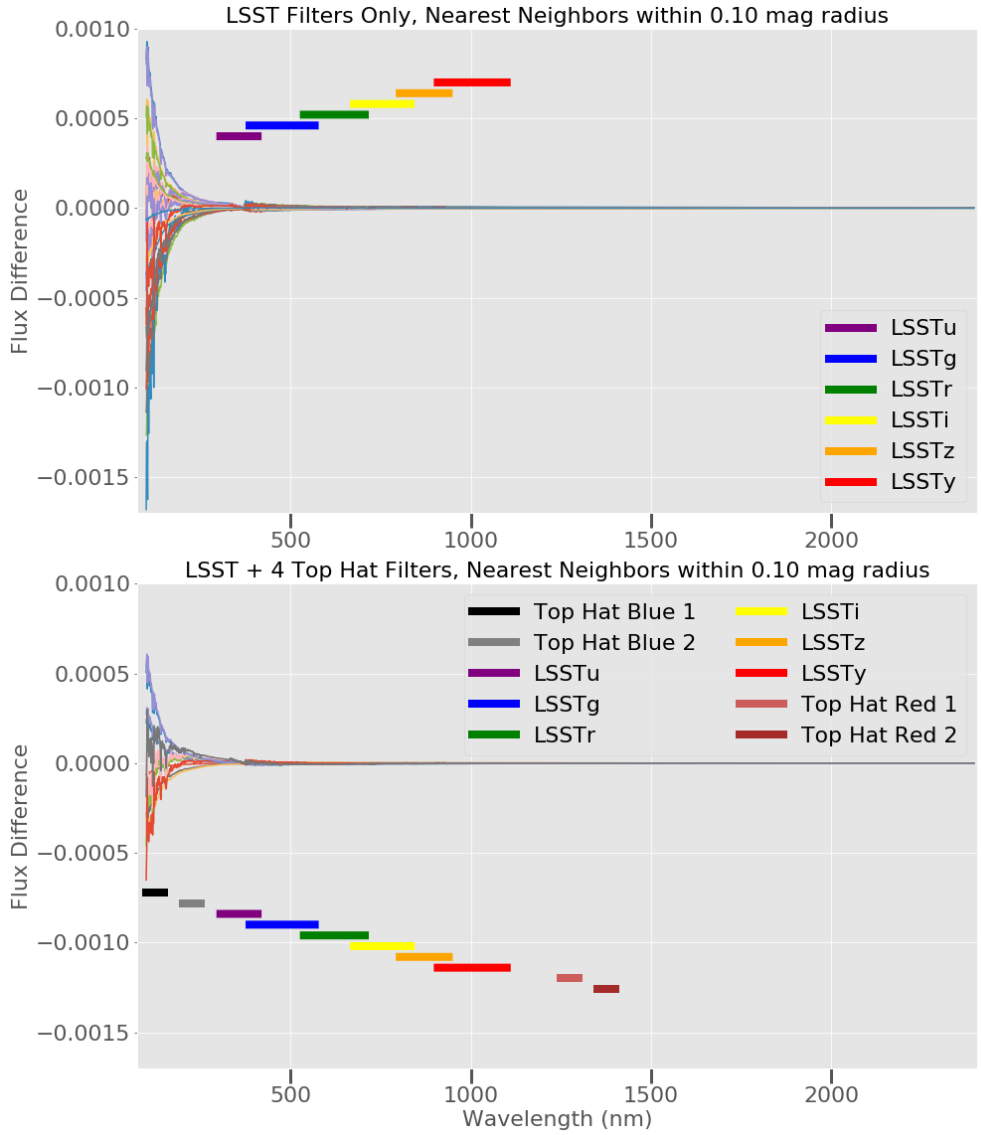

Figure 3.5: Top: Difference between a single BC03 spectrum and all other spectra within a radius of 0.1 mags in the 5-dimensional LSST color space. The wavelength span of the LSST filters are shown as the colored horizontal bars. Bottom: Difference between a single BC03 spectrum and all other spectra within a radius of 0.1 mags in the 9-dimensional LSST+4 top hat filters color space. The wavelength span of the LSST filters and the added top hat filters are shown as the colored horizontal bars.

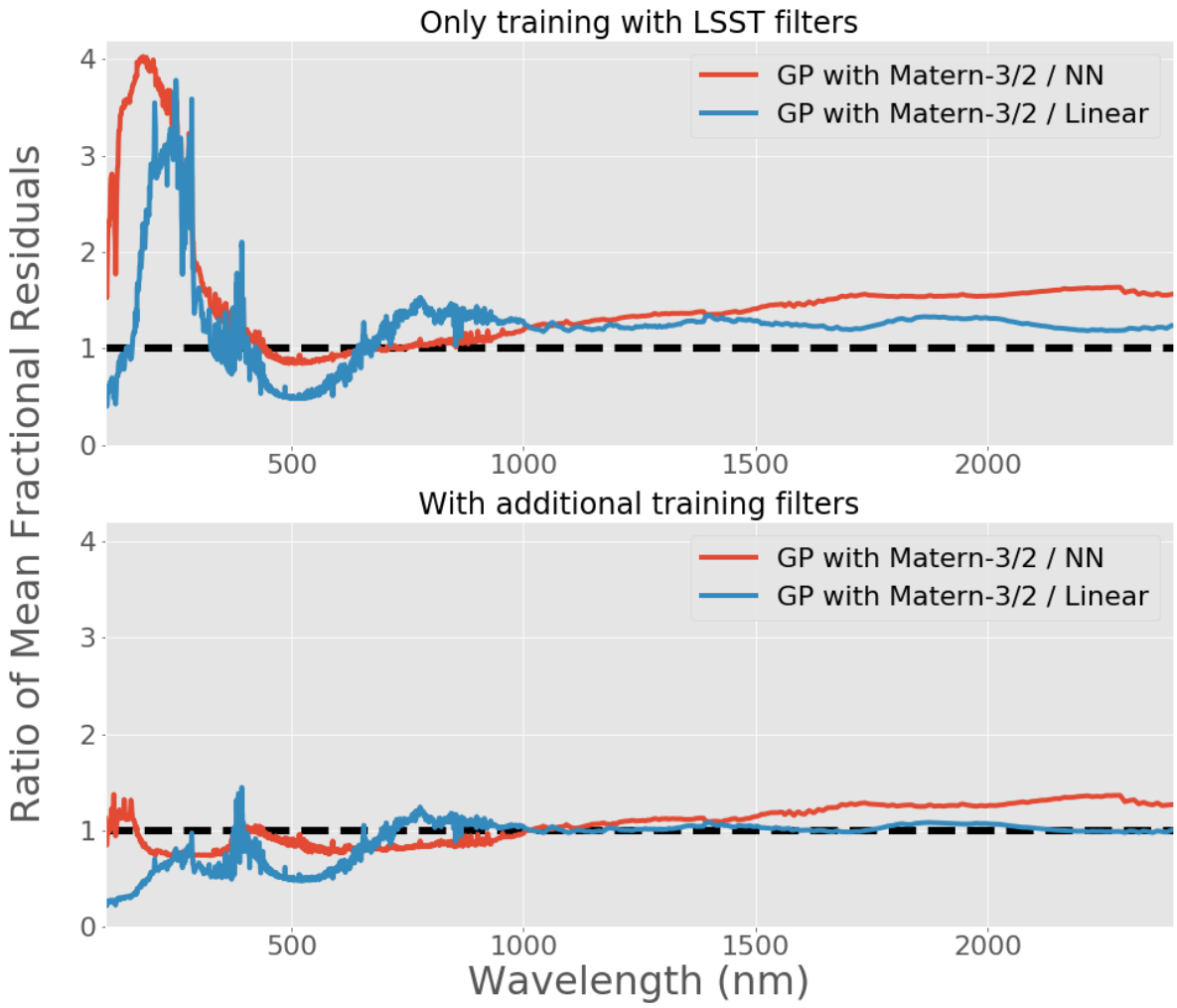

Figure 3.6: Top: Ratio of the fractional error from our method compared to that from nearest neighbor and linear interpolation training with only 5 LSST colors. Bottom: Ratio of the same methods using 9 colors to train the hyperparameters of the Gaussian Process.

spectral range, but most especially in the shorter wavelengths that now have training filters covering that wavelength range. Table 3.4 shows the new values for the results using the GPR with new top hat filters. The mean residual error using the GP method decreased with every kernel. For the Matern-3/2 kernel it dropped from 51.2% to 25.6% now beating the nearest neighbor and linear techniques. In the range between 300 and 1200 nm there is improvement for 3 of the 4 kernels as well. Finally, we created an interquartile (IQ) mean by only using the middle 50% of estimated SEDs for each method based upon the mean error of the overall spectrum. We wanted to look at the IQ mean in order to see if outliers were having a large effect on our estimated SEDs in this test. The bottom row of Table 3.4 shows training with the additional filters leads our GP method with the exponential kernel to perform the best by a considerable amount according to this measurement. This suggests that the exponential kernel does a good job reproducing the spectra, but also produces larger errors in some spectra compared to the Matern-3/2 kernel. Understanding where these errors are present could be possible since Gaussian Processes produce estimates of the variance in its results and this is addressed in our discussion of future work in Chapter 5. Knowing when to accept and when to reject our estimates could allow us to produce an even better set of estimated SEDs by combining estimates from multiple kernels.

To understand what is causing the improvement in results with the additional filters we compared the hyperparameters for all four kernels to the hyperparameter values from training without additional filters. We saw that the distance hyperparameter was consistenly larger when we trained with the additional filters. This indicates that the distance that a training point affects the regression is larger when we use the additional filters. This could be because the features in the spectra that lie in wavelengths outside the range of the LSST filters vary much more slowly as we move between different types of spectra with different LSST colors. Figure 3.6 shows that the nearest neighbors still works best when looking beyond 1200 nm and this result would be consistent with the nearest spectrum in LSST colors being a good approximation a large distance away from the original training point. In fact, we will look at this in more detail in the next section.

Overall, the results of this test indicate that adding artificial filters helps our method with all kernels but we still need to do further work to reduce outliers and produce more accurate SEDs consistently. So far, we have just tested using simple top hat filters and additional development of these artificial filters can attempt to maximize the information used to identify the relationships in color space between spectra. For now though it is encouraging that the simple additional filters lead the Matern-3/2 kernel to beat the nearest neighbor and linear methods when estimating a more complete spectrum and for the exponential kernel to beat them decisively when looking at the IQ mean results. Further work will hopefully help to reduce the outliers and lead to further improvements in the GP method over the other methods across the full wavelength range.

3.4.3 Extrapolation in Color Space

One of the challenges of using template sets is that available templates do not always cover the color space of interest. Improving estimates of SEDs in regions of color space that require extrapolation is one area that our method is able to improve compared to the other methods. In order to show this we sorted all the results from the previous tests by Euclidean distance from the color coordinate of the test point to the nearest training neighbor. Figure 3.7 shows the results for all objects within a radius of 0.6 magnitudes in color space to a training point. This covers over 95% of our test points. The top plot in Figure 3.7 shows the results from our test restricted to the optical wavelengths of spectra. In this test the GP method outperforms all the way through these distances but achieves best results over the nearest neighbors at larger distances going to lower than one-third of the error in the nearest neighbor and linear interpolation methods consistently across the distances. In the bottom plot we show results from our third test using spectra covering wavelengths from 99-2400 nm and using the additional top hat filters for training. In that test we saw that overall the nearest neighbors method was able to beat our GP method, but looking at this plot we see that the GP method is able to catch up and beat the nearest neighbors method when the test colors get further away from the color space of the original template set. However, the

GP method does not outperform the nearest neighbors by the same amount even at large distances from the nearest training point. This lends more evidence to the idea that the features in wavelengths beyond 1200 nm are valid over a larger range of LSST color space and why the larger distance hyperparameters we get from training with alternate filters helps our performance.

In comparison to the nearest neighbors method, this is likely due to the ability to extrapolate our weights to values outside the range present in the original basis set. For instance, consider a sharp break in between two filters that leads to a red color for a galaxy in these two filters. If we observe a galaxy with a redder color than we have in our training set it is not possible to add more weight to the spectral feature that causes this red color beyond what is present in the training set. This means that averaging the nearest neighbors will not get a spectrum with sufficient strength in this feature, but by using the basis SEDs and being able to extrapolate the weight given to the basis SED beyond the individual spectra in the training set allows us to better match the proper spectrum for that color. Linearly interpolating spectra can give a spectrum with a higher weight to this feature but will also affect many other features that may not be relevant. Using basis spectra allows us to target the relevant basis SED and thus extrapolate to new areas of color space better than the alternate methods for SED estimation.

3.4.4 Using Narrowband Filters

In order to further show the capabilities of our technique we also applied it to a set of narrowband filters once again using the same SED sets as we did in 3.4.2. We chose a set of 4 narrowband filters from the Hubble Space Telescope ACS/HRC [4]. The filters are centered at approximately 344, 502, 658 and 892 nm and range in effective width from 56.7 to 149.1 nm. We ran the same tests as 3.4.2 and 3.4.2 and present the results in Tables 3.5 and 3.6.

The results are very similar to the wider LSST filters. In the first test we outperform the

[4]Filter transmission curves collected from: `http://svo2.cab.inta-csic.es/svo/theory/fps3/index.php?mode=browse&gname=HST&gname2=ACS_HRC`

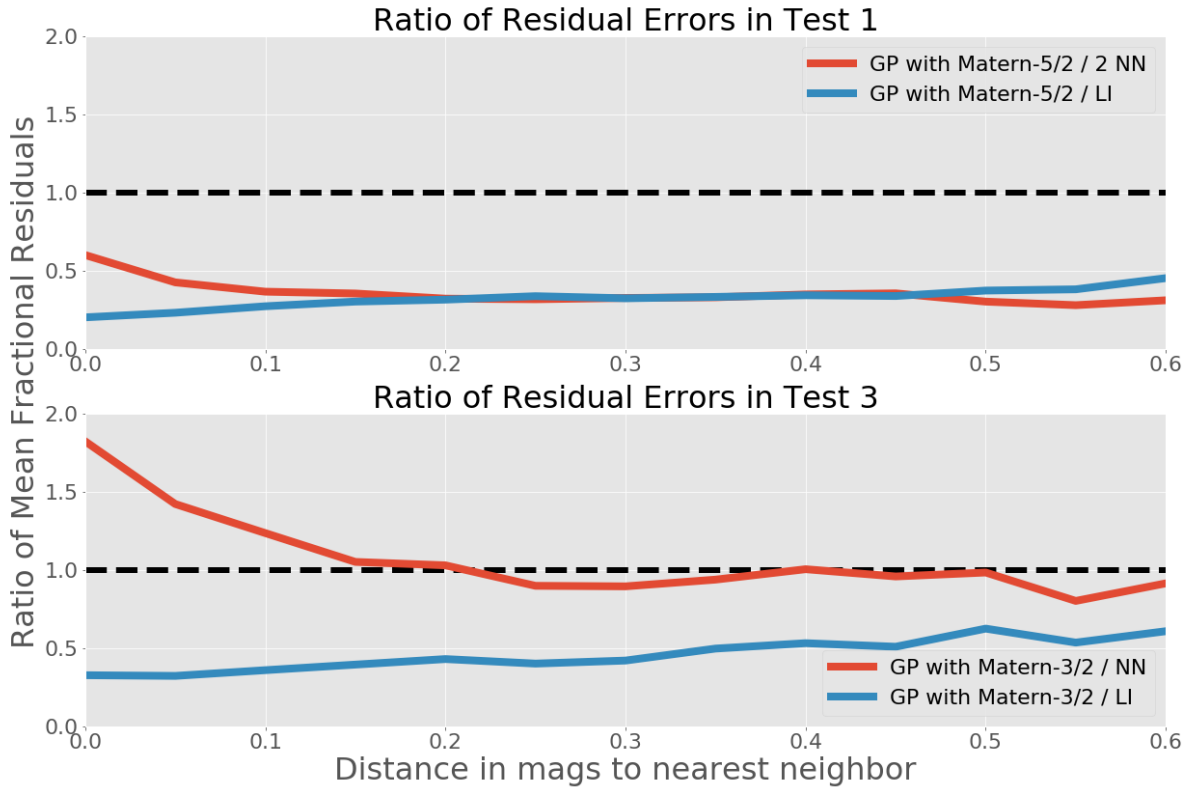

Figure 3.7: Here we compare the ratio of mean residual errors between SED estimation methods as a function of Euclidean distance to the nearest training set point. In both cases the Gaussian Process method improves results relative to the nearest neighbor as the distance to the nearest training point gets further. Top: Results from the test only using the optical wavelengths. Bottom: Results from the final test with additional filters and wavlengths 99-2400 nm.

.

Table 3.5: Percentage residual errors in flux with narrowband filters limiting SEDs to 299-1200 nm.

Error Metric	Wavelengths nm	Exp.	Sq. Exp.	Mat.-3/2	Mat.-5/2	NN	2 NN	Linear
Mean Error	299 - 1200	7.51%	4.74%	3.78%	3.98%	10.1%	9.57%	7.42%
IQ Mean Error	299 - 1200	4.38%	2.71%	2.80%	2.53%	7.74%	6.53%	5.17%

Note: Residuals are errors in flux between true and estimated SEDs.

Table 3.6: Percentage Residual Errors in flux for with narrowband filters extending SEDs to 99-2400 nm.

Error Metric	Wavelengths nm	Exp.	Sq. Exp.	Mat.-3/2	Mat.-5/2	NN	2 NN	Linear
Mean Error	99 - 2400	31.0%	137.7%	27.6%	58.4%	25.4%	30.6%	33.6%
Mean Error	299 - 1200	12.5%	33.6%	14.9%	21.7%	15.1%	14.3%	11.9%
IQ Mean Error	99 - 2400	11.5%	21.8%	16.6%	15.9%	13.8%	13.1%	12.1%

Note: Residuals are errors in flux between true and estimated SEDs.

nearest neighbor and linear methods in 3 of the 4 kernels with mean residual error as low as 3.78% for the Matern-3/2 kernel. We also tested on spectra extended to 99-2400 nm using the same artificial filters plus the LSST y filter as training filters to fit the hyperparameters like we did in 3.4.2. We added the LSST y filter to the training set to bridge the gap between the reddest narrowband filter and the red top hat filters. In this test we get comparable results to the nearest neighbors method when using the Matern-3/2 kernel. Once again, a refined set of training filters may be able to further improve our results, but for a simple set the results are encouraging that even with the limited amount of the spectrum sampled by broadband filters and with only 4 filters we are able to estimate a large portion of the spectrum to almost the same accuracy.

3.5 Expanding Template Sets for Photometric Redshifts

While the overall goal of our estimation method is to expand template sets for a variety of applications, one of the most common uses for template sets is in estimating photometric redshifts. As a demonstration of the benefits of our technique we performed photometric redshift estimation on a mock catalog with BC03 training spectra. In this section, we first establish the benefits of larger template sets on photometric redshift estimation. We then show how our technique can be used to improve photometric redshift estimation by expanding a template set through the generation of new estimated SEDs at specific locations in color space. Additionally, it is important to note that while Gaussian Processes have been applied to photometric redshifts previously (Way et al. 2009; Almosallam et al. 2016; Leistedt and Hogg 2017) our method is a general framework for expanding template sets and constructing SEDs with PCA coefficients. Below we use an existing photometric redshift method with an expanded template set as an example application since it is easy to compare results by plugging in different template sets with the same catalogs and code. Other possible applications of our method include representing large sets of SEDs with only a few PCA coefficients or creating continuous distributions of SEDs using features other than colors. We discuss these further in Chapter 5.

3.5.1 Mock Catalog and Method

Our simulated catalog was 100,000 objects from a larger catalog used in (Graham et al. 2018) created using the galaxy formation model of Gonzalez-Perez et al. (2014) with redshifts up to z=6, but we focus our results on the 89,471 objects with $z \leq 3$. The photometry in the catalog is in the SDSS ugriz and Pan-Starrs y filters. The catalog came without errors so we generated photometric errors of a simulated LSST-like survey using the LSST simulations software stack (Connolly et al. 2014).

To calculate redshifts we used LePHARE (PHotometric Analysis for Redshift Estima-

tions)[5] (Arnouts et al. 1999; Ilbert et al. 2006). The code is an SED template fitting code that uses a chi-square method comparing the photometric flux of the templates at a sequence of redshifts to the catalog photometry to find the best template and redshift match for each galaxy.

3.5.2 Expanding template sets

We wanted to consider the impact of the number of templates used in a photometric redshift analysis. Along with the original set of 10 template SEDs, we created a set of 60 BC03 templates by randomly selecting 50 additional spectra from the full SED library and a set of the 10 original templates plus 50 templates created using an exponential kernel function trained with the method explained in Section 3.4.2. We used the SDSS ugriz and Pan-Starrs y filters along with the same 4 top-hat filters from before. To decide where in color space to estimate new colors we performed a k-means clustering on the catalog in the SDSS ugriz + Pan-Starrs y color space with k=50. We then used the locations of the color centers as the locations to estimate new SEDs. While this method of estimating color locations is done on the redshifted colors and thus will cover some areas with unrealistic rest frame colors it will make sure the rest frame SEDs of the majority of galaxies are covered. We then estimated redshifts using LePHARE for all objects in the mock catalog using the 3 template sets. Since we were trying to isolate the effect of changing template sets on the redshift estimation we included no priors when running the code. Therefore, the results presented here are a worst case scenario run without optimizing the redshift estimation in other ways. Table 3.7 shows the results of four statistics calculated from the photo-z estimation. We define the distance corrected difference in redshift to be $\Delta z = \frac{z_{true} - z_{phot}}{1+z_{true}}$ and calculate four statistics as used by the LSST:

- Bias: Outlier rejected bias. Mean of differences between true and estimated redshift across interquartile range (IQR). $\overline{\Delta z_{25-75\%}}$

[5] http://www.cfht.hawaii.edu/∼arnouts/LEPHARE/lephare.html

Table 3.7: Statistics for photometric redshifts for catalog objects with z $\leq$ 3

Template Set	Bias	Std. Dev.	IQR St. Dev.	Outlier Frac.
10 BC03 Templates	0.051	0.279	0.121	0.196
60 BC03 Templates	0.013	0.173	0.036	0.154
10 BC03 + 50 Exp Kernel	0.007	0.217	0.048	0.131

- Standard Deviation: Standard deviation of Δz. $\sqrt{\overline{(\Delta z - \overline{\Delta z})^2}}$
- Standard Deviation of IQR: The spread of the difference between true and estimated redshift for the middle 50% of differences divided by 1.349 to compare to standard deviation. $\frac{\Delta z_{75\%} - \Delta z_{25\%}}{1.349}$
- Fraction of Outliers: The fraction of catalog objects with Δz greater than 0.06 or 3 times the Standard Deviation of the IQR whichever is greater.

Table 3.7 and Figure 3.8 show that using a larger template set improves the redshift estimation in the range z $\leq$ 3. The standard deviation for the redshift residuals is reduced by 38% when going from 10 to 60 and the bias falls by 74%. Therefore, the coverage of templates in color space does make a difference in the accuracy of photometric redshift estimation and increasing the number of templates is beneficial. While this is an idealized case since we used the templates that were used to create the simulated catalog colors, we do expect to see that using our method to create additional realistic template SEDs will provide measurable improvement in photometric redshift estimation.

Comparing the exponential kernel method to the original 10 templates reveals significant improvements when using our technique to expand the template set. Our estimated SEDs successfully improve estimates across all measured statistics in the range z $\leq$ 3 including improving the standard deviation by 22.0%, the standard deviation of the IQR by 60.6%

and the bias by 86.2%. Figure 3.9 compares the scatter plots for the two runs side-by-side. The exponential kernel templates help eliminate some of the longer horizontal features found in the scatter plot for the 10 templates on their own.

We can also consider our 50 estimated SEDs against the results from adding 50 of the original template SEDs. The 60 template set in Section 3.5.2 had a better standard deviation in its estimates 0.173 to our 0.217 and the standard deviation of the IQR at 0.036 to our 0.048, but we were able to better it in the bias by 0.006. Furthermore, as noted above the 60 templates were an idealized case and it is not surprising that we were not able to match the standard deviation. But, it is a very positive sign that our best 50% of results represented in the IQR standard deviation and outlier rejected bias approach the levels of the 60 BC03 templates and in the case of the bias are able to improve upon that set.

3.6 Conclusion

We have shown that Gaussian process interpolation of template set eigencoefficients is able to create a continuous interpolation and extrapolation from a training set of SEDs to the SEDs for other points in color space. This mapping provides SEDs that improve upon standard interpolation techniques currently used both in the estimation of the true spectrum and in the generation of colors from the predicted SEDs. For the wavelength range where photometric filters and spectra overlap we can improve the mean error estimate of the spectrum for a given location in color space by over 65%. Furthermore, Section 3.4.3 indicates that the best improvements come as the test points get further from the training data. As an example application we demonstrated that our method can help photometric redshift estimation. We improved the standard deviation of the error in photometric redshifts by over 24.8% and lowered the outlier rejected bias by over 87.5% compared to the original template set. In the future we hope to extend the applications to improving outputs for galaxy evolution modelling codes and other areas where SEDs are used to calculate galaxy properties or in generation of mock catalogs. Overall, this technique is a powerful addition to the astronomical toolbox anywhere interpolation or extrapolation of template SEDs would

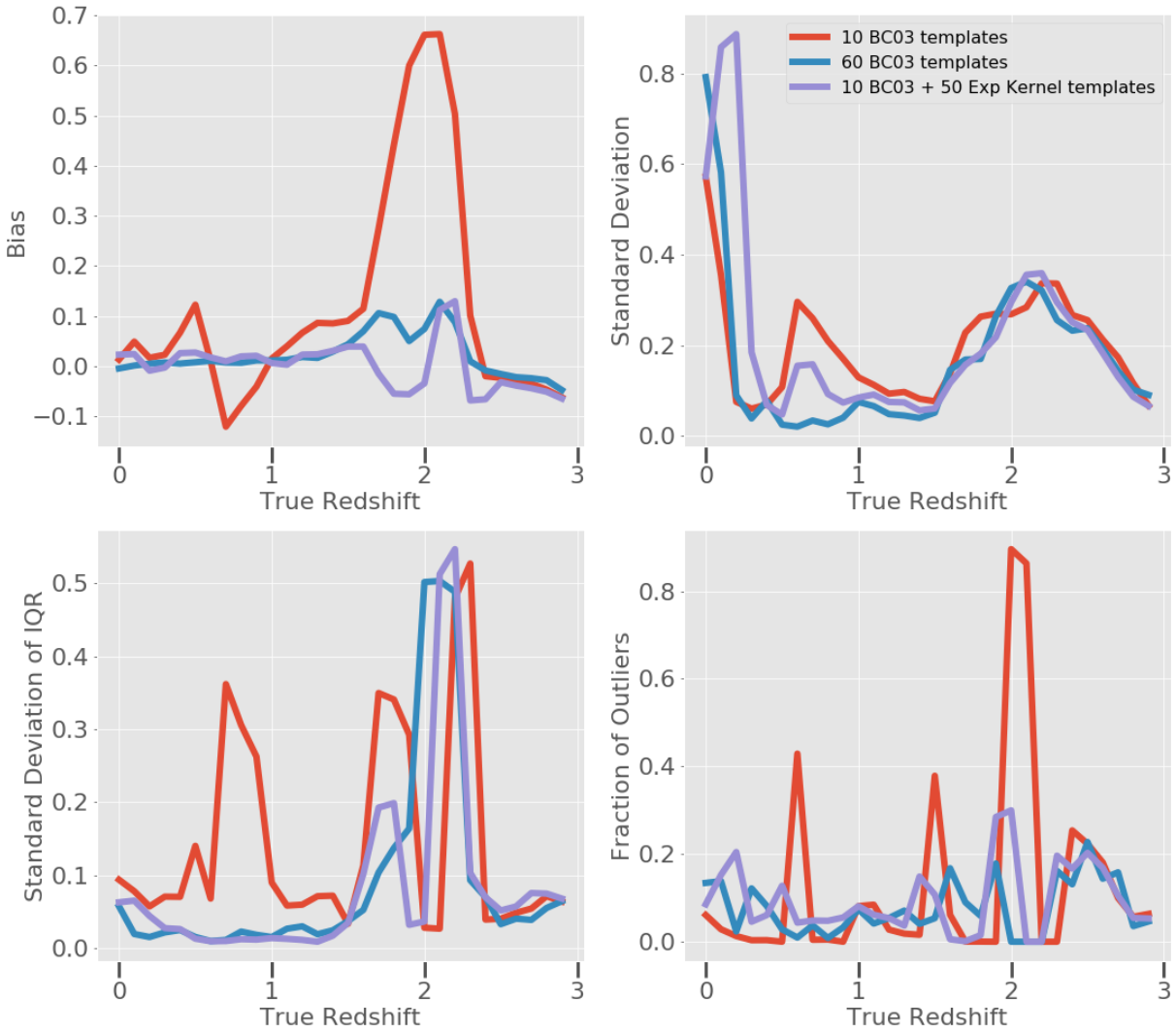

Figure 3.8: Comparing descriptive statistics from photometric redshift estimation with 10 and 60 BC03 templates to 10 BC03 templates + 50 estimated SEDs created using the Gaussian Process estimation method with an exponential kernel in color space. The addition of Gaussian process interpolated templates improves the redshift estimation in the range z $\leq$ 3 compared to the original 10 templates from which they are derived and produce results comparable to using 60 BC03 templates.

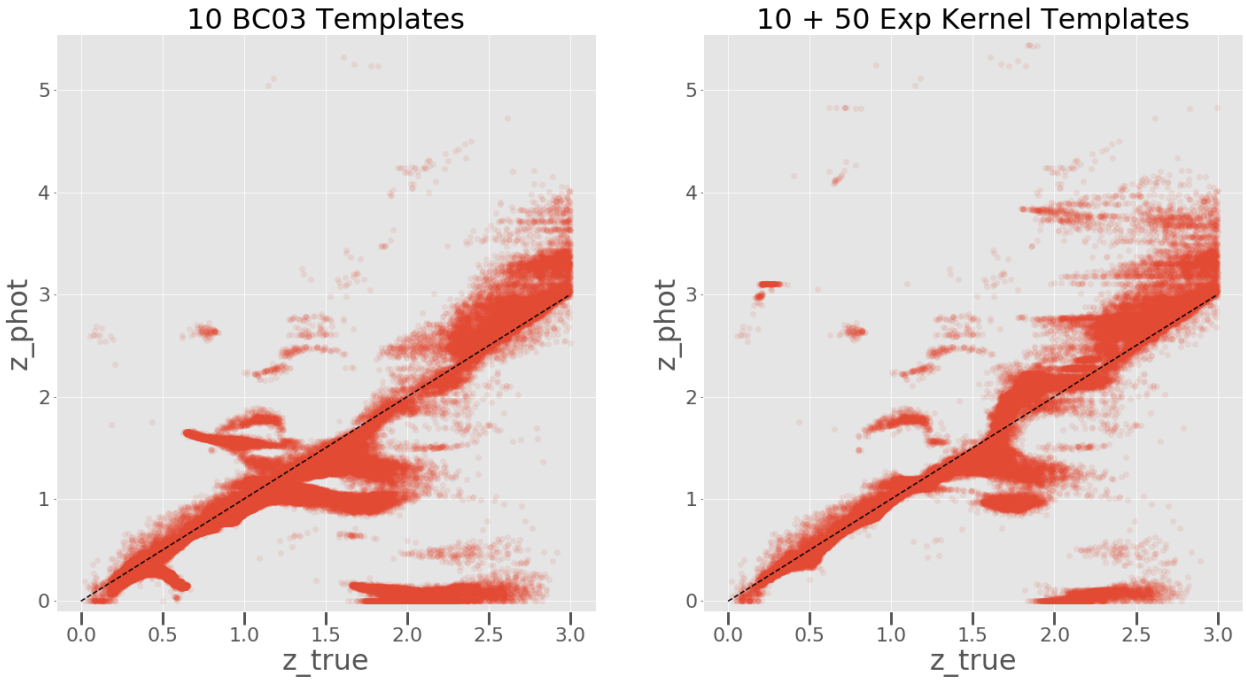

Figure 3.9: Scatter plots comparing the true redshift from the mock catalog and the estimated redshift from photometry when using 10 BC03 templates (left) and adding 50 estimated SEDs using our technique with an exponential kernel (right). Notice how the additional templates help eliminate some of the horizontal features that appear when only using the 10 templates on their own.

be useful and our Python code, ESP (**E**stimating **S**pectra from **P**hotometry), is documented and openly available on github at `https://github.com/jbkalmbach/esp`. It also includes a jupyter notebook with the code to reproduce all plots in our paper.

Chapter 4

COLOR SPACE DATA AUGMENTATION FOR PHOTOMETRIC REDSHIFTS

4.1 Introduction

We use photometric redshifts because of the cost and difficulty of obtaining spectroscopic redshifts for every galaxy. Empirical photo-z methods, however, can make use of a limited sample of spectroscopic redshifts from galaxies with observed colors to map out the broader relationship between colors and redshifts. But when these training sets are not entirely representative of the observed galaxy catalog in some property (such as colors or magnitudes) then the training set could bias the results and provide incorrect photo-z values (Collister and Lahav 2004).

Complete training sets for LSST photometric redshift estimation will require a minimum of a few tens of thousands of spectra (Newman et al. 2015). In order to fill in a significant gap in parameter space gathering hundreds or thousands of spectroscopic measurements may be required. Using a 10 meter class telescope like the Keck Observatory costs approximately \$53,700 per night[1]. Masters et al. (2019) recently used 23.5 nights of Keck time to get 3,171 high quality spectroscopic redshifts in a spectroscopic survey designed to map out poorly sampled regions of color space. Combining these amounts works out to approximately 1 week of observing time and $\sim \$400,000$ per 1000 spectroscopic redshifts. Therefore, reducing the need for 1000 additional spectroscopic measurments in a sparsely sampled area of color space can lead to significant savings in cost and observing time.

In this chapter we will explore techniques to artificially generate additional data to add to the training set. By adding this synthetic data we hope to improve photo-z performance

[1] http://ast.noao.edu/system/tsip/more-info/time-calc-keck

in areas of color space that have low to zero coverage and perhaps mitigate the need for additional training data in entire regions of color space.

4.2 Data Augmentation

Machine learning methods require large amounts of data to properly learn relationships in training sets that can be applied more generally. If a training set is not fully representative then the machine learning method will not perform well when it is applied to new data. For instance, Ribeiro et al. (2016) trained a network to identify wolves versus huskies where all the training images of wolves intentionally had snow in the background and the huskies did not. The classifier then identified a new image as a wolf every time there was snow or a light background in the image and husky otherwise "regardless of animal color, position, pose, etc" (Ribeiro et al. 2016). This shows the importance of using a wide variety of training examples that eliminate the possibility of overfitting to extraneous features. When a given training set is unrepresentative there are two options: 1) collect more training data or find additional sources of training data or 2) generate new, synthetic training data by manipulating the existing training data or from generative models. When it is difficult or costly to gather additional real training data we must use the second option. Methods to generate new, synthetic data are known as *data augmentation* methods.

In practice, a wide range of data augmentation methods are used to improve performance when training data is not sufficient either in overall volume or in overlap with the properties of the observed data set. For instance, Wong et al. (2016) demonstrated that for Convolutional Neural Networks (CNNs) increasing the training data available decreased the classification error of handwritten digits. They distorted the shapes of the original input images of handwritten digits and found that adding the new, distorted images decreased the error rate in the classifications of digits. Similar techniques of data augmentation are often used in astronomical applications of CNNs. The Deep-HITS CNN classifier of Cabrera-Vives et al. (2017) looked for point source transient objects but the real events are rare compared to the false candidates from background fluctuations or CCD artifacts among other possibil-

ities. Since these artifacts were readily available in all images it was easy to compile a large set of negative training data from the actual images but they needed to rely upon artificially generated point sources to create a comparable number of positive sources. They selected postage stamps of PSF-like sources already present in their data and reinserted them at different locations in the same image thus producing realistic transient point sources. They then trained their neural network image classification using the template, science, and 2 types of difference image postage stamp images to reject the artifacts and keep point sources that mimic the generated training point sources.

Data augmentation is not just useful for neural networks but for other types of machine learning classifiers as well. Revsbech et al. (2018) used data augmentation to help train a diffusion map and random forest based classifier for supernovae light curves. They used Gaussian Processes to generate additional light curve templates for training. Furthermore, Hoyle et al. (2015) applied data augmentation to improve photometric redshift estimation of faint SDSS data using a decision tree-based regression method. They used SDSS data with spectroscopic redshifts but with an apparent r-band magnitude cut to separate into a bright training set and a fainter test set similar to what is shown in Figure 4.1. This is a common problem in photometric redshift estimation because of the difficulty associated with getting faint spectroscopic redshifts. Spectroscopic datasets are often applied to deeper photometric catalogs and there may be galaxy types present in the faint sample that do not appear in the bright, spectroscopic sample. The authors augmented the bright training data with data from two sources: the first was from a simulated catalog of galaxies generated based upon the Millennium simulation (Springel et al. 2005) and the second was a collection of samples from the bright training data recalculated at different redshifts using the *K-Correct* code (Blanton et al. 2003). The results showed that including augmented data in the training sets allowed the authors to mitigate 41 percent of the degradation in the width of the photo-z error distribution created by an unrepresentative training set.

In this work we go beyond a simple apparent magnitude cut and into cutting out full regions of color space. This is motivated by the challenge of designing spectroscopic training

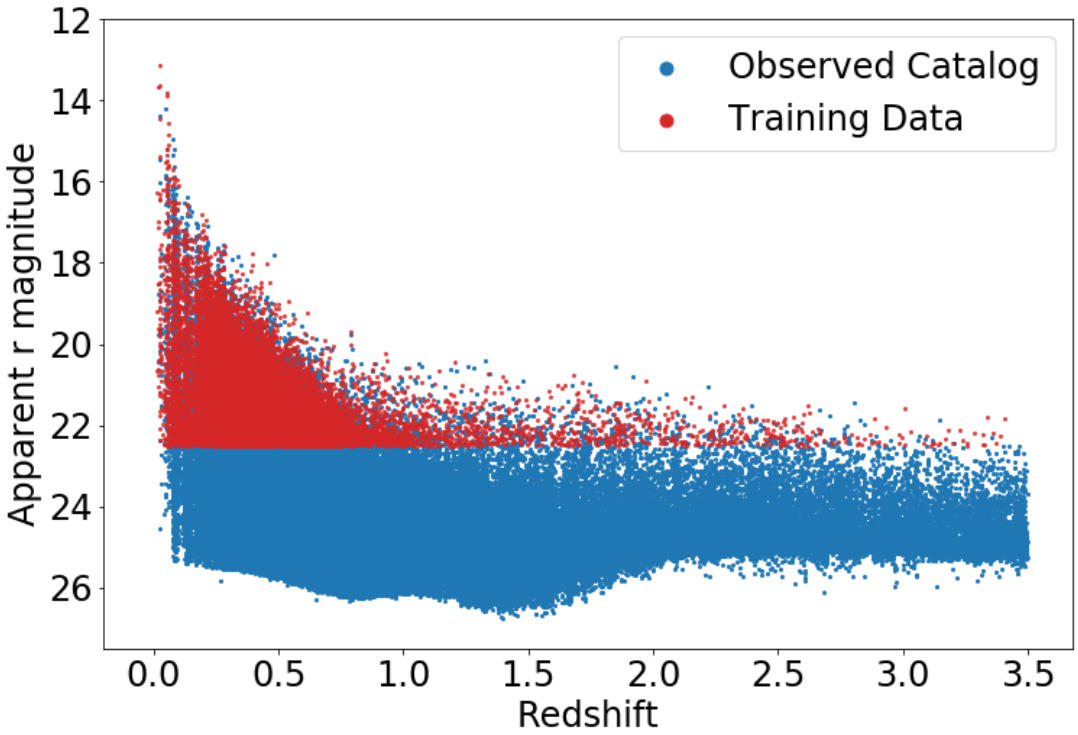

Figure 4.1: Example of an observational catalog that goes fainter than the available training data. This leads to an absence of training data at higher redshifts.

sets for photometric redshifts. The cost and difficulty of obtaining spectroscopic redshifts is a driver of photometric redshifts, but complete spectroscopic training sets are necessary for empirical photo-z to work. Masters et al. (2015) used Self-Organizing Maps (SOMs) to map out the high-dimensional color space into two dimensions and target the areas of color space with the highest need for additional data. Our approach is complementary and seeks to improve photo-z performance in areas that have low to zero coverage in the color space and perhaps mitigate the need for additional training data in entire regions of color space. Next we describe the creation of a training set to explore the effects of removing a sample of galaxies from a region of color space.

4.3 Sparse Color Space Training Sets

In order to study the impact of data augmentation we first create a training set for our baseline photo-z estimator by removing training galaxies from an area of color space while keeping objects in the test set with similar colors. We then compare results of photo-z with and without the fully representative training set. What we aim to understand is how do the photo-z results change with no spectroscopic redshifts in a region of color space and then how does the addition of limited amounts of new spectroscopic information change the results.

4.3.1 Sample Photo-Z Catalogs

We first take a simulated photometric catalog and randomly sort the catalog into a full training set of 200,000 galaxies and a test set of 50,000 galaxies. Our sample photometric catalog comes from a larger catalog created for Graham et al. (2018) and is derived from the Millennium simulation (Springel et al. 2005). The catalog contains simulated data in the six LSST bands (*ugrizy*) and includes redshifts up to $z = 3.5$ but we restrict ourselves to the range $z \leq 2.0$.

To create an unrepresentative training sample we cluster the full training catalog into 4 groups. We first scale the colors to have zero mean and a standard deviation of 1 in each color and then using K-Means clustering from the scikit-learn programming package (Grisel

et al. 2016). The clusters formed in the the training set color space are shown in Figure 4.2. In addition to the full training set we create 2 different types of training sets for our experiments by removing one of the pictured groups from the original training catalog and subsequently adding back 0, 10, 100 or 1000 galaxies for reduced coverage in the color space of interest. For our experiments below we removed "Group 0" from the clustering shown in Figure 4.2. This leaves us with an unrepresentative training catalog containing 153,382 training points to compare against the full training catalog. We chose to remove a cluster of points from color space rather than a cut on the edge of the color distribution because eventually we want to understand how many spectroscopic samples do we need to infer the relationship between color and redshift in densely populated regions of color space to the accuracy needed for dark energy measurements.

4.3.2 Photo-Z Method

To measure our results we use the Color Matched Nearest Neighbors algorithm (CMNN) of Graham et al. (2018) which we used previously in §2.8.2. We use the settings described in the original Graham et al. (2018) paper taking the nearest neighbor measured using the Mahalanobis distance in color space and do not return results when the distance is greater than the value given by the 0.68 level of the Percent Point Function (PPF) of the Chi-Square measurement. We compare the results with 4 photo-z metrics as well as comparing the number of test points that did not return a result since there were no training points within the 0.68 level of the PPF. The four metrics we use are calculated using the distance corrected difference in redshift, $\Delta z = \frac{z_{true} - z_{phot}}{1+z_{true}}$, and are:

- Bias: Mean of differences between true and estimated redshift. $\overline{\Delta z}$
- Standard Deviation: Standard deviation of Δz. $\sqrt{\overline{(\Delta z - \overline{\Delta z})^2}}$
- Standard Deviation of Interquartile Range: The spread of the difference between true and estimated redshift for the middle 50% of differences divided by 1.349 to compare

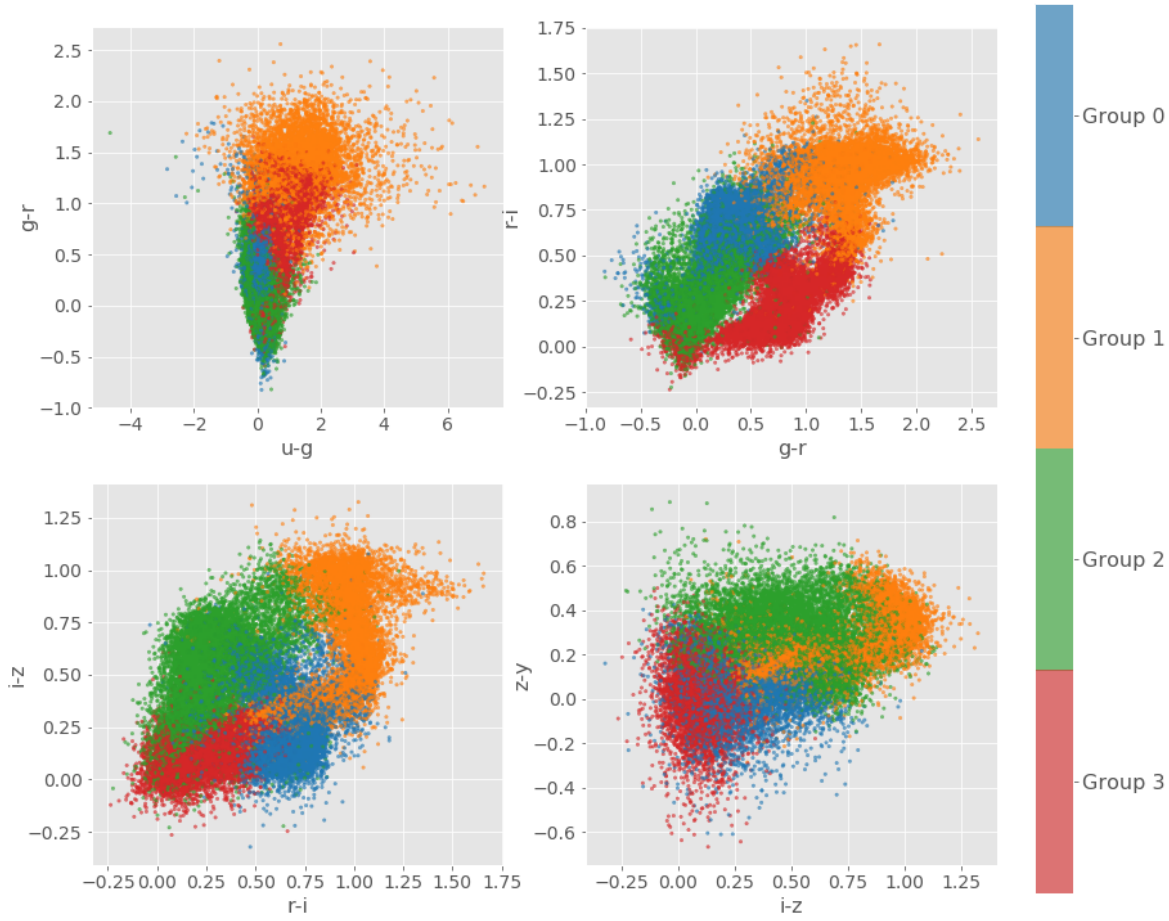

Figure 4.2: 4 groups in color-color space created by K-Means Clustering. Group 0 was targeted for our experiments.

to standard deviation. $\frac{\Delta z_{75\%} - \Delta z_{25\%}}{1.349}$

- Fraction of Outliers: The fraction of catalog objects with a Δz greater than 0.15.

We use these metrics to compare to the photometric redshift requirements over the interval $0 < z < 3$ for LSST which are: i) a maximum standard deviation of 0.05 with a goal of 0.02, ii) a fraction of 3σ outliers of $< 10\%$ at all redshifts and iii) a bias < 0.003. We set our outliers at 0.15 to be the maximum allowable 3σ value.

4.4 Photo-Z with Unrepresentative Color Space Training

We first used the CMNN method to observe the effects of removing the color space on the photo-z results. We ran photo-z on all 4 of our reduced coverage catalogs and compare the results for the full test set in Figure 4.3 and we present the results focused on the missing region of color space in Figure 4.4 and Table 4.1. Looking at the results we see that for every metric removing galaxies from the color space degrades the photo-z performance including the number of valid results that the CMNN method returns. In the same way, we notice that adding back more galaxies into the training sets improves photo-z performance consistently. Results in every metric improve at every step as we go from no galaxies in the removed color space to 10 then 100 and finally 1,000. The degradation in results as a color space is less than fully sampled is the problem we want to target with data augmentation by adding in synthetic data to the training sets.

To perform data augmentation on our training sets we explored two main approaches: 1) Generative Adversarial Networks (GANs) and 2) two different applications of the *ESP* code developed in Chapter 3. We describe the methods and results in the next section.

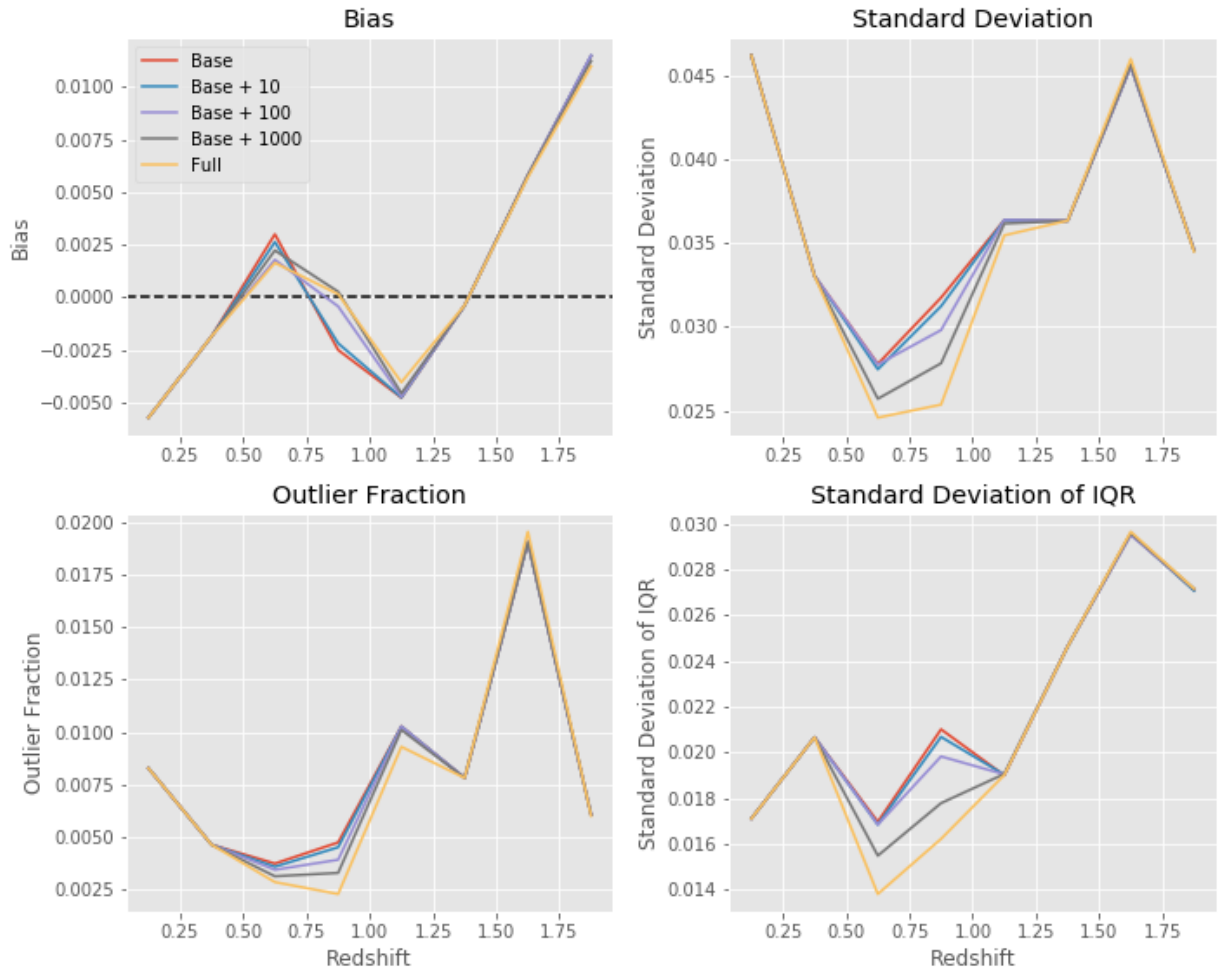

Figure 4.3: Color Matched Nearest Neighbor photo-z results on the full test set comparing the results from the full training set and training sets with 0 (Base), 10, 100 and 1,000 galaxies added into the removed "Group 0" region of color space.

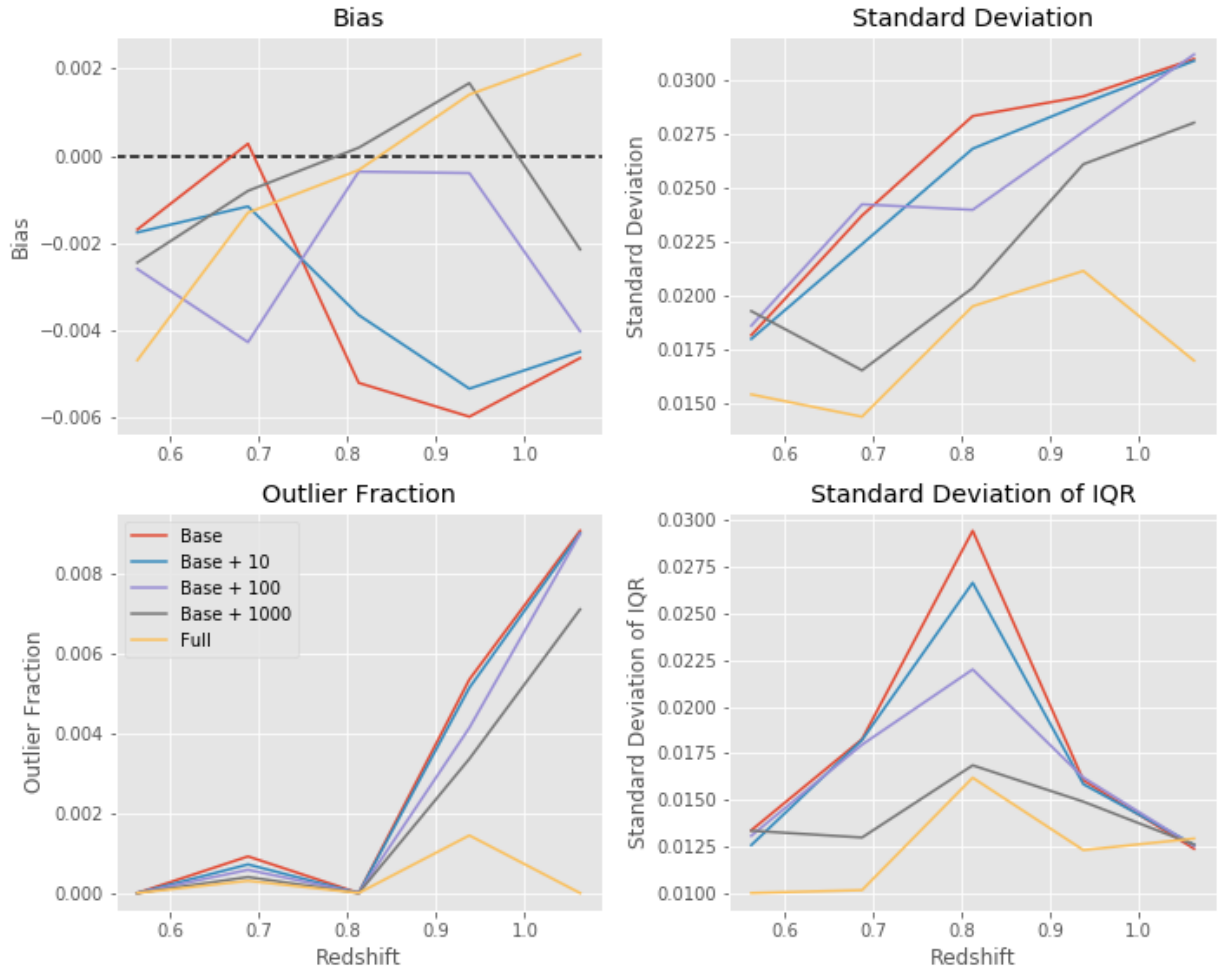

Figure 4.4: Color Matched Nearest Neighbor photo-z results on the test set galaxies in the sparsely sampled region of colors space. We compare the results from the full training set and training sets with 0 (Base), 10, 100 and 1,000 galaxies added into the removed "Group 0" region of color space. For clarity we only include the redshift range $0.5 \leq z < 1.125$ which includes over 95% of the test objects that fall in the removed color space.

Table 4.1: Photo-Z results for the test points in the sparsely sampled region of color space comparing catalogs with 0 (Base), 10, 100, or 1,000 galaxies added back out of 46,618 galaxies removed.

	Full	Base	Base + 10	Base + 100	Base + 1000
Bias	-1.3e-6	-0.0027	-0.0025	-0.0013	-0.0003
Standard Deviation	0.0200	0.0289	0.0278	0.0269	0.0229
Standard Deviation of IQR	0.0131	0.0198	0.0194	0.0182	0.0151
Outlier Fraction	0.0011	0.0038	0.0034	0.0027	0.0019
Returned Results	11,199	4,731	5,344	6,608	8,754

4.5 Photo-Z Augmentation Methods

4.5.1 Generative Adversarial Nets

Introduction to GANs

For our data augmentation we choose to generate data using a technique known as Generative Adversarial Nets (GANs) and first presented by Goodfellow et al. (2014). The basic GAN framework is made up of two competing Artificial Neural Networks (ANNs). ANNs are machine learning methods with an architecture inspired by the human brain and they are able to find complex patterns in data (we include a more detailed introduction to ANNs in § 4.8.1). The first ANN is known as the *generator*, G, and takes input from a noise distribution $p_{\mathbf{z}}(\mathbf{z})$ and outputs data $G(\mathbf{z})$ based upon a learned mapping designed to generate output that looks like it belongs the input data distribution $p_{\mathbf{x}}(\mathbf{x})$ we wish to learn. To determine whether the output looks like the original data distribution we train a second network known as the *discriminator*, D. The discriminator takes in real data from $p_{\mathbf{x}}(\mathbf{x})$ and fake data from the generator ($G(\mathbf{z})$). It then outputs a single parameter between 0 and 1 which is the probability of the input data belonging to $p_{\mathbf{x}}(\mathbf{x})$. In each training loop the

discriminator updates to maximally assign the correct labels to actual data, $p_{\mathbf{x}}(\mathbf{x})$, and the output from the current state of the generator, $G(\mathbf{z})$. Then in the same pass through the training loop we update the generator to fool the discriminator better by minimizing the loss function, $\log(1-D(G(\mathbf{z})))$. When the generative model is properly trained the discriminator will reach a point where it is no longer able to accurately determine what is false and its output will be ~ 0.5 for input from both $p_{\mathbf{x}}(\mathbf{x})$ and $G(\mathbf{z})$.

Since the development of the original GAN – often called Vanilla-GAN – there have been a proliferation of alternative GAN algorithms. Some of these algorithms have achieved notable success in the field of computer vision with GANs being developed for tasks ranging from upscaling images to higher resolutions (Ledig et al. 2016) to image-to-image translations (e.g. converting day photos to night or black and white photos to color) (Isola et al. 2016).

Using GANs on images is an active area of development in astronomy as well. Schawinski et al. (2017) trained GANs on postage stamps of galaxies and then used the GAN generator to reconstruct estimates of the original image from postage stamps with artificially degraded seeing and noise properties. The GAN was able to recover features from the noisy images at a level that outperformed traditional deconvolution. Stark et al. (2018) used GANs to measure flux in AGN and the extended host galaxy separately. This used a conditional GAN to generate images of the galaxy on its own based upon an images of the host galaxy plus AGN point source. Other applications of GANs involve retrieving exoplanetary atmospheres (Zingales and Waldmann 2018) and quickly generating large cosmic web simulations with GANs (Rodríguez et al. 2018).

GAN for Photo-Z

For our photometric redshift GAN our goal is to create a neural network generator that can produce new catalog samples of colors and redshifts that match the distributions present in the original training data. For our GAN architecture we designed 2 basic MLP-type neural networks to work as the generator and discriminator based upon the original algorithm of Goodfellow et al. (2014). We use the program libraries of pytorch (Paszke et al. 2017) to

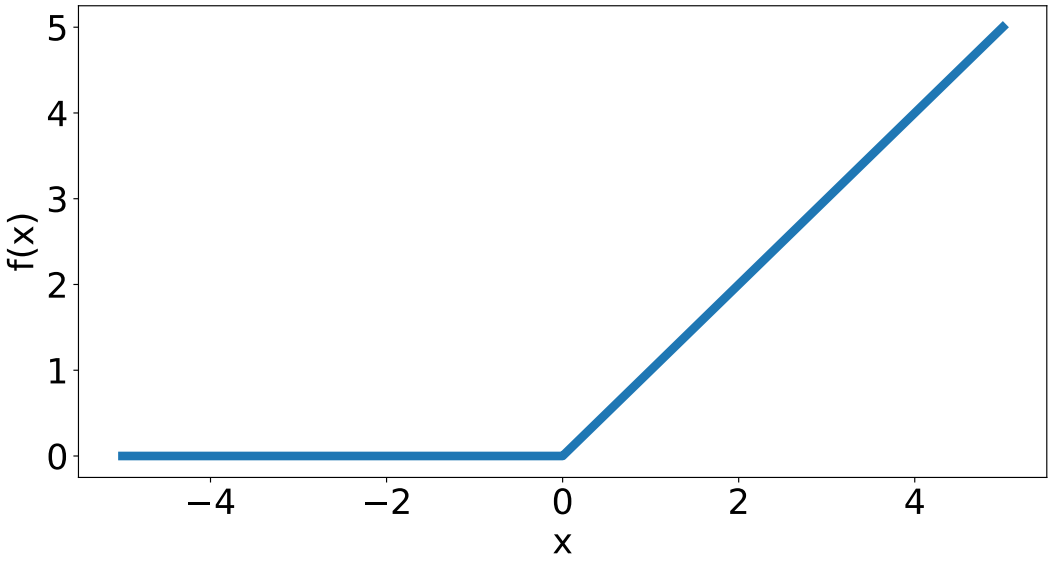

Figure 4.5: The Rectified Linear Unit (ReLU) function

construct and train our model. Pytorch is an open-source set of python libraries and widely used in industry. Our discriminator models consist of an input layer that accepts the colors and redshifts from true or generated galaxies and passes them to three hidden layers of 12, 24, and 12 nodes respectively. This represents 2x or 4x the number of input dimensions of the original data and is designed to be small enough to prevent overfitting and to keep training time down. The last layer connects to an output layer consisting of a single node that is activated by a sigmoid function and returns the probability of a real or generated galaxy. The network is fully connected, meaning that all nodes in one layer are connected to all in the adjacent layers. The activation function for all hidden nodes is the Rectified Linear Unit (ReLU) function defined as $f(x) = \max(0, x)$ and shown in Figure 4.5.

For the generator we use the same layout of hidden nodes (12, 24, 12) but our input node is a six-dimensional vector where each value is drawn from an independent 1-d random unit normal distribution. The output is also a six-dimensional vector but is the generated redshift and colors of an artificial data point. This output is designed to be a sample from the

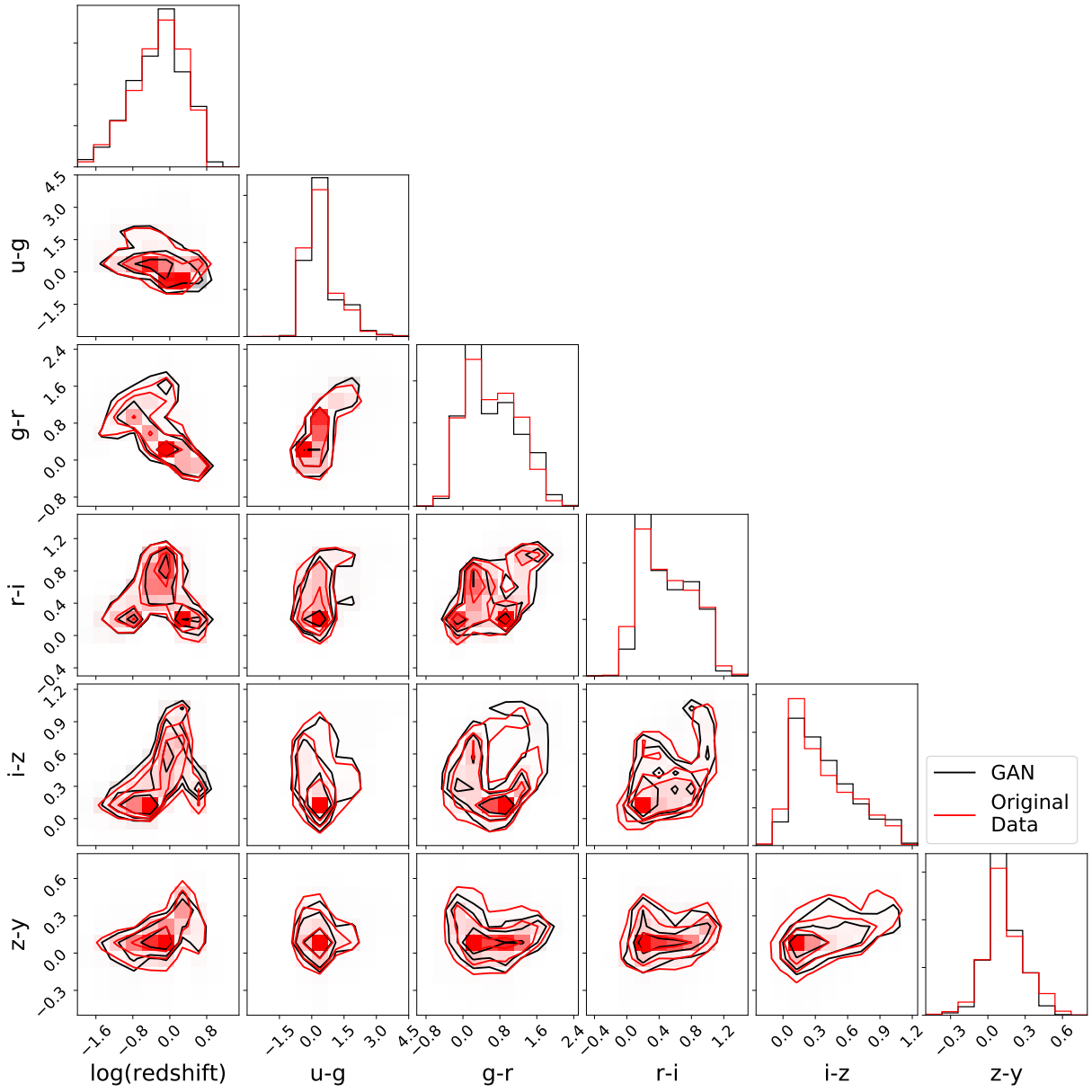

Figure 4.6: Comparing a GAN generated catalog to the original training catalog of 200,000 simulated galaxies with colors (in mags) and redshifts after 500 epochs of training. The GAN catalog learns to generate data that mimics the relationships in the training data as shown by the density plot overlays comparing the original training catalog to a GAN generated catalog of equal size.

redshift-color space of the original catalog and more closely resembles the true distribution as the training evolves. We use a GPU to speed up training and minibatches of 512. To avoid negative redshifts in generated output we train with the logarithm of the redshift. Finally, as an example we run 500 epochs of our GAN architecture on the full training set to show how the GAN performed. Figure 4.6 shows how the 200,000 points of generated data starts to approach the true catalog.

4.5.2 Augmentation with ESP

To apply data augmentation with templates we use a similar technique as in §3.5. We use our code *ESP* to take a template set and create a set of eigenspectra and fit a Gaussian Process to the PCA coefficients of the template set using the colors of the templates and the input variables. Two main differences from previous work are, the use of a larger template set and the application to redshifted color space. The new templates are the 129 templates from Brown et al. (2014) with emission lines masked and the masked regions interpolated. To calculate redshifted templates we redshift the original templates and compute the colors. We then fit a Gaussian Process with these colors and the unchanged PCA coefficients. The PCA coefficients are unchanged because the eigenspectra have not changed but the observed wavelengths of the spectrum are different based upon the actual redshift. In this way we can calculate a template at a given redshift and use the same template at a series of redshifts by simply changing the observed wavelengths for the spectrum appropriately. We need to be able to calculate the PCA coefficients at different redshifts because we want to generate new data from templates derived from training set galaxies. The galaxies in the training set are measured in their observed colors not in the rest frame so we calculate the templates at the appropriate redshift for the colors observed.

We tried two different methods for data augmentation with templates. The first method used all the available data in the region *outside* of the censored color space and as a result we call it the *Exterior Template* method. We started by binning all available training data in the exterior region by redshift. In each redshift bin we fit a new Gaussian Process for the

redshifted template colors and the unchanged PCA coefficients. For all training data in that bin we then calculated PCA coefficients for each training point and calculated an estimated SED. We then blueshifted the SED into the $z = 0$ rest frame and calculated the corresponding colors. Doing this over the whole training catalog transformed the redshifted training data all into a rest frame approximation of the color space covered by all the galaxies in the catalog. We took our derived rest frame color space from the available training data and used mini batched K-Means clustering to identify 150 regions of the rest frame color space. We used *ESP* to generate templates for the 150 locations in color space. Our motivation to do this is that the region of color space being cut out corresponds to galaxies with a certain redshift distribution but similar galaxies at different redshifts would lie in a different region of color space (see the tracks traced out by galaxies at a variety of redshifts in Figure 2.6). For these new interpolated templates we can redshift and calculate colors in the observed catalog color space and see the color tracks go through the target area. This provides us with information on the color-redshift relation in the missing region. We then added this redshift-color data into the training catalog and used this augmented catalog for photo-z.

When there was training data in the region of color space targeted we applied ESP in an alternative way. In this case we took the limited data available inside the sparsely sampled color space and only created templates based upon the redshift and color of the available data. We then calculated the colors of this handful of templates across the redshift range of the overall training catalog. In this case we added a much more limited set of new redshift-color data to the original training catalog based upon the smaller number of templates before running the photo-z code. Since this method uses the points *inside* the target region of color space we call this method the *Interior Templates* method.

4.5.3 Augmenting catalogs with no coverage

In our first experiment we used a catalog where we removed all objects in the color space cluster and used the remaining objects from the other clusters as the training set. Since in this case there is no data for the GAN to approximate the distribution we only use the *ESP*

based template generation method in this experiment. We then use the method described in §4.5.2 to generate 150 new templates. We then redshift these new templates to generate augmented data across the redshift range of the training catalog. These augmented data are shown in Figure 4.7 with the labels according to the color space assigned by the K-Means clustering. The figure shows that the augmented data provide training samples in the removed color space identified as "Group 0". As an approximate way to augment the catalog data where it would be needed we select the "Group 0" augmented data from the labels predicted by the K-Means function we used to remove the data. While in a real situation we would only have an approximate idea of where the missing color space exactly would be we use this method as a way to provide a proof of concept test for our method. In the future we will need to develop a way to select which points to keep in a more realistic way.

Results

Adding the additional data into the training set we then run the photo-z codes and examine the results. The first impact we notice is that over the full data set 1,045 more test points return a result due to the improved coverage. Without the additional training points the chance that test points within the removed color space are close enough for a training point to be within the distance threshold of the CMNN method is reduced. For reference, the full training set has an additional 6,487 matches for a total of 47,608 test points that return a result. This means that the augmented catalog provides a gain of 16% towards full training set coverage. If we just look at the test points that are in the color space we removed from the training catalog we find that all but four of the newly matched test points are in this area. Moving to our actual performance metrics we see the results on the full test set in Figure 4.8. Figure 4.9 only includes results for test points in the area of color space we removed and highlights the redshift range ($0.5 \leq z < 1.125$) that includes 95% of these test results. Looking at this figure and at Table 4.2 we see that the data augmentation degrades results in bias and in the IQR standard deviation, but improves the overall standard deviation and

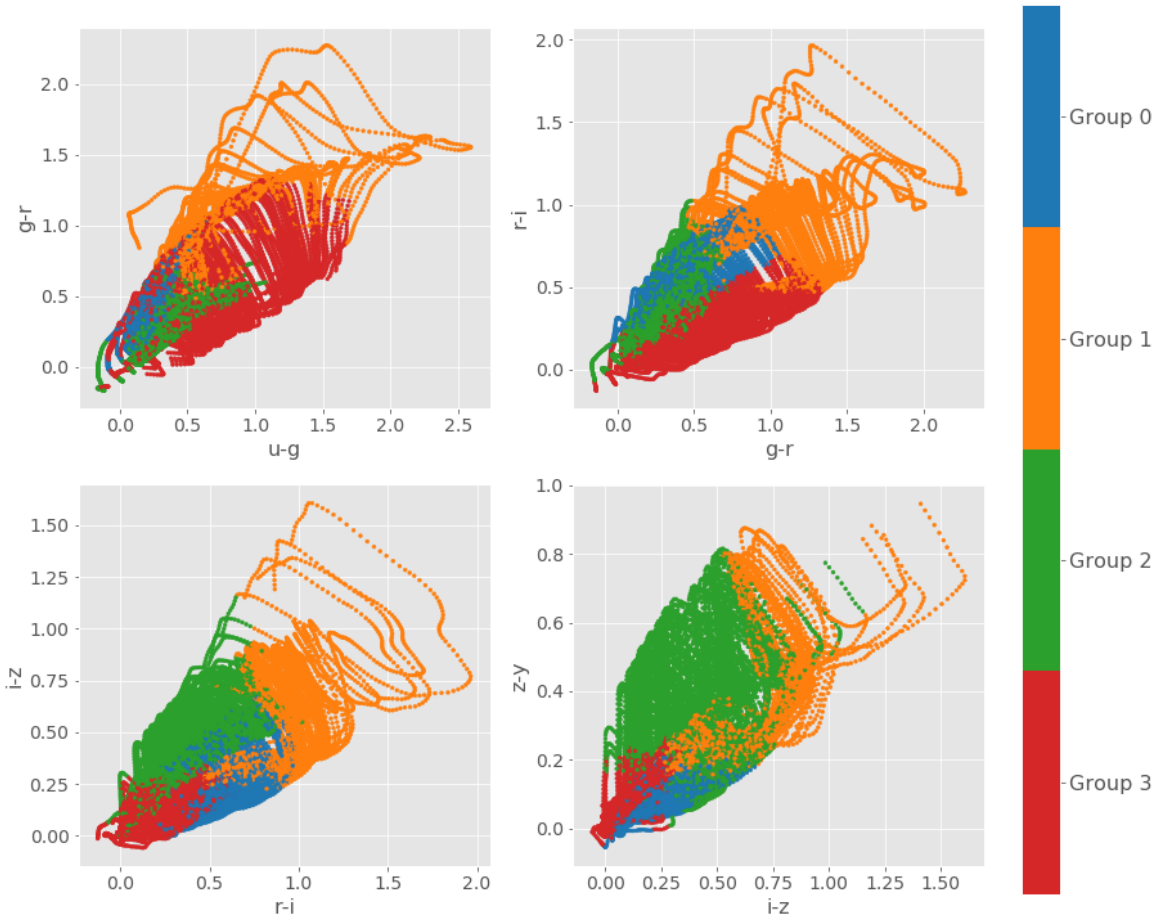

Figure 4.7: The new training data supplied by the templates fit from the training catalog with no training coverage in Group 0. Notice that the generated data supplies training samples in Group 0 now.

Table 4.2: Photo-Z results for the test points in the removed region of color space

	Full	Base	Exterior Templates
Bias	-1.3e-6	-0.0027	-0.0048
Standard Deviation	0.0200	0.0289	0.0279
Standard Deviation of IQR	0.0131	0.0198	0.0207
Outlier Fraction	0.0011	0.0038	0.0029
Returned Results (out of 11,784)	11,199	4,731	5,772

reduces outliers at the same time as increasing the number of results returned by the CMNN method. In fact, the data augmentation reduces outliers 22.6% in the removed color space while including 22.0% more results. The 76% increase in bias is harmful, especially since it takes the value outside the range of the LSST requirements. Looking at Figure 4.9 we see that around a redshift of 0.7 is where there is a large offset in the bias that drives this difference. At this redshift in the other metrics we see a larger IQR standard deviation but the outliers and standard deviation are actually lower. This might be because there is an added template that is well matched in color space to a group of test galaxies but is not quite the correct type of galaxy for this region of color space. This could be because we are not sampling the rest frame color space enough with 150 templates and we actually need more. The ideal number of templates to use with this method is a question that we need to explore further and answer in future work.

4.5.4 Augmenting catalogs with sparse coverage

In the next experiment we add back 10 data points into the section of color space we previously removed to measure what benefits data augmentation can provide after gathering 10 spectroscopic redshifts in this region. At the beginning of this chapter we highlighted the high cost of gathering spectroscopic redshifts so in this section we want to see how much

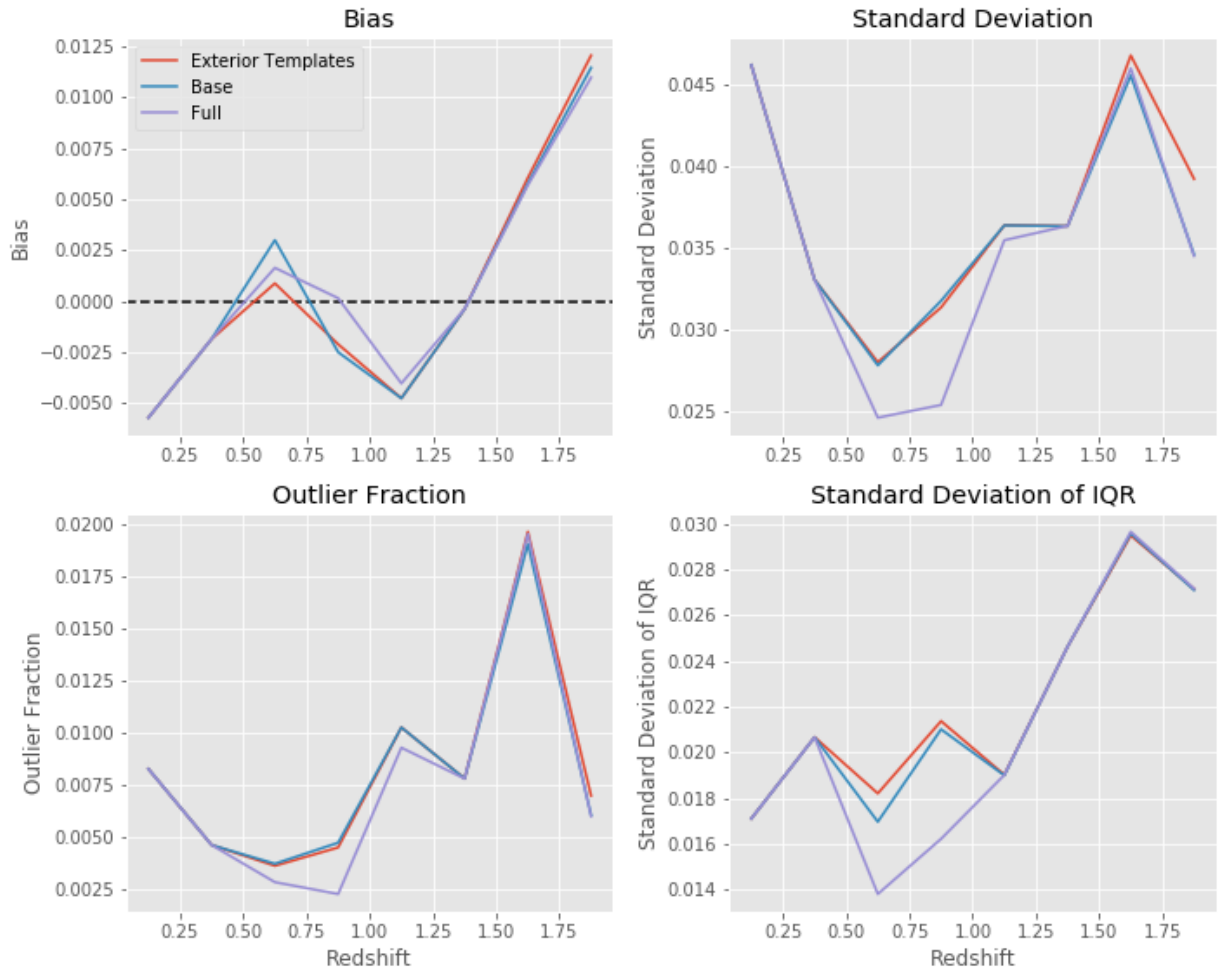

Figure 4.8: Color Matched Nearest Neighbor photo-z results on the full test set comparing the results from the full training set, the base training set with samples removed from a region of color space, and the template augmented training set.

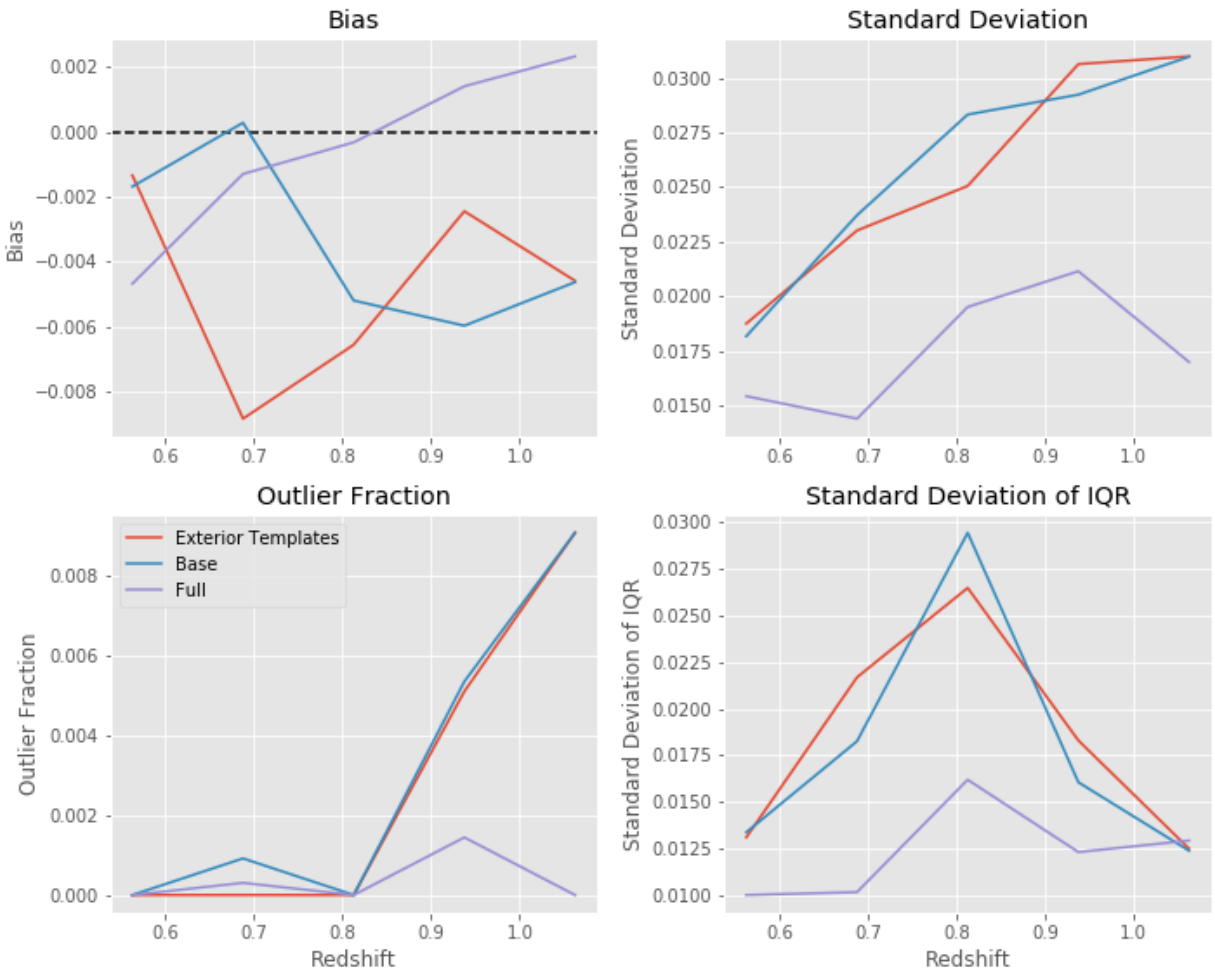

Figure 4.9: Color Matched Nearest Neighbor photo-z results on the test points only in the region of color space removed from the base training data. The plots compare the results using the full training set, the base training set with samples removed from a region of color space, and the template augmented training set.

benefit we can derive for photo-z from a minimal, relatively inexpensive set of spectroscopic redshifts supplemented with artificial data. The presence of a small amount of training data in the targeted region of color space enables the use of both our interior and exterior template methods and a test of using GANs for augmentation. For the exterior template method we use the same augmented catalog as the previous section and add in the 10 new training points. For the interior template method we start from templates derived for the 10 points now available in the color space training set. We then calculate the colors for these templates across the full redshift range of the catalog and keep the redshift-color combinations that remain inside the sparse color region.

To set up the GAN based method we take the base training catalogs with the 10 points in the sparse color space and include 2500 copies of these points so that they make up 25,000 points in a total catalog of 178,382 data points. We do this to increase the weight that the GAN training will give to this region of color space. We ran for 2000 epochs which took approximately 6 hours on a desktop machine with a GPU and saw the loss for the discriminator and generator approach convergence. Convergence occurs in a GAN when the generator produces data that fools the discriminator so well that it isn't sure what is real data and what is generated and can only assign 50% probability to all the data points.

Results

We ran photo-z with the CMNN code on the three new training catalogs and the new base catalog that includes 10 points in the previously removed color space (labelled *Base + 10*). We present the results in Figure 4.10 for the full color space. Figure 4.11 shows the results in the sparsely sampled color space and limited to same redshift range as Figure 4.9. The complete statistics for the sparsely sampled region of color space are detailed in Table 4.3.

Focusing on the test set performance in the region of sparsely sampled color space we first notice that the GAN catalog does not provide any improvement. In the positive sense, it also does not seem to adversely affect the results a significant amount only slightly degrading the bias and standard deviation by 0.0001 each. This is likely due to a failure of the GAN to

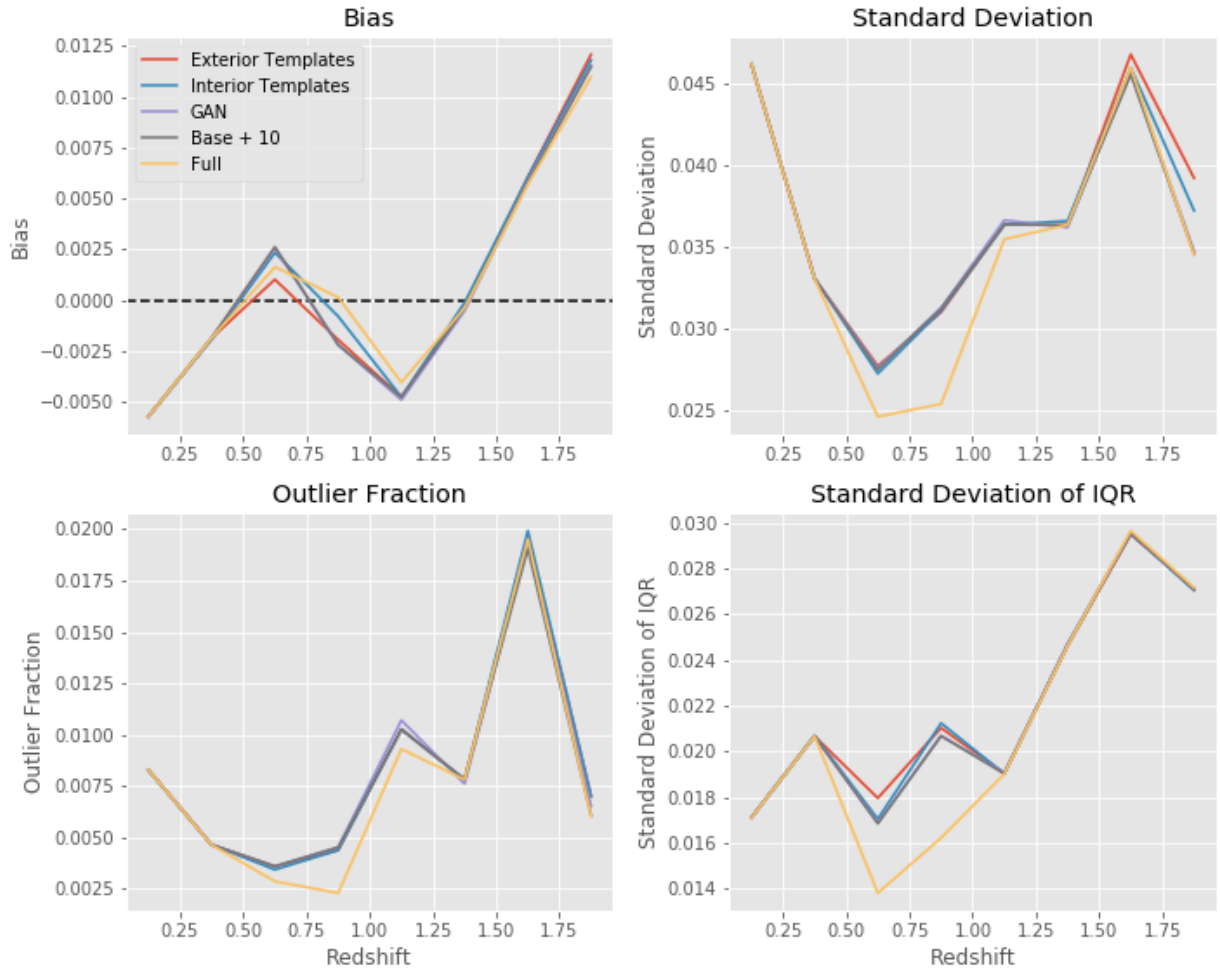

Figure 4.10: Color Matched Nearest Neighbor photo-z results on the full test set comparing the results from the full training set, the sparse training set with only 10 samples from a removed region of color space, the 2 template augmented training sets and the GAN augmented training set.

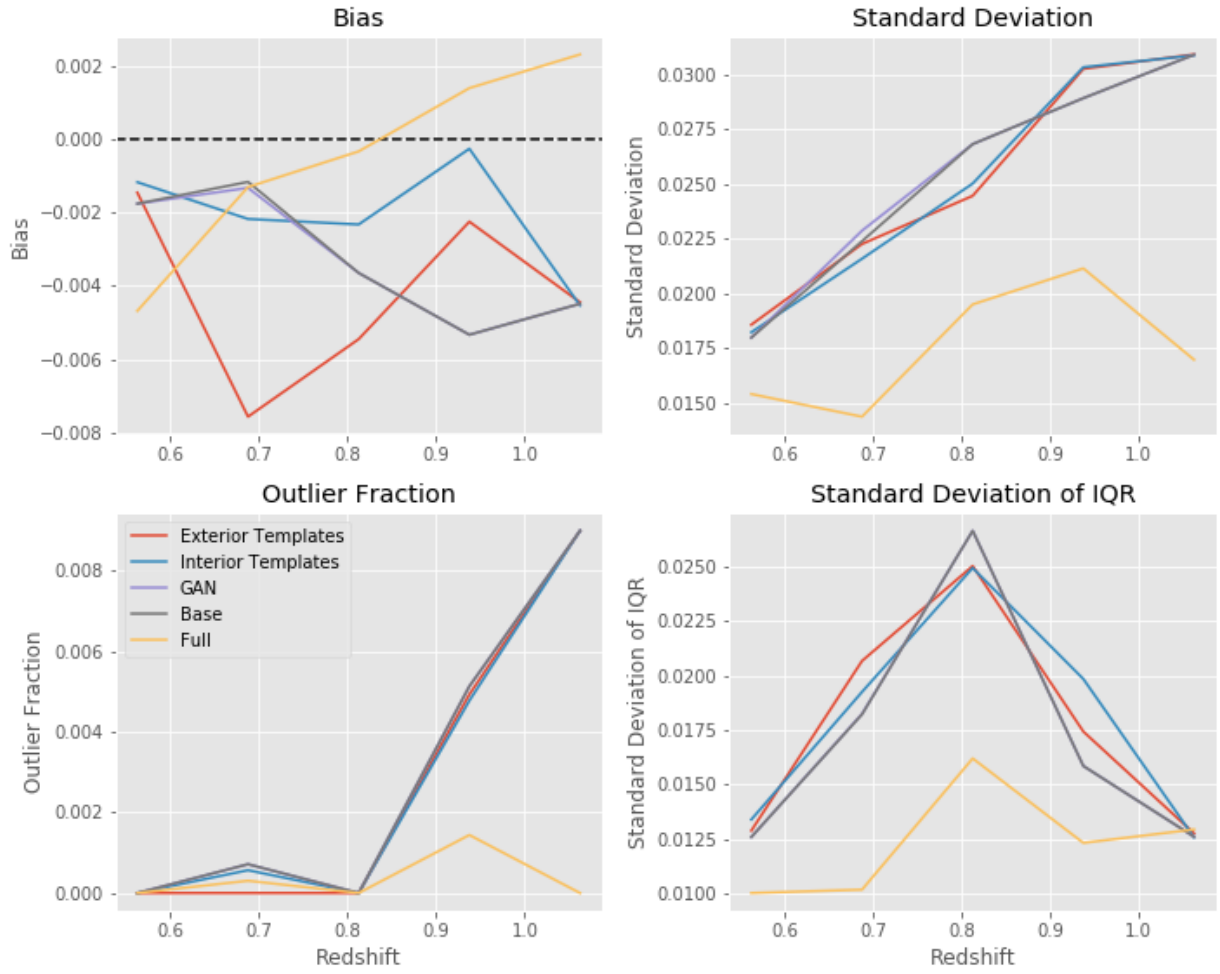

Figure 4.11: Color Matched Nearest Neighbor photo-z results on the test set comparing the results in the region of color space where we removed all but 10 points. The results come from photo-z using the full training set, the sparse training set with only 10 samples from a removed region of color space, the 2 template augmented training sets and the GAN augmented training set.

Table 4.3: Photo-Z results for the test points in the sparsely sampled region of color space

	Full	Base+10	Ext. Temp.	Int. Temp.	GAN
Bias	-1.3e-6	-0.0025	-0.0042	-0.0012	-0.0026
Standard Deviation	0.0200	0.0278	0.0272	0.0272	0.0279
Standard Dev. of IQR	0.0131	0.0194	0.0200	0.0200	0.0194
Outlier Fraction	0.0011	0.0034	0.0028	0.0030	0.0034
Ret. Results (out of 11,784)	11,199	5,344	6,102	5,909	5,344

extrapolate as we had hoped. We included the full training catalog outside the color space in an effort to see if the GAN would find a distribution that generated results that interpolated between the large gaps in color space. This did not happen as we show in Figure 4.12 where in the left plot we show that in the color space the GAN produced generated samples only around the few points available. This is in contrast to the entire region of missing color space show in the right plot.

On the other hand, the two template based methods did have a measurable effect on photo-z performance in the sparse region of color space. Both provided more successful outputs from the CMNN code with the exterior template method providing 758 more results which makes up 12.9% of the difference between the unrepresentative training set and the full training set. The exterior template method and the interior template method reduce the standard deviation by 2.0% in each case. They also both reduce the fraction of outliers with the exterior templates doing a better job with a 17.3% reduction compared to the 9.6% reduction from the interior templates. Both catalogs slightly perform worse in the standard deviation of the IQR but in both cases the degradation is $< 3\%$. The biggest difference between the two methods is that the interior template method does provide a large benefit in the bias reducing it by over 50% compare to the sparse catalog while the exterior templates make the bias 84% worse. The exterior template method has mixed results here as in the

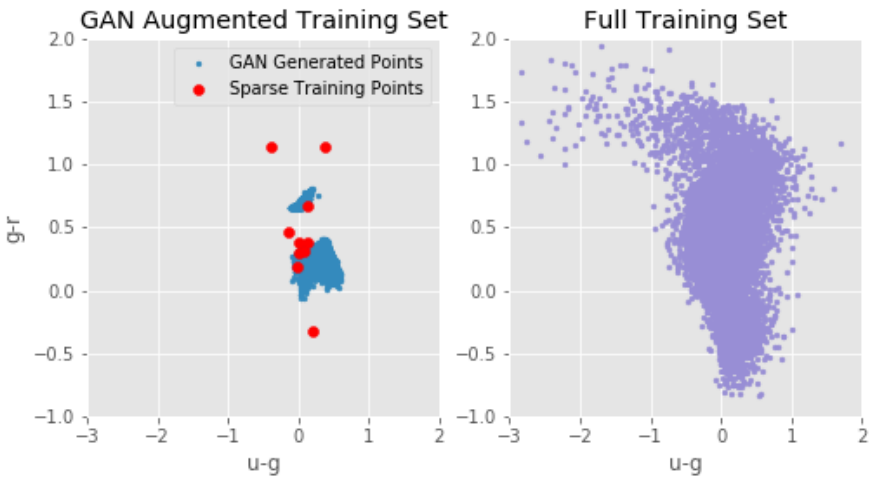

Figure 4.12: The *u-g* vs. *g-r* training points in the sparsely sampled region of color space in our experiments. Notice how the GAN training catalog only produces samples very near to the training points provided in this region of color space even though there are many training points outside of this space that could provide a basis for interpolation. Left: The training points in the GAN augmented training set. Right: The training points available in the full training set.

case with no available training data, but the interior template method appears to be have an overall positive effect. The interior templates improve bias, outlier fraction, and standard deviation with more valid photo-z results than the 10 training points alone.

4.6 Discussion

Overall, it seems that the interior template method provides the best increase in photo-z performance when augmenting a training catalog that does not match the color space coverage of the test catalog. There remains, however, a large amount of improvement available to approach the performance of a well-matched complete training catalog. Comparing the results in Tables 4.1 and 4.3 we see that the interior template method is only able to make up the difference of going from 10 to 100 templates in the bias and is well short of making up

the difference between 10 templates in the color space and the full training set. The exterior templates provide benefits in increasing the number of valid photo-z results using the CMNN method and reducing outliers at the same time but suffers from a significant increase in bias. As we noted above, changing the number of templates we use to span our estimate rest frame color space may be one way to fix this. This is something to look at in our future work. In order to try to get the added improvement in reducing outliers from the exterior templates with the bias improvement of interior templates we did try to merge the two methods into one catalog but this did not provide a combination of maximal improvement in both metrics as hoped.

Both of the template based methods we explored show potential and are worth exploring further with the goal of better approaching the performance of a full training set. One of the next steps is to evaluate the performance of these methods over different areas of color space. In this work we only examined cutting out a particular piece of color space, but based upon the features that contribute to give galaxies different colors at different redshifts it would be interesting to categorize the performance as a function of the color space.

The GAN method can reproduce the large scale features of the catalog, but when we try to use it to replicate the features in a sparsely sampled area it fails even when oversampling the sparse points. We used oversampling to try and increase the presence of the sparse points in the dataset but this did not cause the GAN to generate samples in the regions of color space between sparse points. It may be that the Vanilla-GAN optimiziation which relies upon a probability of classification for the discriminator network is not the right architecture for this problem. One of the known problems of the Vanilla-GAN is known as mode collapse where the generator finds an area of the distribution that will always fool the discriminator and only selects samples from this point. We may be encountering a similar problem where we have discrete points located in an isolated region of color space and the generator does not attempt to interpolate between the regions of color space The generator only provides points very similar to the points already there. Alternate implementations of GANs like the Wasserstein-GAN (Arjovsky et al. 2017) are designed to overcome mode collapse issues and

may provide better results in future work.

4.7 Conclusion

In this work we attempted to mitigate the negative effects of unrepresentative training sets in photometric redshift estimation with data augmentation. We implemented data augmentation with Generative Adversarial Networks which have been successful augmenting image datasets but did not have success with the Vanilla-GAN method. We hope to try other GAN implementations to potentially open up a new avenue in data augmentation of astronomical catalogs through deep learning.

We showed that our template estimation method from Chapter 3 can be used for data augmentation of unrepresentative training sets. We came up with two possible applications of the method. The method we called the *Exterior Template* approach showed that when there is no coverage at all in a region of color space we could reduce the fraction of outliers 22.6% in the censored color space while achieving a valid photo-z for 22.0% more results from the output of the CMNN photo-z code. When 10 training samples were added to the space the exterior template method still lowered the standard deviation by 2.0% and reduced outliers in the region by 17.3% while also increasing the number of matched test results by 14.2% over the sparse catalog. Our *Interior Template* approach lowered the bias 51% and reduced outliers 9.6% with a greater number of valid photo-z results than the base catalog. This shows that our template based data augmentation methods are able to increase the number of reliable results for empirical photometric redshift estimation in areas of poorly sampled color space.

4.8 Supplementary Information

4.8.1 Artificial Neural Networks

The GAN used in this chapter is an application of artificial neural networks. Artificial neural networks are machine learning methods that relate sets of input data with outputs through

a series of interconnected layers of nodes. The basic model for this type of machine learning and the one we will be using in this work is the multilayer perceptron (MLP) also known as a feedforward network. The feedforward name came from the fact that these models take input $\mathbf{x}$ and pass it through a series of computations to derive an output $\mathbf{y}$ without any feedback from the output back into the model (Goodfellow et al. 2016).

Figure 4.13 shows an example architecture for an MLP where the data passes through the network from left to right. The input data enters an input layer on the left and the middle layers–commonly known as hidden layers since their output is not exposed–gather input from the nodes in the layer to the left, transform it, and pass it on until finally producing final values in the output layer. The connections between nodes represent where output from a node is passed as input to a node in the next layer. In each of these nodes the input data for that node is weighted and then summed along with a bias term before being evaluated with a non-linear activation function that provides the output value for that node to pass on to the next layer. For example, in a single node if the inputs from the nodes in the previous layer are $\mathbf{x_1}, ..., \mathbf{x_n}$ with weights $w_1, ..., w_n$ we can include a constant bias term b with weight w_0. Using the activation function $\mathbf{A}$ then the output of the node $\mathbf{y}$ is given by the following equation:

$$\mathbf{y} = \mathbf{A}(w_0 b + \sum_{1}^{n} w_i \mathbf{x_i}) \tag{4.1}$$

The weights (including w_0) are values that we learn by training the MLP with a desired loss function. As with traditional optimization, learning is achieved through minimizing the loss function by gradient descent and using the backpropagation algorithm. The design of an MLP model requires choosing the number of layers and nodes, the activation functions, and loss function (Goodfellow et al. 2016). Choices in the design of an MLP can make a significant difference in the effectiveness of the model.

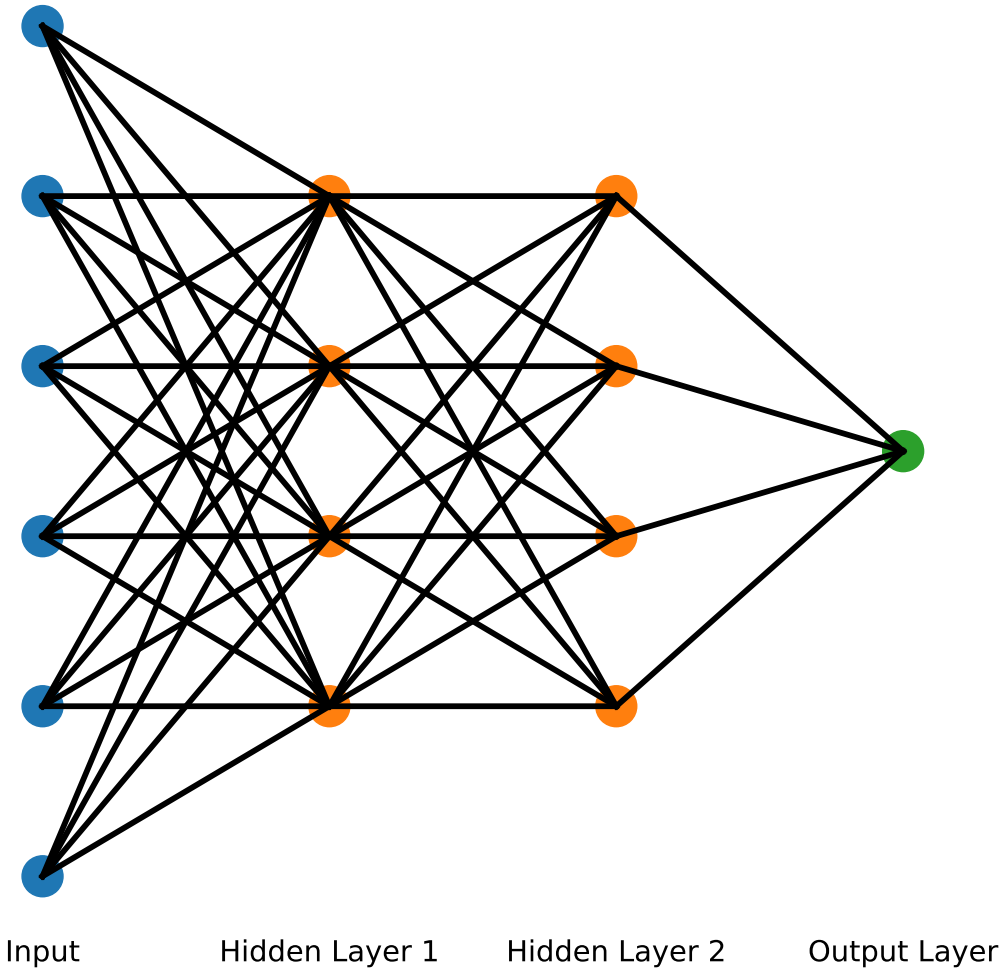

Figure 4.13: Example of a multilayer perceptron network with 2 hidden layers.

Mini-Batch Gradient Descent

Gradient descent is a common optimization algorithm where the gradient of a function is used to update the parameters towards the values that will minimize the function. The change in the parameters is controlled by a learning rate α so that a single update in the algorithm for a function $f(\mathbf{x})$ looks like $\mathbf{x} := \mathbf{x} - \alpha * \nabla_{\mathbf{x}} f(\mathbf{x})$. In the training of a neural network the function we are minimizing is the loss function we have chosen in our design and we are updating the weights within the neural network each time we iterate through the gradient descent algorithm. In addition, the learning rate is a hyperparameter of the model that must also be tuned carefully.

When all of the training data are used for an update of the weights it is called batch gradient descent, but we can also use a single value of the data each time ("stochastic" or "online" gradient descent) or we can use subsets of the data each time to update ("mini-batch" gradient descent). Smaller batch sizes converge more quickly in terms of overall computation but going all the way down to single samples underutilizes the potential for parallelization in the calculation (Goodfellow et al. 2016). As a result, mini-batch gradient descent is a good middle ground in practice and used in the networks described in this work.

Back-Propagation

The back-propogation algorithm (Rumelhart et al. 1986) is the method we use to update training weights in the network and improve the model after the feed forward of the data through the network of weights and non-linear activation functions. This algorithm is designed to minimize the loss function used when comparing the output of the network ($\hat{y}_{ij}$) to the true values from the training data (y_{ij}), $Loss = L(\hat{y}_{ij}, y_{ij})$, where i is index of the training examples and j is the index over the vector of values in a single output. If the output is one-dimensional then the loss function simplifies to $L(\hat{y}_i, y_i)$.

For instance, in the simple network shown in Figure 4.14 the output is $y_{out} = A(w_2 \cdot x_h) = A(z_{out}))$ where A is the activation function and we set $w_2 \cdot x_h = z_{out}$. The loss is then

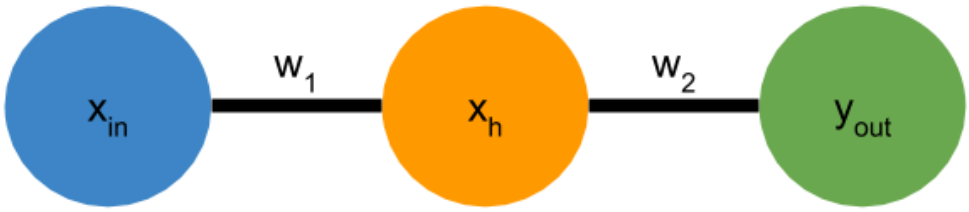

Figure 4.14: Simple neural network with one-dimensional input and output and one hidden layer.

$L(y_{out}, y_{true})$. To calculate an update for weight w_2 we start by taking the derivative of L with respect to the weight and using the chain rule, $\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial A} \cdot \frac{\partial A}{\partial z_{out}} \cdot \frac{\partial z_{out}}{\partial w_2}$. Then we use gradient descent with a learning rate set to α to modify the weight: $w_2 := w_2 - \alpha * \frac{\partial L}{\partial w_2}$

Astronomical Applications

As mentioned in §1.1.2 artificial neural networks have been used for photometric redshift estimation since Firth et al. (2003). The *ANNz* (Collister and Lahav 2004) and *ANNz2* (Sadeh et al. 2016) codes use the MLP network architecture to relate photometric inputs to redshifts. Recent work to calculate photometric redshifts directly from postage stamp images of galaxies by Hoyle (2016) and Pasquet et al. (2019) use a different type of neural network architecture known as the convolutional neural network (CNN). CNNs use adjustable kernels over images to create different feature maps in the next layer of the network (Pasquet et al. 2019). Due to their effectiveness on 2-d image data CNNs are also useful for detection on postage stamps like the Deep-HITS transient detection for the High Cadence Transient Survey (HiTS) (Cabrera-Vives et al. 2017) or for classification directly from images like the star-galaxy classification demonstrated by Kim and Brunner (2017). A final class of neural networks that are used in astronomy are recurrent neural networks (RNNs). RNNs are neural networks that specialize in sequential data and are able to handle data of different lengths

thus making them ideal for time-series data (Goodfellow et al. 2016). For instance, Naul et al. (2018) used RNNs to classify variable stars from unevenly sampled time-series data.

Chapter 5

CONCLUSION

We have drawn together a variety of state of the art machine learning and information theory methods to improve photometric redshift performance in this work. In Chapter 2 we developed a way to apply information theory to optimize the photometric bandpasses we use to make observations in a way that provided maximum benefit for photometric redshift estimation. To do this we optimized in the space of photometric colors derived from a range of templates of different galaxy types at different redshifts. Our six filters optimized for an LSST-like survey improved photo-z performance 3% overall at redshifts up to 2.3 over the LSST filters and outliers up to 7%. We also showed, however, that the LSST filters share common traits with an optimal filter set such as a small amount of overlap between each adjacent filter and a nearly top hat shape.

We then developed tools in Chapter 3 to enlarge template sets into broader regions of color space starting from a small training set of templates used for a template-based photometric redshift algorithm. Using the code we developed called *ESP* we can create realistic templates for a given location in color space better than alternate methods of estimating templates. We applied our method to photo-z estimation using a template based photo-z code, *LePhare* (Arnouts et al. 1999; Ilbert et al. 2006). Our results demonstrated that our method can improve the standard deviation of the error in photometric redshifts by over 24.8% and lower the outlier rejected bias by over 87.5% over the original template set.

Finally, we applied our template generation tools and a new machine learning method called Generative Adversarial Networks (GANs) in Chapter 4 to provide data augmentation of photo-z training sets that are missing regions of color space. We then used the augmented training sets with an empirical photo-z method and quantified the improvement. Our best

approach used the available galaxy colors and redshifts outside the missing color region of color space to generate a series of rest frame templates that describe the training set galaxies. We then redshifted these templates through the range of redshifts available in the training catalog and filled in the missing region of color space with new color-redshift data. Compared to a training set with zero coverage in the missing color space we could improve photo-z in the missing region by increasing the number of test galaxies that matched to a training point by 24.7% while at the same time reducing outliers in this space by 19.8%.

5.1 Future Work

5.1.1 Optimizing filters based upon Information Gain

In the end of Chapter 2 we mentioned that using templates that extend further into the UV range would let us investigate how optical filters could help break the Lyman vs Balmer break degeneracy in color space. We would also like to introduce more complex priors that could help our applied photo-z performance. We only used a simple redshift prior, but many options to enhance the priors we use exist. For example, instead of sampling each template with a uniform probability we could include a prior to weight certain templates of galaxy types more heavily at different redshifts. Introducing more advanced priors could help tailor our code to produce filters truly optimized for the practical application of photometric redshift estimation.

We would also like to extend the information gain methodology to design filters that optimize properties for stellar observations or quasar selection. For example, templates for different stellar types could be used to design filters that optimize observations to determine stellar properties. We applied our methodology to galaxies and photometric redshifts but we can easily apply it to any set of templates to find the ideal filters that will differentiate between the corresponding astronomical objects.

Finally, exploring different types of filters from the broadband filters we used in our work. We could look at how a large number of narrowband filters or a comb filter would

be optimized for photometric redshifts. Or we could move away from trapezoidal filters and allow more complex shapes in the design of an individual filter.

5.1.2 Estimating Spectra from Photometry

We have presented the results of our estimation technique to produce realistic and useful SEDs for studying galaxies. The technique relies upon Gaussian process regression which provides a measure of the variance around the mean estimate for each input value. In our method we only used the mean value for eigencoefficients from the GP to create new SEDs, but there is information in the variance results of the GP that we could use to quantify the uncertainty in our predicted SEDs. This uncertainty could help in deciding which areas of color space can be confidently extrapolated with our technique using a given training set. Or as mentioned in 3.4.2 we could use the information to combine the best estimates from a set of kernels to cover color space more completely and accurately than only using a single kernel. In Section 3.4.3 we showed our method is better than others at extrapolating to new areas of color space, but has limits due to the nature of the Gaussian Processes. Understanding the limits may come from studying the accompanying error estimates. We could potentially use this information to provide uncertainty estimates on our predicted SEDs and then add the information to Bayesian analyses like those used in photometric redshift estimation.

We would additionally like to improve our implementation of Gaussian Process regression to take into account that we expect correlations between the different PCA coefficients for the same galaxy. We currently fit a separate GP for each coefficient, but it would make sense to try and jointly estimate all the PCA coefficients for a galaxy. This will require modifying the existing GP regression algorithm in a new way.

Finally, we want to extend the use of the tools developed in this paper to other applications. The ability to describe a basis for constructing SEDs with PCA coefficients could allow galaxy evolution codes to retain an SED for a galaxy at each time step in a simulation. Storing only a set of around 10 PCA coefficients would be more practical in terms of memory use compared to storing the full spectrum at each time step. Furthermore, we could explore

possibly creating continuous distributions of galaxy SEDs in other feature spaces such as metallicity and age. This would help semi-analytical models of galaxies create more realistic color distributions in mock catalogs by providing SEDs that are not restricted to the finite grid of metallicity, age and other properties that simulated SEDs like BC03 currently allow.

5.1.3 Color Space Data Augmentation for Photometric Redshifts

We encountered some promising results from our template based augmentation methods but we have a large amount of possible improvement between our current results and the results available from a full training set. One possible way to improve things may be using different starting template sets or accounting for observational effects in the observed catalog colors. Any improvements to our template generation method will help since more accurate templates will give more accurate color-redshift values. In addition, we need to understand if the results are similar in different regions of color space or with different ranges of redshift.

One thing we used to simplify our experiments in Chapter 4 was to use our K-Means clustering to fill in the exact color space we removed. This oversimplification cannot be used when applying this to a true mismatch between a real training set and test set. Experimenting with true rather than simulated data will allow us to understand how best to do this in real situations.

We would like to see if different GAN architectures can deliver better results in the sparsely sample color space and possibly be used to improve results. In our experiment we removed over 46,000 training objects from the color space and added back only 10 meaning we added back only 0.02%. Perhaps at a much less sparse realization, say 10% or 4600 objects we would still have photo-z performance significantly below the full training set, but would also have enough data for the GAN to generate samples across a wider region of the missing color space.

BIBLIOGRAPHY

Adami, C. (2004). Information theory in molecular biology. *Physics of Life Reviews*, 1:3–22.

Aigrain, S., Hodgkin, S. T., Irwin, M. J., Lewis, J. R., and Roberts, S. J. (2015). Precise time series photometry for the Kepler-2.0 mission. MNRAS, 447:2880–2893.

Allen, R. L., Bernstein, G. M., and Malhotra, R. (2001). The Edge of the Solar System. ApJ, 549:L241–L244.

Almosallam, I. A., Jarvis, M. J., and Roberts, S. J. (2016). GPZ: non-stationary sparse Gaussian processes for heteroscedastic uncertainty estimation in photometric redshifts. MNRAS, 462:726–739.

Ambikasaran, S., Foreman-Mackey, D., Greengard, L., Hogg, D. W., and O'Neil, M. (2015). Fast Direct Methods for Gaussian Processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38.

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein GAN. *arXiv e-prints*, page arXiv:1701.07875.

Arnouts, S., Cristiani, S., Moscardini, L., Matarrese, S., Lucchin, F., Fontana, A., and Giallongo, E. (1999). Measuring and modelling the redshift evolution of clustering: the Hubble Deep Field North. MNRAS, 310:540–556.

Baum, W. A. (1962). Photoelectric Magnitudes and Red-Shifts. *Problems of Extra-Galactic Research*, 15:390–.

Benitez, N. (2000). Bayesian Photometric Redshift Estimation. *The Astrophysical Journal*, 536(2):571–583.

Bernstein, G. and Khushalani, B. (2000). Orbit Fitting and Uncertainties for Kuiper Belt Objects. AJ, 120:3323–3332.

Bernstein, G. M., Trilling, D. E., Allen, R. L., Brown, M. E., Holman, M., and Malhotra, R. (2004). The Size Distribution of Trans-Neptunian Bodies. AJ, 128:1364–1390.

Blake, C., Collister, A., Bridle, S., and Lahav, O. (2007). Cosmological baryonic and matter densities from 600000 SDSS luminous red galaxies with photometric redshifts. MNRAS, 374:1527–1548.

Blanton, M. R., Hogg, D. W., Bahcall, N. A., Brinkmann, J., Britton, M., Connolly, A. J., Csabai, I., Fukugita, M., Loveday, J., Meiksin, A., Munn, J. A., Nichol, R. C., Okamura, S., Quinn, T., Schneider, D. P., Shimasaku, K., Strauss, M. A., Tegmark, M., Vogeley, M. S., and Weinberg, D. H. (2003). The Galaxy Luminosity Function and Luminosity Density at Redshift z = 0.1. ApJ, 592:819–838.

Blanton, M. R. and Roweis, S. (2007). K-Corrections and Filter Transformations in the Ultraviolet, Optical, and Near-Infrared. AJ, 133:734–754.

Bolzonella, M., Miralles, J. M., and Pellò, R. (2000). Photometric redshifts based on standard SED fitting procedures. *Astronomy and Astrophysics*, 363:476–492.

Bosch, J. (2015). Algorithms for detection and coaddition. `https://github.com/lsst-dm/algorithm-docs/blob/master/2015-07-det%2Bcoadd-slides/slides.ipynb`.

Bosch, J., Armstrong, R., Bickerton, S., Furusawa, H., Ikeda, H., Koike, M., Lupton, R., Mineo, S., Price, P., Takata, T., Tanaka, M., Yasuda, N., AlSayyad, Y., Becker, A. C., Coulton, W., Coupon, J., Garmilla, J., Huang, S., Krughoff, K. S., Lang, D., Leauthaud, A., Lim, K.-T., Lust, N. B., MacArthur, L. A., Mandelbaum, R., Miyatake, H., Miyazaki, S., Murata, R., More, S., Okura, Y., Owen, R., Swinbank, J. D., Strauss, M. A., Yamada, Y., and Yamanoi, H. (2018). The Hyper Suprime-Cam software pipeline. PASJ, 70:S5.

Brammer, G. B., van Dokkum, P. G., and Coppi, P. (2008). EAZY: A Fast, Public Photometric Redshift Code. *The Astrophysical Journal*, 686(2):1503.

Brown, M. E. (2001). The Inclination Distribution of the Kuiper Belt. AJ, 121:2804–2814.

Brown, M. J. I., Moustakas, J., Smith, J. D. T., da Cunha, E., Jarrett, T. H., Imanishi, M., Armus, L., Brandl, B. R., and Peek, J. E. G. (2014). An Atlas of Galaxy Spectral Energy Distributions from the Ultraviolet to the Mid-infrared. *The Astrophysical Journal Supplement Series*, 212:18.

Bruzual, A. and Charlot, S. (1993). Spectral evolution of stellar populations using isochrone synthesis. ApJ, 405:538–553.

Bruzual, G. and Charlot, S. (2003). Stellar population synthesis at the resolution of 2003. MNRAS, 344:1000–1028.

Cabrera-Vives, G., Reyes, I., Förster, F., Estévez, P. A., and Maureira, J.-C. (2017). Deep-HiTS: Rotation Invariant Convolutional Neural Network for Transient Detection. ApJ, 836:97.

Calzetti, D., Kinney, A. L., and Storchi-Bergmann, T. (1994). Dust extinction of the stellar continua in starburst galaxies: The ultraviolet and optical extinction law. ApJ, 429:582–601.

Carliles, S., Budavári, T., Heinis, S., Priebe, C., and Szalay, A. S. (2010). Random Forests for Photometric Redshifts. *The Astrophysical Journal*, 712(1):511.

Carrasco Kind, M. and Brunner, R. J. (2013). TPZ: photometric redshift PDFs and ancillary information by using prediction trees and random forests. *Monthly Notices of the Royal Astronomical Society*, 432(2):1483–1501.

Chabrier, G. (2003). Galactic Stellar and Substellar Initial Mass Function. PASP, 115:763–795.

Cincotta, P. M., Mendez, M., and Nunez, J. A. (1995). Astronomical Time Series Analysis. I. A Search for Periodicity Using Information Entropy. *The Astrophysical Journal*, 449:231.

Coleman, G. D., Wu, C.-C., and Weedman, D. W. (1980). Colors and magnitudes predicted for high redshift galaxies. ApJS, 43:393–416.

Collister, A. A. and Lahav, O. (2004). ANNz: Estimating Photometric Redshifts Using Artificial Neural Networks. *Publications of the Astronomical Society of the Pacific*, 116(818):345–351.

Connolly, A. J., Angeli, G. Z., Chandrasekharan, S., Claver, C. F., Cook, K., Ivezic, Z., Jones, R. L., Krughoff, K. S., Peng, E.-H., Peterson, J., Petry, C., Rasmussen, A. P., Ridgway, S. T., Saha, A., Sembroski, G., vanderPlas, J., and Yoachim, P. (2014). An end-to-end simulation framework for the Large Synoptic Survey Telescope. In *Modeling, Systems Engineering, and Project Management for Astronomy VI*, volume 9150 of Proc. SPIE, page 915014.

Connolly, A. J., Csabai, I., Szalay, A. S., Koo, D. C., Kron, R. G., and Munn, J. A. (1995). Slicing Through Multicolor Space: Galaxy Redshifts from Broadband Photometry. AJ, 110:2655.

Denneau, L., Jedicke, R., Grav, T., Granvik, M., Kubica, J., Milani, A., Vereš, P., Wainscoat, R., Chang, D., Pierfederici, F., Kaiser, N., Chambers, K. C., Heasley, J. N., Magnier, E. A., Price, P. A., Myers, J., Kleyna, J., Hsieh, H., Farnocchia, D., Waters, C., Sweeney, W. H., Green, D., Bolin, B., Burgett, W. S., Morgan, J. S., Tonry, J. L., Hodapp, K. W., Chastel, S., Chesley, S., Fitzsimmons, A., Holman, M., Spahr, T., Tholen, D., Williams, G. V., Abe, S., Armstrong, J. D., Bressi, T. H., Holmes, R., Lister, T., McMillan, R. S., Micheli, M., Ryan, E. V., Ryan, W. H., and Scotti, J. V. (2013). The Pan-STARRS Moving Object Processing System. PASP, 125:357.

Ebden, M. (2015). Gaussian Processes: A Quick Introduction. *ArXiv e-prints*.

Eisenstein, D. J., Zehavi, I., Hogg, D. W., Scoccimarro, R., Blanton, M. R., Nichol, R. C., Scranton, R., Seo, H.-J., Tegmark, M., Zheng, Z., Anderson, S. F., Annis, J., Bahcall, N., Brinkmann, J., Burles, S., Castander, F. J., Connolly, A., Csabai, I., Doi, M., Fukugita, M., Frieman, J. A., Glazebrook, K., Gunn, J. E., Hendry, J. S., Hennessy, G., Ivezi, Z., Kent, S., Knapp, G. R., Lin, H., Loh, Y.-S., Lupton, R. H., Margon, B., McKay, T. A., Meiksin, A., Munn, J. A., Pope, A., Richmond, M. W., Schlegel, D., Schneider, D. P., Shimasaku, K., Stoughton, C., Strauss, M. A., SubbaRao, M., Szalay, A. S., Szapudi, I., Tucker, D. L., Yanny, B., and York, D. G. (2005). Detection of the baryon acoustic peak in the large-scale correlation function of sdss luminous red galaxies. *The Astrophysical Journal*, 633(2):560.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pages 226–231. AAAI Press.

Firth, A. E., Lahav, O., and Somerville, R. S. (2003). Estimating photometric redshifts with artificial neural networks. *Monthly Notices of the Royal Astronomical Society*, 339(4):1195–1202.

Flaugher, B., Diehl, H. T., Honscheid, K., Abbott, T. M. C., Alvarez, O., Angstadt, R., Annis, J. T., Antonik, M., Ballester, O., Beaufore, L., Bernstein, G. M., Bernstein, R. A., Bigelow, B., Bonati, M., Boprie, D., Brooks, D., Buckley-Geer, E. J., Campa, J., Cardiel-Sas, L., Castander, F. J., Castilla, J., Cease, H., Cela-Ruiz, J. M., Chappa, S., Chi, E., Cooper, C., da Costa, L. N., Dede, E., Derylo, G., DePoy, D. L., de Vicente, J., Doel, P., Drlica-Wagner, A., Eiting, J., Elliott, A. E., Emes, J., Estrada, J., Neto, A. F., Finley, D. A., Flores, R., Frieman, J., Gerdes, D., Gladders, M. D., Gregory, B., Gutierrez, G. R., Hao, J., Holland, S. E., Holm, S., Huffman, D., Jackson, C., James, D. J., Jonas, M., Karcher, A., Karliner, I., Kent, S., Kessler, R., Kozlovsky, M., Kron, R. G., Kubik, D., Kuehn, K., Kuhlmann, S., Kuk, K., Lahav, O., Lathrop, A., Lee, J., Levi, M. E.,

Lewis, P., Li, T. S., Mandrichenko, I., Marshall, J. L., Martinez, G., Merritt, K. W., Miquel, R., Muoz, F., Neilsen, E. H., Nichol, R. C., Nord, B., Ogando, R., Olsen, J., Palaio, N., Patton, K., Peoples, J., Plazas, A. A., Rauch, J., Reil, K., Rheault, J.-P., Roe, N. A., Rogers, H., Roodman, A., Sanchez, E., Scarpine, V., Schindler, R. H., Schmidt, R., Schmitt, R., Schubnell, M., Schultz, K., Schurter, P., Scott, L., Serrano, S., Shaw, T. M., Smith, R. C., Soares-Santos, M., Stefanik, A., Stuermer, W., Suchyta, E., Sypniewski, A., Tarle, G., Thaler, J., Tighe, R., Tran, C., Tucker, D., Walker, A. R., Wang, G., Watson, M., Weaverdyck, C., Wester, W., Woods, R., Yanny, B., and Collaboration, T. D. (2015). The dark energy camera. *The Astronomical Journal*, 150(5):150.

Förster, F., Maureira, J. C., San Martín, J., Hamuy, M., Martínez, J., Huijse, P., Cabrera, G., Galbany, L., de Jaeger, T., González-Gaitán, S., Anderson, J. P., Kunkarayakti, H., Pignata, G., Bufano, F., Littín, J., Olivares, F., Medina, G., Smith, R. C., Vivas, A. K., Estévez, P. A., Muñoz, R., and Vera, E. (2016). The High Cadence Transient Survey (HITS). I. Survey Design and Supernova Shock Breakout Constraints. ApJ, 832:155.

Fraser, W. C., Kavelaars, J. J., Holman, M. J., Pritchet, C. J., Gladman, B. J., Grav, T., Jones, R. L., MacWilliams, J., and Petit, J.-M. (2008). The Kuiper belt luminosity function from m=21 to 26. Icarus, 195:827–843.

Gladman, B. and Kavelaars, J. J. (1997). Kuiper Belt searches from the Palomar 5-m telescope. A&A, 317:L35–L38.

Gonzalez-Perez, V., Lacey, C. G., Baugh, C. M., Lagos, C. D. P., Helly, J., Campbell, D. J. R., and Mitchell, P. D. (2014). How sensitive are predicted galaxy luminosities to the choice of stellar population synthesis model? MNRAS, 439:264–283.

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

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S.,

Courville, A., and Bengio, Y. (2014). Generative Adversarial Networks. *arXiv e-prints*, page arXiv:1406.2661.

Gorecki, A., Abate, A., Ansari, R., Barrau, A., Baumont, S., Moniez, M., and Ricol, J.-S. (2014). A new method to improve photometric redshift reconstruction. Applications to the Large Synoptic Survey Telescope. A&A, 561:A128.

Graham, M., Connolly, A., Wang, W., Schmidt, S., Morrison, C., Ivezić, Z., Fabbro, S., Côté, P., Daniel, S., Jones, R., Jurić, M., and Yoachim, P. (2019). *submitted to ApJ*.

Graham, M. J., Drake, A. J., Djorgovski, S. G., Mahabal, A. A., and Donalek, C. (2013). Using conditional entropy to identify periodicity. *Monthly Notices of the Royal Astronomical Society*, 434(3):2629–2635.

Graham, M. J., Drake, A. J., Djorgovski, S. G., Mahabal, A. A., Donalek, C., Duan, V., and Maker, A. (2013). A comparison of period finding algorithms. MNRAS, 434(4):3423–3444.

Graham, M. L., Connolly, A. J., Ivezić, Ž., Schmidt, S. J., Jones, R. L., Jurić, M., Daniel, S. F., and Yoachim, P. (2018). Photometric Redshifts with the LSST: Evaluating Survey Observing Strategies. AJ, 155:1.

Grisel, O., Mueller, A., Pedregosa, F., Lars, Gramfort, A., Louppe, G., Prettenhofer, P., Blondel, M., Niculae, V., Joly, A., Nothman, J., Vanderplas, J., Kumar, M., Layton, R., Varoquaux, N., Dawe, N., Schnberger, J., Engemann, D. A., Li, W., V, R. R., Woolam, C., Eren, K., Eustache, Fabisch, A., Passos, A., bthirion, and Fritsch, V. (2016). scikit-learn: 0.17.1 release tag for doi.

Head, T., MechCoder, Louppe, G., Shcherbatyi, I., fcharras, Vincius, Z., cmmalone, Schrder, C., nel215, Campos, N., Young, T., Cereda, S., Fan, T., rene rex, Shi, K. K., Schwabedal, J., carlosdanielcsantos, Hvass-Labs, Pak, M., SoManyUsernamesTaken, Callaway, F., Estve, L., Besson, L., Cherti, M., Pfannschmidt, K., Linzberger, F., Cauet, C., Gut, A., Mueller, A., and Fabisch, A. (2018). scikit-optimize/scikit-optimize: v0.5.2.

Heinze, A. N., Metchev, S., and Trollo, J. (2015). Digital Tracking Observations Can Discover Asteroids 10 Times Fainter Than Conventional Searches. AJ, 150:125.

Hoyle, B. (2016). Measuring photometric redshifts using galaxy images and Deep Neural Networks. *Astronomy and Computing*, 16:34–40.

Hoyle, B., Rau, M. M., Bonnett, C., Seitz, S., and Weller, J. (2015). Data augmentation for machine learning redshifts applied to Sloan Digital Sky Survey galaxies. MNRAS, 450:305–316.

Hubble, E. (1929). A Relation between Distance and Radial Velocity among Extra-Galactic Nebulae. In *Proceedings of the National Academy of Sciences of the United States of America*, pages 168–173. Mount Wilson Observatory, Carnegie Institution of Washington.

Ilbert, O., Arnouts, S., McCracken, H. J., Bolzonella, M., Bertin, E., Le Fèvre, O., Mellier, Y., Zamorani, G., Pellò, R., Iovino, A., Tresse, L., Le Brun, V., Bottini, D., Garilli, B., Maccagni, D., Picat, J. P., Scaramella, R., Scodeggio, M., Vettolani, G., Zanichelli, A., Adami, C., Bardelli, S., Cappi, A., Charlot, S., Ciliegi, P., Contini, T., Cucciati, O., Foucaud, S., Franzetti, P., Gavignaud, I., Guzzo, L., Marano, B., Marinoni, C., Mazure, A., Meneux, B., Merighi, R., Paltani, S., Pollo, A., Pozzetti, L., Radovich, M., Zucca, E., Bondi, M., Bongiorno, A., Busarello, G., de La Torre, S., Gregorini, L., Lamareille, F., Mathez, G., Merluzzi, P., Ripepi, V., Rizzo, D., and Vergani, D. (2006). Accurate photometric redshifts for the CFHT legacy survey calibrated using the VIMOS VLT deep survey. A&A, 457:841–856.

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2016). Image-to-Image Translation with Conditional Adversarial Networks. *arXiv e-prints*, page arXiv:1611.07004.

Ivezić, v., Tyson, J. A., Acosta, E., Allsman, R., Anderson, S. F., Andrew, J., Angel, J. R. P., Axelrod, T. S., Barr, J. D., Becker, A. C., et al. (2008). Lsst: from science drivers to reference design and anticipated data products.

Jain, B., Spergel, D., Bean, R., Connolly, A., Dell'antonio, I., Frieman, J., Gawiser, E., Gehrels, N., Gladney, L., Heitmann, K., Helou, G., Hirata, C., Ho, S., Ivezić, Ž., Jarvis, M., Kahn, S., Kalirai, J., Kim, A., Lupton, R., Mand elbaum, R., Marshall, P., Newman, J. A., Perlmutter, S., Postman, M., Rhodes, J., Strauss, M. A., Tyson, J. A., Walkowicz, L., and Wood-Vasey, W. M. (2015). The Whole is Greater than the Sum of the Parts: Optimizing the Joint Science Return from LSST, Euclid and WFIRST. *arXiv e-prints*, page arXiv:1501.07897.

Johnston, L. A. and Krishnamurthy, V. (2002). Performance analysis of a dynamic programming track before detect algorithm. *IEEE Transactions on Aerospace and Electronic Systems*, 38(1):228–242.

Jurić, M., Kantor, J., Lim, K., Lupton, R. H., Dubois-Felsmann, G., Jenness, T., Axelrod, T. S., Aleksić, J., Allsman, R. A., AlSayyad, Y., Alt, J., Armstrong, R., Basney, J., Becker, A. C., Becla, J., Bickerton, S. J., Biswas, R., Bosch, J., Boutigny, D., Carrasco Kind, M., Ciardi, D. R., Connolly, A. J., Daniel, S. F., Daues, G. E., Economou, F., Chiang, H.-F., Fausti, A., Fisher-Levine, M., Freemon, D. M., Gee, P., Gris, P., Hernandez, F., Hoblitt, J., Ivezić, Ž., Jammes, F., Jevremović, D., Jones, R. L., Bryce Kalmbach, J., Kasliwal, V. P., Krughoff, K. S., Lang, D., Lurie, J., Lust, N. B., Mullally, F., MacArthur, L. A., Melchior, P., Moeyens, J., Nidever, D. L., Owen, R., Parejko, J. K., Peterson, J. M., Petravick, D., Pietrowicz, S. R., Price, P. A., Reiss, D. J., Shaw, R. A., Sick, J., Slater, C. T., Strauss, M. A., Sullivan, I. S., Swinbank, J. D., Van Dyk, S., Vujčić, V., Withers, A., Yoachim, P., and LSST Project, f. t. (2015). The LSST Data Management System. *ArXiv e-prints*.

Kaiser, N. (2004). *Addition of Images with Varying Seeing.* *http://spider.ipac.caltech.edu/staff/fmasci/home/astro_refs/PanStars_Coadder.pdf*.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45.

Kalmbach, J. B. (2018). jbkalmbach/siggi: Siggi v0.1.5.

Kim, E. J. and Brunner, R. J. (2017). Star-galaxy classification using deep convolutional neural networks. MNRAS, 464:4463–4475.

Koo, D. C. (1985). Optical multicolors - A poor person's Z machine for galaxies. AJ, 90:418–440.

Kubica, J., Denneau, L., Grav, T., Heasley, J., Jedicke, R., Masiero, J., Milani, A., Moore, A., Tholen, D., and Wainscoat, R. J. (2007). Efficient intra- and inter-night linking of asteroid detections using kd-trees. Icarus, 189:151–168.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86.

Lang, D., Hogg, D. W., Jester, S., and Rix, H.-W. (2009). Measuring the Undetectable: Proper Motions and Parallaxes of Very Faint Sources. AJ, 137:4400–4411.

Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., and Shi, W. (2016). Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. *arXiv e-prints*, page arXiv:1609.04802.

Leistedt, B. and Hogg, D. (2017). Data-driven, interpretable photometric redshifts trained on heterogeneous and unrepresentative data. *Astrophysical Journal*, 838(1).

Lenz, D. D., Newberg, J., Rosner, R., Richards, G. T., and Stoughton, C. (1998). Photometric Separation of Stellar Properties Using SDSS Filters. ApJS, 119:121–140.

Lindley, D. V. (1956). On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, 27(4):986–1005.

Loh, E. D. and Spillar, E. J. (1986). Photometric redshifts of galaxies. *Astrophysical Journal*, 303:154–161.

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

Ma, Z., Hu, W., and Huterer, D. (2006). Effects of Photometric Redshift Uncertainties on Weak-Lensing Tomography. ApJ, 636:21–29.

Masters, D., Capak, P., Stern, D., Ilbert, O., Salvato, M., Schmidt, S., Longo, G., Rhodes, J., Paltani, S., Mobasher, B., Hoekstra, H., Hildebrandt, H., Coupon, J., Steinhardt, C., Speagle, J., Faisst, A., Kalinich, A., Brodwin, M., Brescia, M., and Cavuoti, S. (2015). Mapping the Galaxy Color-Redshift Relation: Optimal Photometric Redshift Calibration Strategies for Cosmology Surveys. ApJ, 813(1):53.

Masters, D. C., Stern, D. K., Cohen, J. G., Capak, P. L., Stanford, S. A., Hernitschek, N., Galametz, A., Davidzon, I., Rhodes, J. D., and Sand ers, D. (2019). The Complete Calibration of the ColorRedshift Relation (C3R2) Survey: Analysis and Data Release 2. ApJ, 877(2):81.

Naul, B., Bloom, J. S., Pérez, F., and van der Walt, S. (2018). A recurrent neural network for classification of unevenly sampled variable stars. *Nature Astronomy*, 2:151–155.

Newman, J. A., Abate, A., Abdalla, F. B., Allam, S., Allen, S. W., Ansari, R., Bailey, S., Barkhouse, W. A., Beers, T. C., Blanton, M. R., Brodwin, M., Brownstein, J. R., Brunner, R. J., Carrasco Kind, M., Cervantes-Cota, J. L., Cheu, E., Chisari, N. E., Colless, M., Comparat, J., Coupon, J., Cunha, C. E., de la Macorra, A., Dell'Antonio, I. P., Frye, B. L., Gawiser, E. J., Gehrels, N., Grady, K., Hagen, A., Hall, P. B., Hearin, A. P., Hildebrandt, H., Hirata, C. M., Ho, S., Honscheid, K., Huterer, D., Ivezić, Ž., Kneib, J.-P., Kruk, J. W., Lahav, O., Mandelbaum, R., Marshall, J. L., Matthews, D. J., Ménard, B., Miquel, R., Moniez, M., Moos, H. W., Moustakas, J., Myers, A. D., Papovich, C., Peacock, J. A., Park, C., Rahman, M., Rhodes, J., Ricol, J.-S., Sadeh, I., Slozar, A., Schmidt, S. J., Stern, D. K., Anthony Tyson, J., von der Linden, A., Wechsler, R. H.,

Wood-Vasey, W. M., and Zentner, A. R. (2015). Spectroscopic needs for imaging dark energy experiments. *Astroparticle Physics*, 63:81–100.

Ormos, M. and Zibriczky, D. (2015). Entropy-Based Financial Asset Pricing. *arXiv e-prints*, page arXiv:1501.01155.

Padmanabhan, N., Schlegel, D. J., Seljak, U., Makarov, A., Bahcall, N. A., Blanton, M. R., Brinkmann, J., Eisenstein, D. J., Finkbeiner, D. P., Gunn, J. E., Hogg, D. W., Ivezić, Ž., Knapp, G. R., Loveday, J., Lupton, R. H., Nichol, R. C., Schneider, D. P., Strauss, M. A., Tegmark, M., and York, D. G. (2007). The clustering of luminous red galaxies in the Sloan Digital Sky Survey imaging data. MNRAS, 378:852–872.

Pasquet, J., Bertin, E., Treyer, M., Arnouts, S., and Fouchez, D. (2019). Photometric redshifts from sdss images using a convolutional neural network. *A&A*, 621:A26.

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

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Müller, A., Nothman, J., Louppe, G., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, É. (2012). Scikit-learn: Machine Learning in Python. *ArXiv e-prints*.

Peters, C. M., Richards, G. T., Myers, A. D., Strauss, M. A., Schmidt, K. B., Ivezić, Ž., Ross, N. P., MacLeod, C. L., and Riegel, R. (2015). Quasar Classification Using Color and Variability. ApJ, 811(2):95.

Pogosian, L., Corasaniti, P. S., Stephan-Otto, C., Crittenden, R., and Nichol, R. (2005). Tracking dark energy with the integrated Sachs-Wolfe effect: Short and long-term predictions. Phys. Rev. D, 72(10):103519.

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

Reed, I. S., Gagliardi, R. M., and Stotts, L. B. (1988). Optical moving target detection with 3-d matched filtering. *IEEE Transactions on Aerospace and Electronic Systems*, 24(4):327–336.

Revsbech, E. A., Trotta, R., and van Dyk, D. A. (2018). STACCATO: a novel solution to supernova photometric classification with biased training sets. MNRAS, 473:3969–3986.

Rhodes, J., Nichol, R. C., Aubourg, É., Bean, R., Boutigny, D., Bremer, M. N., Capak, P., Cardone, V., Carry, B., Conselice, C. J., Connolly, A. J., Cuilland re, J.-C., Hatch, N. A., Helou, G., Hemmati, S., Hildebrandt, H., Hložek, R., Jones, L., Kahn, S., Kiessling, A., Kitching, T., Lupton, R., Mand elbaum, R., Markovic, K., Marshall, P., Massey, R., Maughan, B. J., Melchior, P., Mellier, Y., Newman, J. A., Robertson, B., Sauvage, M., Schrabback, T., Smith, G. P., Strauss, M. A., Taylor, A., and Von Der Linden, A. (2017). Scientific Synergy between LSST and Euclid. ApJS, 233(2):21.

Ribeiro, M., Singh, S., and Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *arXiv e-prints*, page arXiv:1602.04938.

Rodríguez, A. C., Kacprzak, T., Lucchi, A., Amara, A., Sgier, R., Fluri, J., Hofmann, T., and Réfrégier, A. (2018). Fast cosmic web simulations with generative adversarial networks. *Computational Astrophysics and Cosmology*, 5:4.

Rozovskii, B. L. and Petrov, A. (1999). Optimal nonlinear filtering for track-before-detect in IR image sequences. In Drummond, O. E., editor, *Signal and Data Processing of Small Targets 1999*, volume 3809 of Proc. SPIE, pages 152–163.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323:533–536.

Sadeh, I., Abdalla, F. B., and Lahav, O. (2016). ANNz2: Photometric Redshift and Probability Distribution Function Estimation using Machine Learning. *Publications of the Astronomical Society of the Pacific*, 128(968):104502.

Sawicki, M. J., Lin, H., and Yee, H. K. C. (1997). Evolution of the Galaxy Population Based on Photometric Redshifts in the Hubble Deep Field. AJ, 113:1–12.

Schawinski, K., Zhang, C., Zhang, H., Fowler, L., and Santhanam, G. K. (2017). Generative adversarial networks recover features in astrophysical images of galaxies beyond the deconvolution limit. MNRAS, 467:L110–L114.

Seehars, S., Amara, A., Refregier, A., Paranjape, A., and Akeret, J. (2014). Information gains from cosmic microwave background experiments. *Physical Review D*, 90(2):023533.

Seo, H.-J. and Eisenstein, D. J. (2003). Probing dark energy with baryonic acoustic oscillations from future large galaxy redshift surveys. *The Astrophysical Journal*, 598(2):720.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell system technical journal*, 27:379–423.

Shao, M., Nemati, B., Zhai, C., Turyshev, S. G., Sandhu, J., Hallinan, G., and Harding, L. K. (2014). Finding Very Small Near-Earth Asteroids using Synthetic Tracking. ApJ, 782:1.

Slipher, V. M. (1913). The radial velocity of the Andromeda Nebula. *Lowell Observatory Bulletin*, 2:56–57.

Slipher, V. M. (1915). Spectrographic Observations of Nebulae. *Popular Astronomy*, 23:21–24.

Springel, V., White, S. D. M., Jenkins, A., Frenk, C. S., Yoshida, N., Gao, L., Navarro, J., Thacker, R., Croton, D., Helly, J., Peacock, J. A., Cole, S., Thomas, P., Couchman, H., Evrard, A., Colberg, J., and Pearce, F. (2005). Simulations of the formation, evolution and clustering of galaxies and quasars. Nature, 435:629–636.

Stark, D., Launet, B., Schawinski, K., Zhang, C., Koss, M., Turp, M. D., Sartori, L. F.,

Zhang, H., Chen, Y., and Weigel, A. K. (2018). PSFGAN: a generative adversarial network system for separating quasar point sources and host galaxy light. MNRAS, 477:2513–2527.

Szalay, A. S., Connolly, A. J., and Szokoly, G. P. (1999). Simultaneous Multicolor Detection of Faint Galaxies in the Hubble Deep Field. AJ, 117:68–74.

van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., Yu, T., and the scikit-image contributors (2014). scikit-image: image processing in Python. *PeerJ*, 2:e453.

Virtanen, P., Gommers, R., Oliphant, T. E., Cournapeau, D., Burovski, E., Weckesser, W., alexbrc, Peterson, P., wnbell, mattknox_ca, endolith, van der Walt, S., Laxalde, D., Brett, M., Millman, J., Lars, Mayorov, N., eric jones, Kern, R., Moore, E., GM, P., Schofield, E., Leslie, T., Perktold, J., cookedm, Griffith, B., Nelson, A., Eads, D., Vanderplas, J., Carey, C., Waite, T., Wilson, J., Escalante, A., Falck, R., fullung, Larson, E., Smith, D. B., Harris, C., Archibald, A., Molden, S., Cimrman, R., Henriksen, I., Hilboll, A., Berkenkamp, F., Feng, Y., Burns, C., Taylor, J., Schnell, I., Tsai, R., Nothman, J., Reimer, J., Quintero, E., Nowaczyk, N., Reddy, T., Taylor, J., prabhu, Stevenson, J., Seabold, S., Hochberg, T., Pedregosa, F., Teichmann, M., Bourquin, R., McIntyre, A., Warde-Farley, D., Ingold, G.-L., Kroshko, D., Varilly, P., Gohlke, C., Young, G., Probst, I., Nation, P., Fulton, C., Perez, F., Kulick, J., Vankerschaver, J., Kerr, C., fred.mailhot, Nandana, M., Scopatz, A., Vaught, T., jtravs, van Foreest, N., Robitaille, T., Lee, A., Venthur, B., Boulogne, F., Brodtkorb, P., Bunch, P., Wettinger, R., Grigorievskiy, A., Gaul, A., Silterra, J., chanley, and weinbe58 (2016). scipy/scipy: Scipy 0.18.1.

Way, M. J., Foster, L. V., Gazis, P. R., and Srivastava, A. N. (2009). New Approaches to Photometric Redshift Prediction Via Gaussian Process Regression in the Sloan Digital Sky Survey. ApJ, 706:623–636.

Weir, N., Fayyad, U. M., and Djorgovski, S. (1995). Automated Star/Galaxy Classification for Digitized Poss-II. AJ, 109:2401.

Wong, S. C., Gatt, A., Stamatescu, V., and McDonnell, M. D. (2016). Understanding data augmentation for classification: when to warp? *arXiv e-prints*, page arXiv:1609.08764.

Yip, C. W., Connolly, A. J., Vanden Berk, D. E., Ma, Z., Frieman, J. A., SubbaRao, M., Szalay, A. S., Richards, G. T., Hall, P. B., Schneider, D. P., Hopkins, A. M., Trump, J., and Brinkmann, J. (2004). Spectral Classification of Quasars in the Sloan Digital Sky Survey: Eigenspectra, Redshift, and Luminosity Effects. AJ, 128:2603–2630.

Zingales, T. and Waldmann, I. P. (2018). ExoGAN: Retrieving Exoplanetary Atmospheres Using Deep Convolutional Generative Adversarial Networks. *arXiv.org*, (6):268.

Appendix A

FAST ALGORITHMS FOR SLOW MOVING ASTEROIDS: CONSTRAINTS ON THE DISTRIBUTION OF KUIPER BELT OBJECTS

A.1 Introduction

Traditional approaches for detecting Trans Neptunian Objects (TNOs) rely on the identification of sources within individual images and then linking these sources to generate orbits (Kubica et al. 2007; Denneau et al. 2013). More recently digital tracking or "shift-and-stack" techniques have been developed to search for moving sources below the detection limit of any individual image (Gladman and Kavelaars 1997; Allen et al. 2001; Bernstein et al. 2004; Heinze et al. 2015). These approaches are fundamentally different from the traditional techniques in that they assume a trajectory for an asteroid and align a set of individual images along that trajectory in order to look for evidence for a source.

Shift-and-stack methods share many commonalities with the "track before detect" (TBD) method used for the tracking of satellites and missiles (e.g. Reed et al. 1988; Johnston and Krishnamurthy 2002). This field is mature in literature and implementation, and has been generalized to enable the detection of not just linear motion, but also the tracking of "acutely maneuvering non–cooperative targets" (Rozovskii and Petrov 1999). We will adopt several of the features of TBD, in particular those described in Johnston and Krishnamurthy (2002), who outline the core principles of accumulating the track detection probability, and in quantifying the false alarm probability.

A related approach to faint moving object detection is presented in Lang et al. (2009), who describe a search for high proper motion stars. These objects move by a distance comparable to the PSF FWHM over the course of an entire survey (i.e. years). Thus a

direct image stack is sufficient to detect objects. However, Lang et al. (2009) return to the individual science images to perform a joint fit for the proper motion and parallax of the objects, even though they appear at low signal–to–noise in any individual image. The scaling of detection depth in these techniques goes formally as $\Delta depth = 1.26\ \log(N)$ magnitudes where N is the number of images. This results in additional depth of 1 magnitude after the linking of 6 faint detections, and 2 magnitudes after 40 detections.

The advantage of digital tracking is that we increase the detection limit for a series of N images as $\sqrt{N}$ (assuming a constant point spread function (PSF) and background across all images). For objects having power–law distributions in apparent magnitude – e.g. the double power–law TNO model of Bernstein et al. (2004) – linking 6 epochs of data would yield an increase of 4–7 times as many objects *from the same data*. The disadvantage of digital tracking comes from the combinatorial and computational complexity of having to search a large number of candidate trajectories for each pixel within an image. Digital tracking must combine the individual images along a proposed motion vector that will depend on the assumed distance of the asteroid. Even for slow moving asteroids the number of searches will scale as Nn where n is the number of pixels in an image.

These computational costs have limited the application of digital tracking to searches for slowly moving objects or to narrow "pencil-beam" surveys. In this paper we introduce a new approach that utilizes a probabilistic formalism for the detection of sources in images (removing the need to stack or coadd images) and Graphics Processing Units (GPUs) to massively parallelize the number of searches that can be undertaken concurrently. In §2 we introduce a maximum-likelihood formalism for the detection of sources and extend this for the case of moving objects. In §3 we apply this approach to the High Cadence Transient Survey (HiTS) and describe some of the filtering techniques that were applied to exclude false positives in the candidate asteroids. In §4 we discuss the asteroids detected by this approach and compare their properties to current models and observations.

A.2 Fast tracking and stacking of images

Digital tracking assumes a set of N images have been observed over a period of time (from minutes to days) with the individual images covering approximately the same part of the sky. The individual images are astrometrically shifted along a proposed motion vector, coadded, and then searched for point sources in the resulting coadds (Gladman and Kavelaars 1997; Allen et al. 2001; Bernstein et al. 2004). This approach is illustrated in Figure A.1 where the image on the far right is the sum of the previous three images added along the motion of the highlighted object. Creating a stack optimized for faint, moving objects must include astrometric offsets between the images that correspond to the distance that an object has moved between observations. Unfortunately, this angular velocity is not known *a priori*, and in fact differs for each moving object in the field. Thus a stacked search for Solar System objects in time–series data requires a sequence of coadds, each optimized for a particular motion vector. Due to the combinatorial complexity of the problem, these efforts have traditionally been optimized for TNO recovery, where apparent angular motions are small.

During an observation of a Solar System object, there are apparent motion contributions from the object's own space velocity, as well as from the reflex motion of the Earth, which is primarily due to the Earth's motion around the Sun but also includes the Earth's rotation around its axis. These contribute to an apparent angular motion of the Solar System object, which will trail during an exposure. If the object trails by more than the PSF full-width half-maximum (FWHM) during observation, its signal is spread over additional background pixels, which lessens the overall signal-to-noise (Shao et al. 2014).

These "trailing losses" also apply to a stacked image: the image stack velocity must be close enough to the true velocity of the object to not spread the signal over more than the PSF FWHM. This requirement, together with the range of expected apparent motions of the desired objects, sets the number or sampling of the velocities (or orbits) that must be searched and stacks that must be examined. Typical apparent motions (at opposition) range from $20''$/hr for main belt asteroids at 3 AU to $1''$/hr for TNOs.

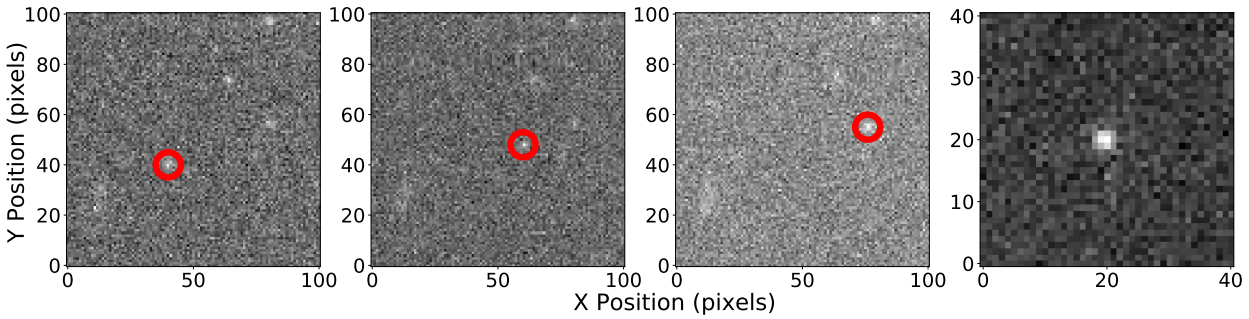

Figure A.1: Shifting and stacking of individual images along the asteroid's trajectory creates a single point source in the stacked image.

A.2.1 A likelihood-based approach for source detection

Form of single pixel likelihood function

Our goal is to derive the likelihood functions for a given source being present in the coaddition of a series of images. First we start by finding the form of the likelihood for a single pixel in a single image. The following derivation is based upon work in Kaiser (2004) and interpreted in Bosch (2015) and Bosch et al. (2018). Photons landing on a pixel in a detector follow Poisson statistics. For a given pixel the probability of counting n photons with an expected value of μ is defined as

$$P(n|\mu) = \frac{\mu^n e^{-\mu}}{n!}. \tag{A.1}$$

This probability can also be interpreted as the likelihood that a model of the sources within an image (hereafter the true image) generates a predicted count μ given we observe n counts on a pixel. The log-likelihood of our prediction model is, therefore,

$$\mathcal{L}(\text{model}) = \ln P(\text{data}|\text{model}) = \ln P(n|\mu) = n \ln \mu - \mu - \ln n! \tag{A.2}$$

Differentiating $\mathcal{L}$ with respect to μ, the log-likelihood has its peak when $\mu_o = n$. Furthermore, if we Taylor expand around this maximum value then we get

$$\mathcal{L}(\text{model}) = n \ln \mu_o - \mu_o - \ln n! + (\frac{n}{\mu_o} - 1)(\mu - \mu_o) - \frac{n}{2\mu_o^2}(\mu - \mu_o)^2 + \dots \tag{A.3}$$

$$\mathcal{L}(\text{model}) = \text{constant} - \frac{1}{2}\frac{(n-\mu)^2}{n} + \dots \tag{A.4}$$

where the constant contains all of the terms that depend only on n and so are independent of our model. Finally, if n is large we can ignore higher order terms and approximate our likelihood function as that of a Gaussian with likelihood proportional to $e^{-\frac{(n-\mu)^2}{2n}}$.

Pixels and PSF

At this point we need to take a step back and understand what exactly goes into calculating photon counts at each pixel. To do this we will follow the derivation laid out in Bosch (2015) but for a two-dimensional image. First, we start with the number of counts $n(x, y)$ we get from the true sky intensity $I(x, y)$ on a pixel centered at (x, y). The observation on our detector will be the true sky intensity convolved with the PSF, $T(x, y)$. In addition, there will be an extra integral to account for the binning into a pixel centered at (x, y) with side length a. That gives us

$$n(x,y) = \int_{x-a/2}^{x+a/2}\int_{y-a/2}^{y+a/2} dv\ du \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} dq\ dp\ I(p,q)\ T(u-p, v-q), \tag{A.5}$$

which can be rewritten in terms of two convolutions where we rewrite the pixel binning integral as a convolution with a square top hat function $H(x, y)$ with height 1 and side length a

$$n(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} dv\ du\ H(x-u, y-v) \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} dq\ dp\ I(p,q)\ T(u-p, v-q) \tag{A.6}$$

$$n(x,y) = [H * I * T](x,y) \tag{A.7}$$

and if we exploit the associative and commutative properties of convolutions we can rewrite this as

$$n(x,y) = [I * (H * T)](x,y) \tag{A.8}$$

and following the procedure in Bosch (2015) simplifying the term in parentheses to

$$T_a(x,y) = (H * T)(x,y) \tag{A.9}$$

and finally ending up with

$$n(x,y) = [I * T_a](x,y) \tag{A.10}$$

so that when we refer to the PSF below as T_a we are referring to the PSF including the pixel transfer function.

Full likelihood function for a single image

Now we understand how to represent the pixel data as a function of the sky and PSF as well as how to write out the likelihood function of a single pixel. We consider an image $\mathbf{x}$, where the ith pixel is $\mathbf{x}_i$ and the real sky is modeled as $f(\mathbf{x})$. Then we use our assumption of a large photon count to justify a Gaussian likelihood function as described in A.2.1 and get

$$L(\mathbf{x}_i) = P(\text{data}|\text{model}) = P(n(\mathbf{x}_i)|f(\mathbf{x}_i)) = \frac{1}{\sqrt{2\pi\sigma^2_{\mathbf{x}_i}}} e^{\frac{-\frac{1}{2}(n(\mathbf{x}_i)-[f*T_a](\mathbf{x}_i))^2}{\sigma^2_{\mathbf{x}_i}}} \tag{A.11}$$

but if we are looking at the entire image plane we need to go from a single variable Gaussian to a multivariate Gaussian distribution. Thus, the likelihood for the full image is

$$L(\mathbf{x}) = P(n(\mathbf{x})|f(\mathbf{x})) = \frac{1}{\sqrt{(2\pi)^k|C|}} e^{-\frac{1}{2}\sum_{i,j}(n(\mathbf{x}_i)-[f*T_a](\mathbf{x}_i))\times C^{-1}(\mathbf{x}_i,\mathbf{x}_j)\times(n(\mathbf{x}_j)-[f*T_a](\mathbf{x}_j))} \tag{A.12}$$

where k is the number of pixels and C is the pixel covariance matrix. If we take the log-likelihood function then we have

$$\mathcal{L}(\mathbf{x}) = -\frac{k}{2}\ln(2\pi) - \frac{1}{2}\ln|C| - \frac{1}{2}\sum_{i,j}(n(\mathbf{x}_i) - [f*T_a](\mathbf{x}_i)) \times C^{-1}(\mathbf{x}_i,\mathbf{x}_j) \times (n(\mathbf{x}_j) - [f*T_a](\mathbf{x}_j)) \tag{A.13}$$

for the full form of the log-likelihood of a given sky model $f(\mathbf{x})$.

Likelihood of detection of a point source

If we want to find a point source at a pixel $\mathbf{y}$ in the image with flux α then the sky model we are proposing is $f(\mathbf{x}) = \alpha\delta(\mathbf{x}-\mathbf{y})$, a delta function located at the pixel lcoation of the source multiplied by the flux. Now let $n(\mathbf{x})$ represent the flux counts for each pixel we have

in a background subtracted image and notice that the first two terms in equation A.13 are independent of the model $f(\mathbf{x})$. Furthermore, in order to keep values positive, we change to the negative log-likelihood function which we are now trying to minimize in order to get the most likely parameters. Thus, we get

$$\mathcal{L}(\alpha, \mathbf{y}) = \text{constant} + \frac{1}{2} \sum_{i,j} (n(\mathbf{x}_i) - \alpha T(\mathbf{x}_i - \mathbf{y})) \times C^{-1}(\mathbf{x}_i, \mathbf{x}_j) \times (n(\mathbf{x}_j) - \alpha T(\mathbf{x}_j - \mathbf{y})) \quad \text{(A.14)}$$

where the constant holds terms that are model independent. Also, we have defined $\alpha T(\mathbf{x}_i - \mathbf{y}) = [f * T_a](\mathbf{x}_i)$ which is the convolution of the point source sky model with the PSF and is thus equivalent to the PSF centered at $\mathbf{y}$. $T(\mathbf{x}_j - \mathbf{y}))$ is the same for the pixels in the j indexed sum. We have two sums here since we are summing across every combination of pixels from the covariance matrix. If we multiply out these terms it looks like the following:

$$\mathcal{L}(\alpha, \mathbf{y}) = \text{constant} + \frac{1}{2} \sum_{i,j} C^{-1}(\mathbf{x}_i, \mathbf{x}_j)(n(\mathbf{x}_i) n(\mathbf{x}_j)) - \alpha \sum_{i,j} C^{-1}(\mathbf{x}_i, \mathbf{x}_j) n(\mathbf{x}_i) T(\mathbf{x}_j - \mathbf{y}) + \frac{\alpha^2}{2} \sum_{i,j} C^{-1}(\mathbf{x}_i, \mathbf{x}_j) T(\mathbf{x}_i - \mathbf{y}) T(\mathbf{x}_j - \mathbf{y}) \quad \text{(A.15)}$$

Furthermore we define the following terms

$$\Psi(\mathbf{y}) = \sum_{i,j} C^{-1}(\mathbf{x}_i, \mathbf{x}_j) n(\mathbf{x}_i) T(\mathbf{x}_j - \mathbf{y}) \quad \text{(A.16)}$$

$$\Phi(\mathbf{y}) = \sum_{i,j} C^{-1}(\mathbf{x}_i, \mathbf{x}_j) T(\mathbf{x}_i - \mathbf{y}) T(\mathbf{x}_j - \mathbf{y}) \quad \text{(A.17)}$$

and once again add into constant the terms that are not model dependent. Finally, this gives us the likelihood in a compact form

$$\mathcal{L}(\alpha, \mathbf{y}) = \text{constant} - \alpha \Psi(\mathbf{y}) + \frac{\alpha^2}{2} \Phi(\mathbf{y}) \quad \text{(A.18)}$$

where $\Psi(\mathbf{y})$ and $\Phi(\mathbf{y})$ are new types of images that can be created using the PSF.

Coaddition of likelihood images for point source detection

We make the assumption that the signal for the majority of candidate detections will be dominated by the background noise and that the background noise is independent in each

pixel. While there may be sources of correlation between pixels such as electronic detector effects (e.g. crosstalk or the dependence of the PSF on intensity), for this treatment we will assume these effects are small enough to ignore and thus will continue as if we have a completely diagonal covariance matrix. This simplifies our calculations into images that can be precomputed with a simple kernel that approximates the PSF.

Thus, our equations for Ψ and Φ images are reduced to

$$\Psi(\mathbf{y}) = \sum_i \frac{1}{\sigma_i^2}\, n(\mathbf{x}_i) T(\mathbf{x}_i - \mathbf{y}) \tag{A.19}$$

$$\Phi(\mathbf{y}) = \sum_i \frac{1}{\sigma_i^2}\, T(\mathbf{x}_i - \mathbf{y})^2 \tag{A.20}$$

where Ψ is the inverse-variance weighted cross-correlation of the PSF and the data, which is also the convolution of the PSF rotated by 180 degrees or just the PSF if it is symmetric. Φ is the effective area of the PSF weighted by the inverse variance (Bosch et al. 2018).

Bright pixels in the image can be dealt with by a mask and we set the inverse variance to zero so that they will add nothing to the likelihood sum. Since trajectories will only cross over this pixel once when we run them over a series of images this serves to give noisy pixels zero weight in the full sum of a trajectory while keeping the information in the other images.

Solving for the maximum-likelihood solution we end up with

$$\frac{\partial \mathcal{L}_{ML}}{\partial \alpha} = -\Psi(\mathbf{y}) + \alpha \Phi(\mathbf{y}) = 0 \tag{A.21}$$

and as a result we find that,

$$\alpha_{ML} = \frac{\Psi(\mathbf{y})}{\Phi(\mathbf{y})} \tag{A.22}$$

and

$$\mathcal{L}_{ML}(\alpha_{ML}, \mathbf{y}) = \text{constant} - \frac{\Psi^2(\mathbf{y})}{2\Phi(\mathbf{y})} \tag{A.23}$$

where α_{ML} is the most likely flux for a source at pixel x_i and $\mathcal{L}_{ML}(\alpha_{ML}, \mathbf{y})$ is the probability of that source given the observation $n(x_i)$. The kernel that maximizes the likelihood of our

flux measurements is, therefore, the 180 degree rotation of the PSF or, in the case of a symmetric PSF, the PSF itself. Additionally, Equation A.23 is the log form of a Gaussian—where by analogy $\Psi \simeq (x - \mu)$ and $\Phi \simeq \sigma^2$—so we can approximate a χ^2 distribution. If we define $\nu = \frac{\Psi}{\sqrt{\Phi}}$ and then make an image of ν, we can call points above some threshold value m to be m-sigma detections (Szalay et al. 1999).

In order to make a coadded likelihood image, we create our Ψ and Φ images from sums across all pixels in all images. This amounts to making Ψ and Φ likelihood images for each of our original images, i, with the appropriate PSF for each respective image and summing them separately so that

$$\Psi_{coadd} = \sum_i \Psi_i(\mathbf{y}_i) \tag{A.24}$$

$$\Phi_{coadd} = \sum_i \Phi_i(\mathbf{y}_i) \tag{A.25}$$

$$\nu_{coadd} = \frac{\Psi_{coadd}}{\sqrt{\Phi_{coadd}}} \tag{A.26}$$

Here, ν is the signal-to-noise of the detection of a source at pixel x within the coadded image. Therefore, the final value for the likelihood of a point object is just the sum of the values at a given set of points in the Ψ image divided by the sum of the values at the same points in the Φ image. This means that if we wish to do a moving object detection, all that is needed is to sum the Ψ and Φ values as shown, but we must use the appropriate pixel coordinates for a given trajectory as the $\mathbf{y}_i$ values in each image—there is no need to shift and stack the likelihood images at any point. The Ψ and Φ images can also be precalculated meaning we are able to store them in memory as many trajectories are searched.

A.2.2 Object detection as Optimization

Detecting moving point sources in a stack of images can now be approached as an optimization problem. For the linear trajectories we are considering, the core task can be reduced to finding the initial positions $\mathbf{y}_0$ and velocities $\mathbf{v}$ so that ν is maximized. For a given potential

object, the best candidate source is then

$$\boldsymbol{argmax}_{\mathbf{y}_0,\mathbf{v}}(\nu_{coadd}) \tag{A.27}$$

We hope to find all unique $\mathbf{y}_0$ and $\mathbf{v}$ such that $\nu(\mathbf{y}_0, \mathbf{v})$ is above a desired detection threshold. The most straightforward computational approach to this problem is to evaluate ν for all possible values of $\mathbf{y}_0$ and $\mathbf{v}$ that describe realistic orbits. An illustration of this approach is shown in Figure A.2 starting from a single value of $\mathbf{y}_0$. This method directly computes Ψ and Φ for every relevant trajectory through all images by sampling pixels along each trajectory and storing the result if the integrated ν is above the threshold. The computational complexity of this algorithm is bounded by $\mathcal{O}(na(tu)^2/p)$ where n is the number of images to be stacked, a is the area on the sky to be searched, p is the area of a single pixel, t is the duration between the first and last image, and u is the range of an object's apparent velocity. The complexity scales with time and velocity range quadratically because all trajectories ending anywhere inside an area with radius tu must be considered.

A.2.3 GPU Implementation

Comprehensively searching a stack of images for moving objects directly requires computing the value of ν billions of times. Fortunately each evaluation of ν is entirely independent from the others, and this allows for natural parallel execution. Rectangular patches of adjacent trajectories are grouped into thread blocks with dimensions 32x16 which are distributed across the GPU's multiprocessors. Ψ and Φ pixels are interleaved in memory and within each thread block horizontally adjacent trajectories access them contiguously, enabling high throughput. To achieve good performance all images must be stored in the GPU's memory at once, which limits the size and number of images that can be used to about 100 4K x 4K images. In general, the runtime of this algorithm can be estimated as Nnk where N is the number of images in the stack, n is the number of trajectories, and k is a performance constant that is experimentally determined by the computer hardware and implementation. In our implementation we have a measured value of k about 2.4×10^{-11} seconds per image

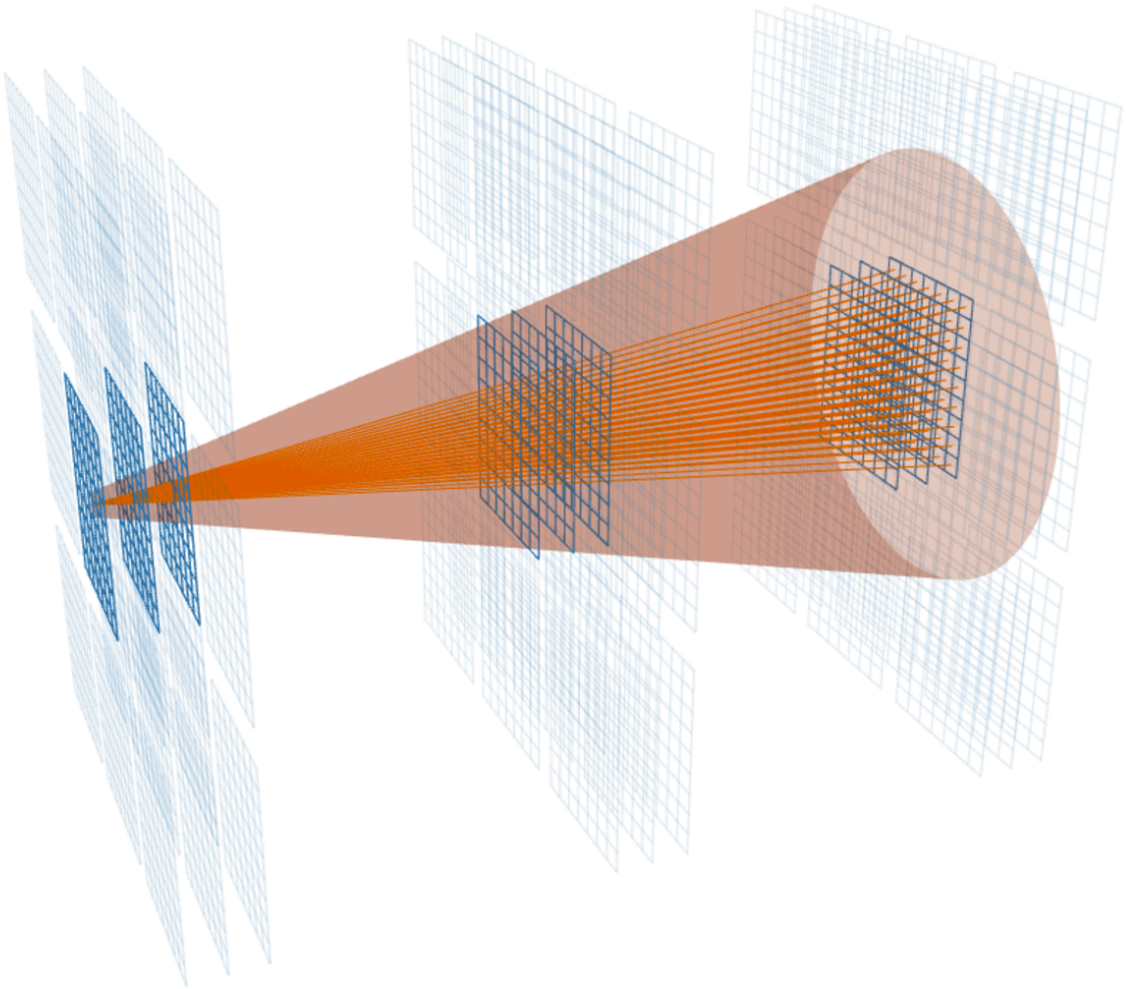

Figure A.2: Visualization of the many trajectories that must be searched in order to cover a defined velocity and angle range over a stack of images of the same field taken at different times.

for each trajectory. Practically this means searching for a wide range of objects (240 billion trajectories) in a stack of 30 images (4K x 4K) takes about 180 seconds. Besides computing ν, we also use the GPU for convolving the images. The convolution is done in a single pass with a spatially invariant kernel. In this work we only used a Gaussian PSF approximation, and could have used a two-pass separable convolution but chose to use a slower single pass to support non-Gaussian PSF's in the future.

A.3 Searching for faint KBOS

A.3.1 The High Cadence Transient Survey (HiTS)

We tested our software using the first three nights (February 17-19, 2015) of the 2015 campaign of the High Cadence Transient Survey (HiTS) (Förster et al. 2016). The 2015 HiTS campaign involved repeatedly visiting 50 3-square-degree fields in the DECam g-band filter. There were typically 5 visits per night to each of the 50 fields, taken with a 1.6hr cadence. The remaining three nights of the 2015 HiTS campaign data, also taken with the DECam g-band, were used for follow up of the detected objects (Förster et al. 2016).

These data were taken using the Dark Energy Camera (DECam) at Cerro Tololo Interamerican Observatory (CTIO). The best quartile of seeing at CTIO is about 0.4" full width at half maximum (FWHM). DECam has 60 2K x 4K CCDs, each with a pixel scale of 0.26 arcsec/pixel. We processed all of the data using the LSST Data Management (DM) Software (Jurić et al. 2015) and ran our software using the warped science images that were laid out in 4K x 4K pixel patches. This means that the effective field of view of each warped image is about 0.34 square degrees (Flaugher et al. 2015). Because we currently do not search trajectories across CCDs, this is the effective field of view of any individual search. We also used the masks and variance planes from the LSST DM processing output.

A.3.2 Application of the KBO search

We created Ψ and Φ images from the individual science images. To mask static objects we identify all sources that are detected at more than 5σ above the background and mask the pixels associated with these sources. Masks for each static source are grown by an additional two pixels (in radius) to exclude lower surface brightness halos around these sources. To ensure that we do not mask bright *moving* sources we require that any masked pixel in a science image must be masked in at least one of the other science images (i.e. a source must be present at the same position in two of the images for it to be defined as static and masked). We apply the union of the mask for all individual images as a global mask to the final science images.

Our search started at every pixel in the earliest observation (in time) for each HITS field and covered a range of 32,000 trajectories across the stacks of 12-13 images corresponding to three nights of HITS data. The 32,000 trajectories came from a linearly-spaced grid of 128 angular steps within $\pm 12^\circ$ of the ecliptic and 250 steps in velocities ranging from 1"–5.7"/hour. This grid was set up so that at the end of the search period, our maximum separation between the final pixels in a search pattern would be no more than approximately 2 PSF widths. This means that that we would be within 1 PSF width of any possible trajectory. For this search, we went down to a signal-to-noise ratio (SNR) threshold of 10, as measured by the ν parameter introduced in Section A.2.1. For these data, this corresponds to a single image limiting magnitude in g of 23.1.

A.3.3 Filtering of candidate trajectories

After running the initial GPU search, we use a set of Python tools to filter out false detections. One such tool is an outlier filtering process that modifies estimates of the flux and variance of an object given a sequence of observations. This is done using the light curve of the candidate trajectory and a variation on the Kalman filter (Kalman 1960) commonly used in signal processing. We filter out any observations that are more than five standard deviations

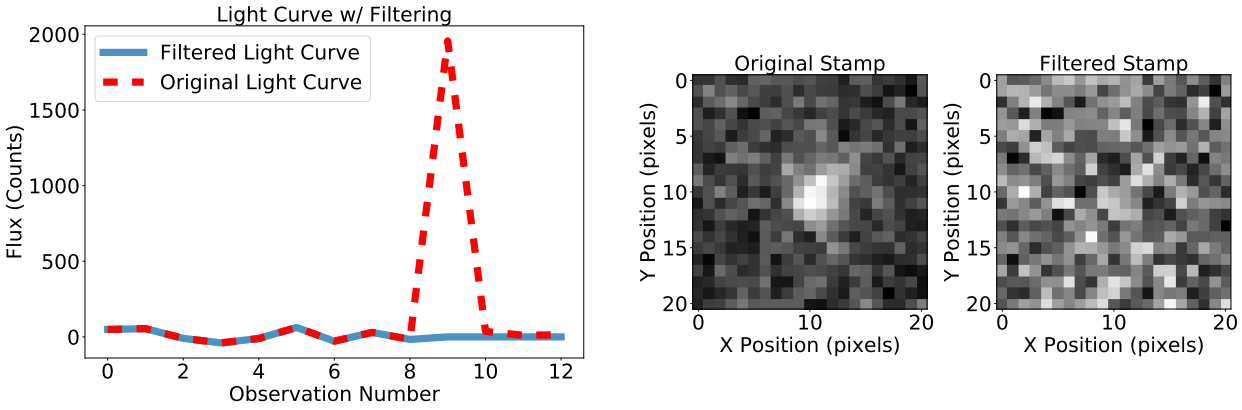

Figure A.3: Left: Change in light curve when an image with an outlier flux is removed from the light curve. Right: Shifted and stacked postage stamps before and after outlier removal. After the single outlier observation is removed by the filter the trajectory is obviously not following a true object and is discarded.

away from the best fit Kalman flux at that observation and use the remaining observations to recalculate the likelihood of the candidate. As an example of this process, Figure A.3 shows postage stamps of the candidate before and after the filtering. In this case a fast, bright, moving object moved across the trajectory in a single image. Excluding that image leads us to the correct decision to discard this candidate trajectory, as the likelihood of an object along this trajectory drops to near zero, once the interloping object is removed.

After the outlier detection we build coadded postage stamps for all remaining candidate trajectories. We then use scikit-image (van der Walt et al. 2014) to calculate the central moments of the images and filter based upon the similarity of the moments to a Gaussian centered at the middle of the stamp. This does a good job of eliminating elongated shapes and trajectories where a bright source appeared in a single image but was off center. For example, in Figure A.4 the left column shows postage stamps of real objects that passed through the filtering. The right column of the figure shows postage stamps that made it through the outlier filtering but were ruled out after the image moment filter.

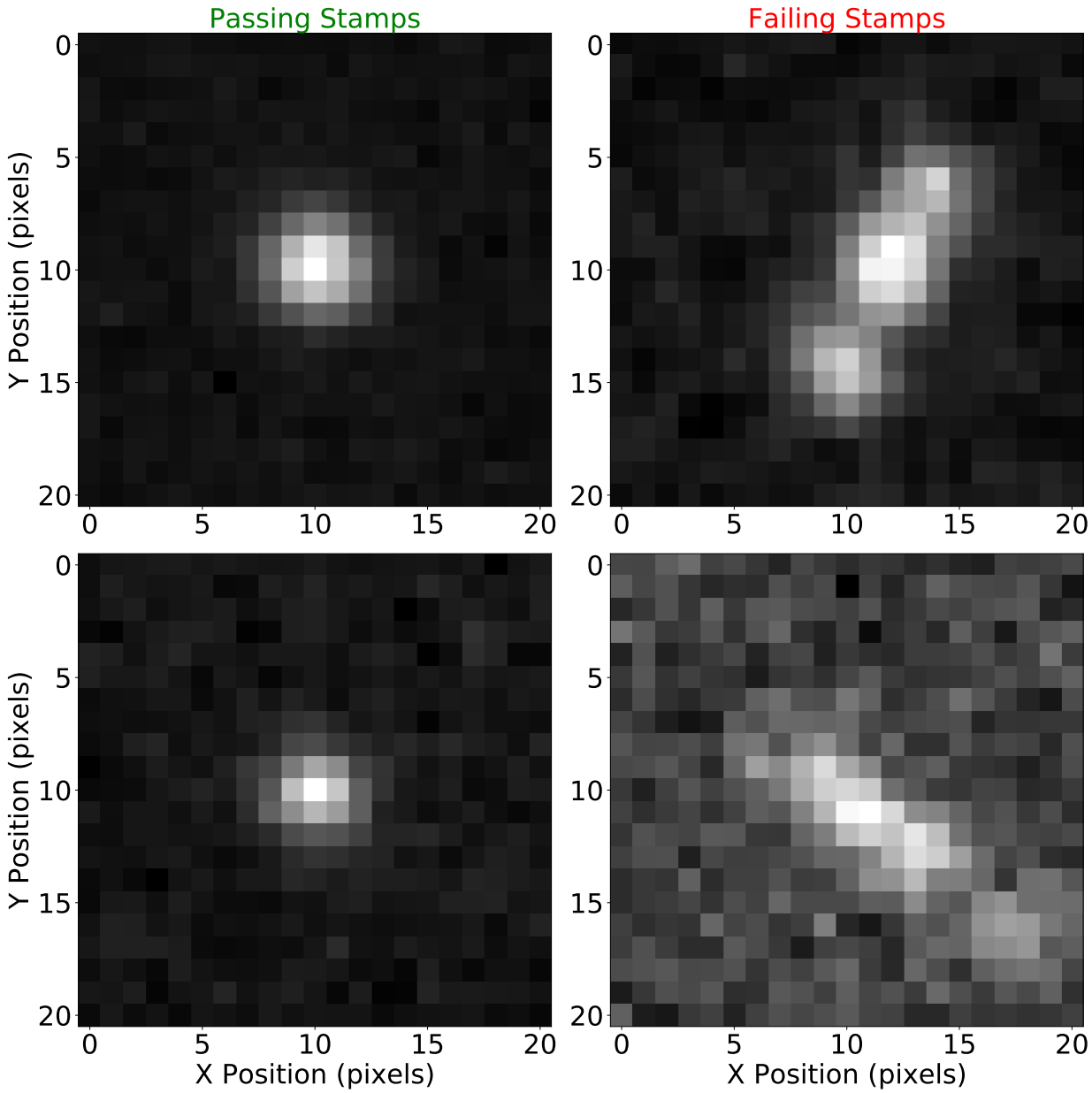

Figure A.4: Left Column: Stamps that passed the lightcurve filter and the image moment filter. Right Column: Stamps that passed the lightcurve filter but were rejected by the image moment filter.

Our final step in the filtering process is clustering to remove duplicate results from the same object. We take the starting x and y pixels and the horizontal and vertical velocities of the candidate trajectories and use the DBSCAN clustering method (Ester et al. 1996) in the scikit-learn python package (Pedregosa et al. 2012) to group similar trajectories. We then take the highest likelihood trajectory for each group and save the results with postage stamps and light curves to file for final examination by eye.

A.3.4 Found Objects

KBO properties in the HiTS field

In total we found 45 KBOs in our search, of which only 6 were previously detected by the Pan-STARRS 1 (PS1) survey according to the Minor Planet Center (MPC). PS1 used the Pan-STARRS Moving Object Processing System (MOPS) which links detections from sources identified in individual difference images (Denneau et al. 2013).

The full information for all our object detections is shown in Table A.1 at the end of the appendix. We used the orbit fitting code of Bernstein and Khushalani (2000) for initial orbit determination and used the remaining HITS data (see Section A.3.1) to get additional observations where possible before submitting to the MPC. From this we estimated that, for the 45 KBOs, semi-major axes ranged from 21 AU to 67 AU and DECam g-band magnitudes ranged from 22.1 to 24.7 mags. Figure A.5 compares semi-major axis to inclination for our discoveries and shows the overlap with KBO populations commonly discussed in the literature.

Comparison with known asteroids

The candidate objects were checked with the MPC database of known objects. Of the 45 objects that were detected, 6 were known. Four of these objects were detected in our search down to a SNR threshold of 10 and they are noted in Table A.1. The 2 remaining objects were much fainter at V magnitudes of 24.1 and 24.3 while the faintest MPC object we recovered in

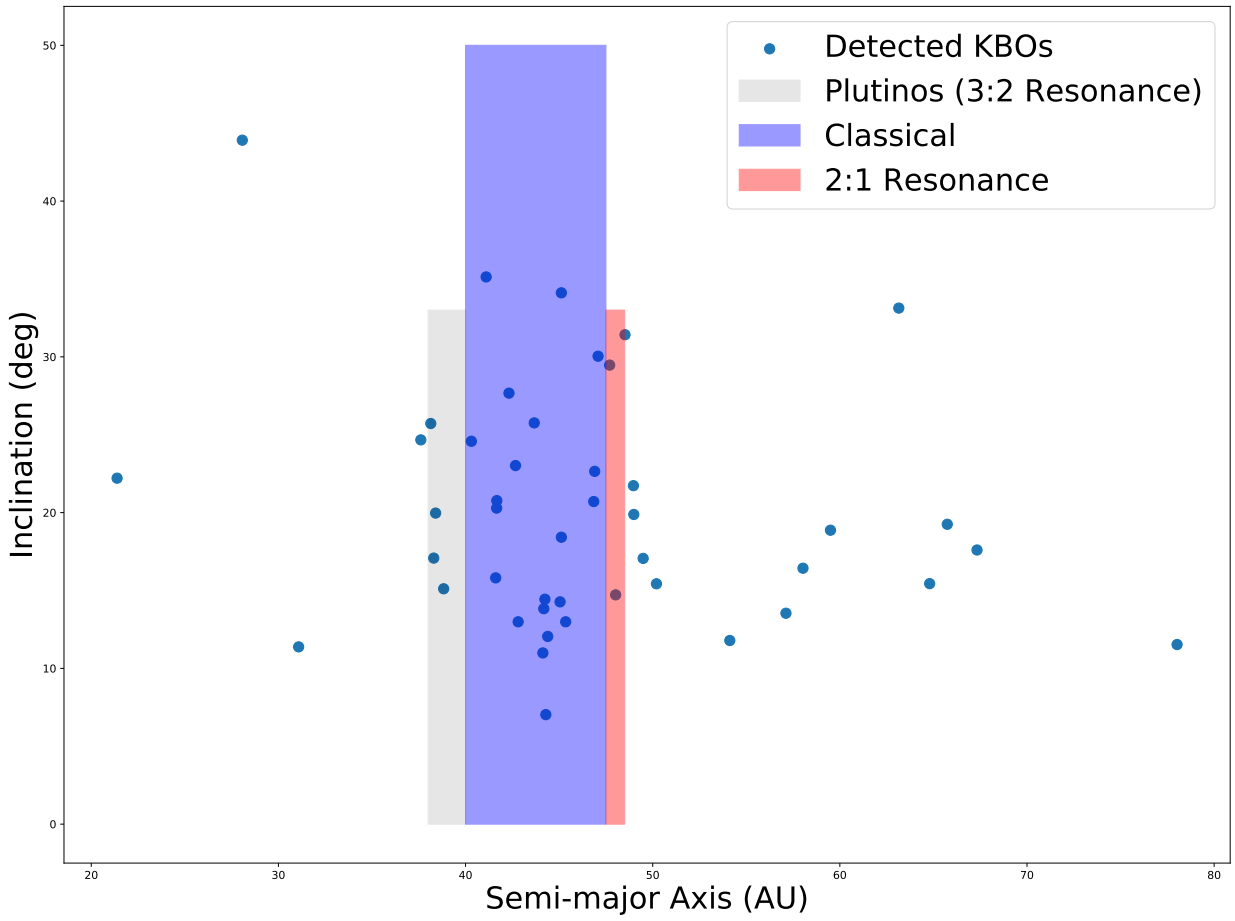

Figure A.5: Semi-major axis versus inclination for detected objects with the range of different KBO populations highlighted.

our search was at V = 23.0. We attempted to recover the objects at a fainter SNR threshold by rerunning the search on the image stacks corresponding to the predicted locations. One of the objects was recovered when we reduced our SNR threshold to 5.96. The final object ran from the edge of one image stack to another between the first and second night. Our code cannot currently account for this situation in our search. However, we ran the search using only the second and third nights of data to find the object and our code was able to find it at a reduced SNR value of 5.31. This means that when going to a faint enough search threshold we were able to recover all of the known KBOs in the search area.

After submission to the MPC database two more results were matched to objects in the MPC catalogs. Previous observations had not predicted the orbits to fall within the observation fields, but were linked and recalculated by the MPC after submission of our results. These new matches are also noted as previously discovered in Table A.1.

A.4 Results

A.4.1 Recovery Efficiency

To understand the efficiencies of our search method, we inserted simulated objects with g magnitudes between 20-26 into the images from one of the 2015 HITS fields and tested our ability to recover these objects. The objects all had a simple Gaussian PSF that matched our search PSF and the velocity angles and magnitudes were uniformly distributed within the ranges of our search parameters. Figure A.6 shows the results in terms of counts on the left and the fraction of the total simulated set on the right. Both plots are a function of DECam g magnitude with bin widths of 0.25 magnitudes. For the right plot we fit an efficiency function of the form $f(m) = f_0/e^{1+\frac{m-L}{w}}$, where f_0 is the efficiency ceiling, L is the 50% detection probability magnitude, and w is the width in magnitudes of the drop-off in sensitivity.

We plotted two efficiency functions, one for the entire range of magnitudes of the simulated objects and then one at magnitudes $g > 21$. The drop-off at the brighter magnitudes

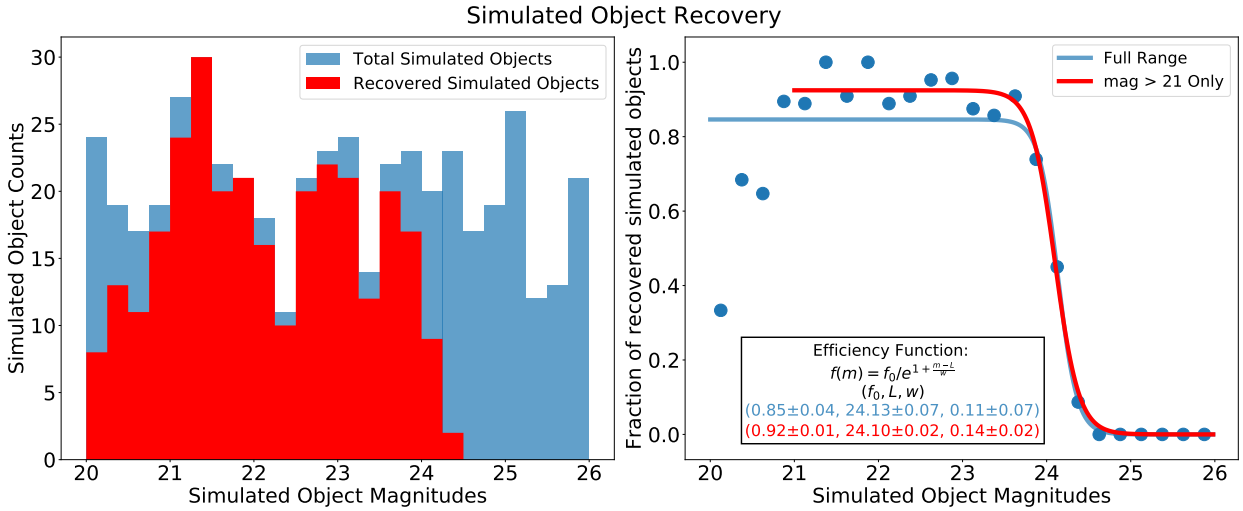

Figure A.6: Recovery of simulated objects inserted into a HITS field. Left: Histogram comparing counts of recovered simulated objects to the full set as a function of magnitude. Right: Fraction of recovered simulated objects as a function of magnitude fitted with an efficiency curve for both the full range of simulated objects and for $g > 21$ only.

is due to the extra masking we did at a specific count threshold. We will say more about this effect in the discussion in Section A.5. On the fainter end we see the limit set by the SNR cutoff at SNR = 10. The detection efficiency ceiling for the full range is $84.6 \pm 3.7\%$ with a 50% threshold at $g = 24.13 \pm .07$ and drop-off width of $0.11 \pm .07$ mags. When we look at efficiencies at $g > 21$ we get an efficiency ceiling of $92.5 \pm 1.2\%$ and a 50% detection threshold at the very similar $g = 24.10 \pm .02$ value and drop-off width of $0.14 \pm .02$ mags.

A.4.2 Comparison with Existing Models

While our original searches used only three nights of data, we had additional nights in the HITS data that we used for supplying additional observations when estimating the orbits of the discovered KBOs. This still limited our baseline for the discovered objects to a range from 3-10 days and made accurate estimates of semi-major axis and eccentricity difficult. We have confident measurements on the inclinations and all our inclination estimates for objects in the MPC catalog matched the published inclinations within 1 σ of the output values from the Bernstein and Khushalani (2000) orbit fitting code. Therefore, we present a comparison of our inclination distribution to the general KBO population such as that in Brown (2001).

Inclination Distribution

Our inclination values range from 7°-44° where the lower limit comes from the fact that our closest field to the ecliptic is at $-6.4°$. Even though our fields are all at moderate latitudes off the ecliptic (as far as $-21.3°$), Brown (2001) provides a method to compare our results to a predicted distribution by using the inclinations of the objects and the ecliptic latitude of discovery.

Brown (2001) estimates an inclination distribution for the full KBO population with a double Gaussian multiplied by $\sin i$:

$$f_t(i) = \sin i[a \exp(\frac{-i^2}{2\sigma_1^2}) + (1-a) \exp(\frac{-i^2}{2\sigma_2^2})] \tag{A.28}$$

where $a = 0.89$, $\sigma_1 = 2.7°$ and $\sigma_2 = 13.2°$. To compare our results to this distribution we follow the method outlined in Section 3 of Brown (2001). For a given inclination distribution, f_t, the probability that an object, j, with discovery latitude, β_j, would have an inclination equal to or below the actual inclination, i_j is given by Equation A.29.

$$P_j = \int_{\beta_j}^{i_j} \frac{f_t(i')}{(\sin^2 i' - \sin^2 \beta_j)^{1/2}} di' \times [\int_{\beta_j}^{\pi/2} \frac{f_t(i')}{(\sin^2 i' - \sin^2 \beta_j)^{1/2}} di'] \tag{A.29}$$

The distribution of P_j for the actual KBO population estimated by Brown (2001) varies uniformly between 0 and 1. Therefore, if we take as the null hypothesis for our observations that they are an unbiased sample down to our magnitude limits and representative of the distribution of KBO inclinations in the fields we searched we should compare the distributions of P_j for our objects to the uniform distribution. To do this we start by calculating P_j for each of our objects and plot their sorted distribution in Figure A.7. We compare our observed distribution to the inclination distribution of Brown (2001) using the Kuiper variant of the Kolmogorov-Smirnov (K-S) test as done in Brown (2001) and according to Equation A.30.

$$D = \max(P_j - j/N) \tag{A.30}$$

The actual test statistic is $D\sqrt{N}$ where N is the sample size which is $N = 45$ in our data set. In order to find the confidence levels for the test, we create 10^5 sets of N=45 random samples drawn from a uniform distribution and calculate $D\sqrt{N}$ comparing to a uniform distribution. The 1σ confidence value occurs when the probability of getting higher than a given $D\sqrt{N}$ value is 84.1%. We find this to be at $D\sqrt{N} = 1.47$.

Finally, we calculate the P_j values using the Monte Carlo methods described in Brown (2001). We first draw 10^5 inclinations from the Brown (2001) distribution and randomly place them along circular orbits. For each of our observed objects, j, we then take all of the Monte Carlo objects within $\pm 0.5°$ of the latitude of discovery, β_j and construct an empirical inclination distribution. In order to derive P_j for an object, we use this distribution to calculate the probability that an object with the given β_j will have an inclination at or below i_j. Using this set of P_j values we then perform the K-S test compared to a uniform

distribution between 0 and 1. We perform the Monte Carlo simulation 1000 times and use the mean $D\sqrt{N}$ value as our test statistic for comparison. The K-S test comparison for one of the Monte Carlo distributions is shown in Figure A.7. Our mean $D\sqrt{N}$ value after 1000 runs was 1.37, corresponding to a confidence level of 75% and within the 1σ level. This means that we cannot reject the hypothesis that our observations come from the Brown (2001) distribution and are consistent with this prediction.

As a further comparison we did a basic survey simulation to determine the expected distribution of objects we would find in the HITS data. We used the Monte Carlo distribution of inclinations and locations of objects scattered around circular orbits from Figure A.7 along with the locations of the HITS fields and approximated the DECam field of view as a circle with 2.2° diameter. With this information, we recorded the objects in the Monte Carlo simulation visible through the survey pattern. We then scaled this distribution to the discovery of the same number of objects that we found in the data and plotted them on top of one another in Figure A.8. A χ^2 test on this data gives $\chi^2 = 2.23$ and a p-value of 0.69, meaning we cannot reject the hypothesis that our observed samples come from the Brown (2001) inclination distribution function for the general KBO population.

Magnitude Distribution

We next compare the magnitudes of our discoveries to the KBO magnitude distribution of Fraser et al. (2008). This distribution gives the number of observed objects per square degree as:

$$N_{obs} = 10^{\alpha(m-m_o)} \tag{A.31}$$

where $\alpha = 0.65 \pm 0.05$ and $m_o = 23.42 \pm 0.13$ for R magnitudes. Since Equation A.31 is based upon the number of expected objects per square degree at the ecliptic we needed to scale our viewing area appropriately. Satisfied by the work in Section A.4.2, where we showed our results are consistent with the Brown (2001) inclination distribution, we converted this to a latitudinal distribution. We then multiplied the 3 square degree DECam viewing area

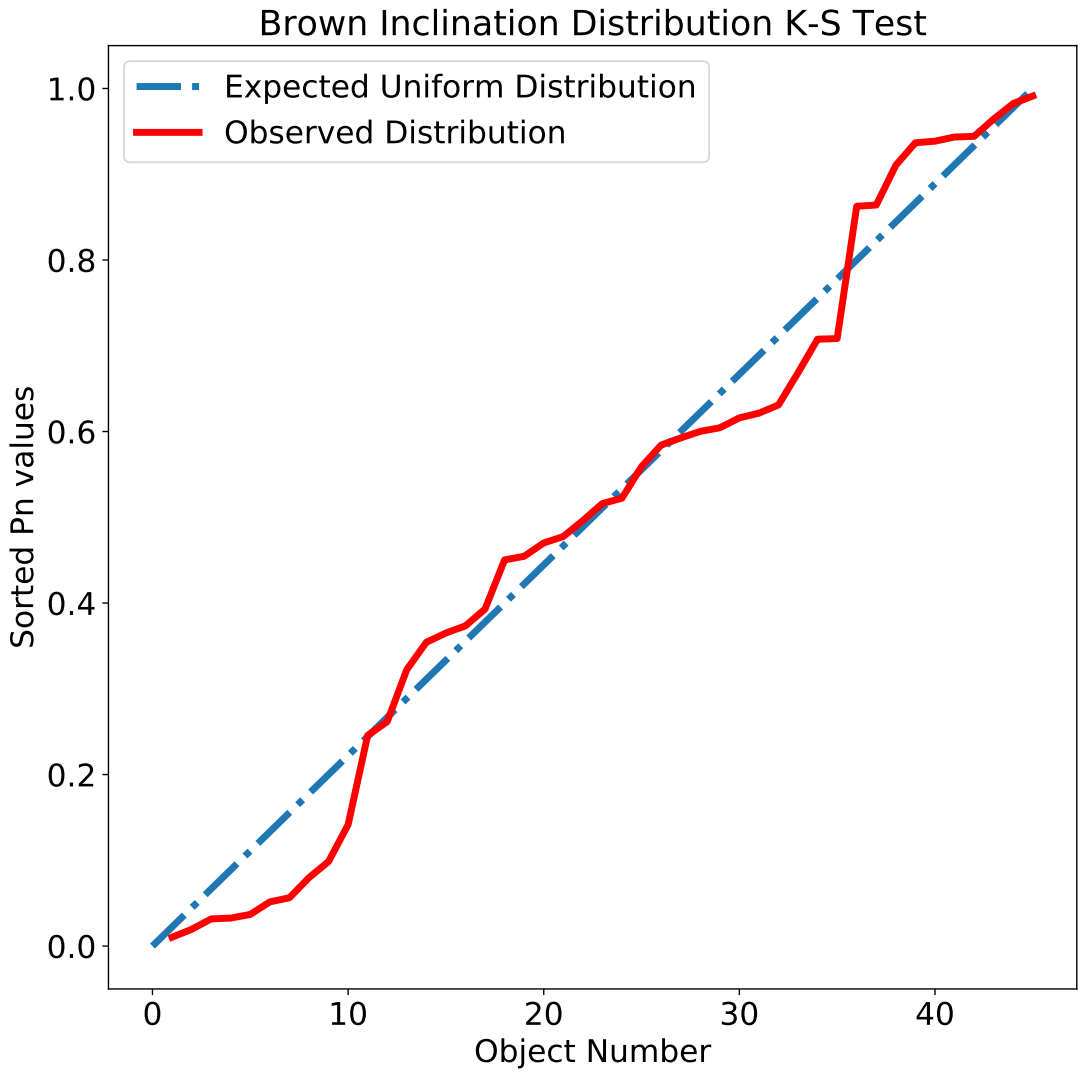

Figure A.7: Kuiper variant of K-S Test comparing Brown (2001) inclination distribution to our recovered results.

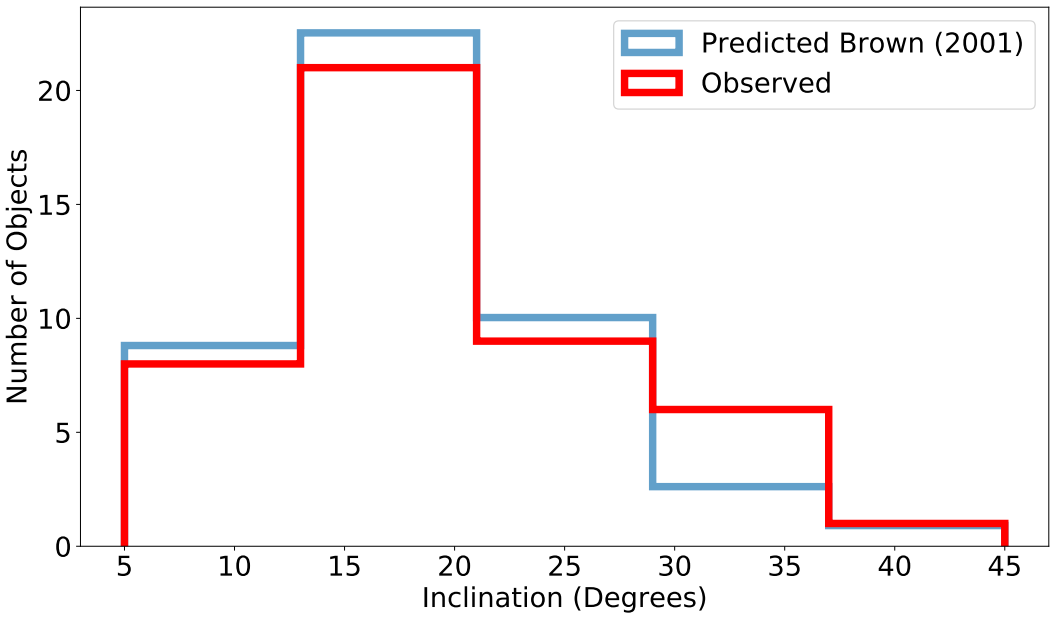

Figure A.8: Inclination distributions of detected objects in the HITS fields compared to predicted Brown distribution accounting for the ecliptic latitudes of the HITS observations and normalized to the same number of discovered objects.

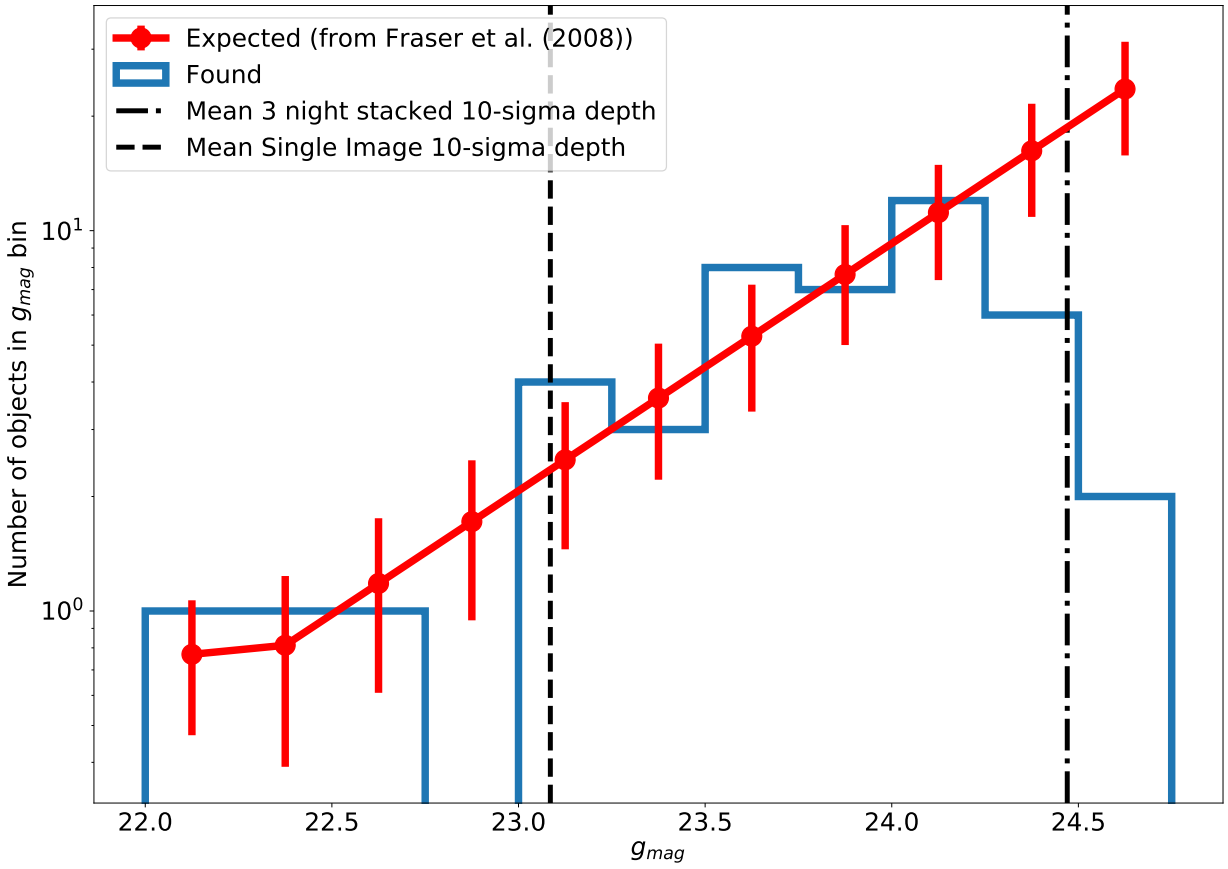

Figure A.9: Completeness comparison of KBOs found in HITS survey using KBMOD. Our results are consistent with a complete sample at 24th magnitude compared to the single image depth of 23.1.

for each field by the fraction of expected objects at the field's ecliptic latitude compared to the maximum value at the ecliptic. This gave us an effective viewing area of 91.05 square degrees. The next step was converting between the R magnitudes used for Equation A.31 and the g magnitudes of our observations. Fraser et al. (2008) also had to do magnitude conversions for various datasets and used a KBO $< g' - R >$ color of 0.95 which we use here as well. Putting together the scaling and magnitude offsets we compare our results to the Fraser et al. (2008) expected number of objects for our survey fields in Figure A.9. The

drop-off expected from Section A.4.1 around 24th magnitude is present and seems to occur after $g = 24.25$ after which we fall below 50% completeness which is consistent with the magnitude where our search drops below 50% efficiency as shown in Section A.4.1. Figure A.9 also shows the 10-sigma threshold we used in our search of the HITS data. A χ^2 test on the data for $g < 24.25$ gives $\chi^2 = 10.21$ and a p-value of 0.25, meaning our results are consistent with the Fraser et al. (2008) luminosity distribution.

A.5 Discussion

A.5.1 Filtering Analysis

We used the field with simulated objects from Section A.4.1 to study the effects of the various stages of our filtering process. The red line in Figure A.10 shows the results for the searches over the full field in our processing which included an extra masking step that we discuss below. We start with the total number of searches based upon the number of grid steps multiplied by the number of pixels on the focal plane. We only keep those above our detection threshold which is the biggest single reduction in the number of possible results. After that, each step in our filtering process is able to reduce the number of false positives by 1-3 orders of magnitude in the number of total results. The most effective is the postage stamp filtering which uses the moments of the postage stamp image and their resemblance to those of a Gaussian source model.

A.5.2 Masking and Threshold Effects

We made a series of decisions in our search based upon our target objects and the use of science images instead of less contaminated difference images. The first choice we made was to include an additional bright pixel mask when creating our Ψ and Φ images. This mask was in addition to the masking described in Section A.3. Any pixel in any image that exceeded 120 counts, corresponding to $g \sim 21$, was masked in our searches. This additional masking covers extended bright halos, that were not covered by our initial footprint masking, and

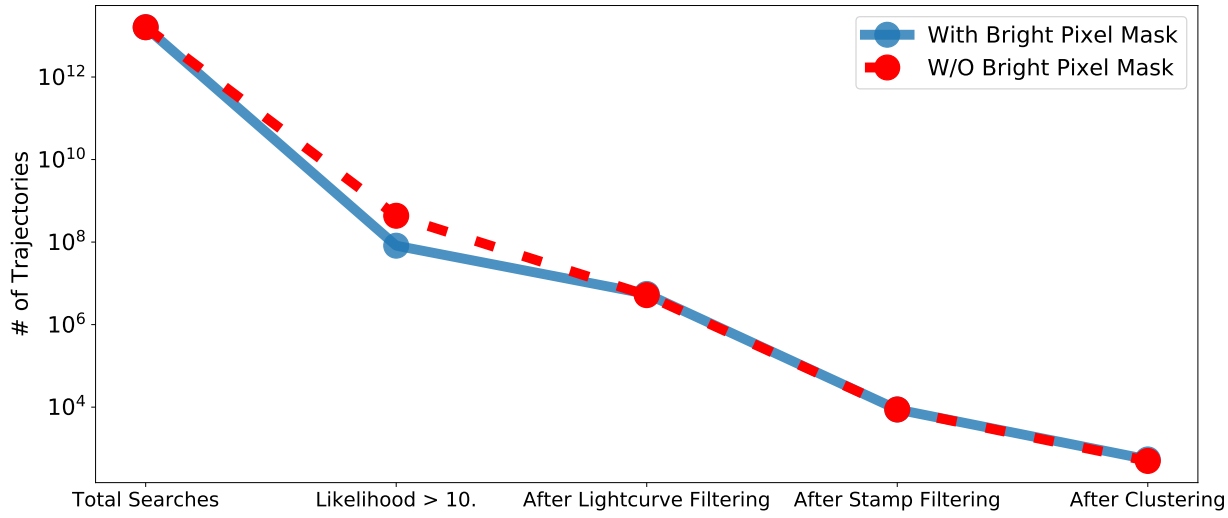

Figure A.10: Number of positive trajectories after each step of filtering the search results from a field with simulated objects inserted.

bright fast moving objects which are likely much closer to Earth than our target population. The effects of additional masking on recovery were tested on the simulated object field (Section 4.1). Comparing the red line with this extra masking to the blue line without it in Figure A.10 shows the major benefit of this masking was a significant decrease in the amount of false positives we had to filter out after our likelihood cut. The amount of positive results after the likelihood cut in Figure A.10 shows about five times more objects to filter after the threshold cut without the masking. A decrease of 4×10^8 false positives over a single field saves us hours of computation time during our analysis since we processed the results through the lightcurve filter at a rate of 70 seconds for 500,000 objects. Over 50 fields this adds up to days of processing time that we were able to avoid.

However, we also looked to see what was the price we paid for this extra computational efficiency. Figure A.11 compares the efficiency curves of Section A.4.1 and Figure A.6 to the results without the additional bright pixel masking. The recovery efficiency is higher across all magnitudes at $92.7 \pm 1.5\%$, but when we compare to the recovery at only $g > 21$ with

the masking it is very similar to the $92.4 \pm 1.2\%$ of that result. The 50% efficiency depth is nearly unchanged at $g = 24.09 \pm .03$ without the masking meaning the bright pixel masking does not affect our overall depth but only the bright end of our recovery at $g < 21$. Using the Fraser et al. (2008) curve for the expected number of KBOs for magnitudes $20 < g < 21$ in our survey fields, we calculate that the number of expected objects in this magnitude range to be less than 0.3. We find this to be an acceptable trade off for the computational savings in this particular example. When going to difference imaging or with a different population of brighter expected objects we do not plan to include this extra masking step.

Another decision we made was to set our threshold for detection at a limit of 10σ instead of a lower, typical catalog level threshold of 5 σ. This decision was motivated by the use of science images and the extra noise that would be present as this threshold was lowered. We looked at the increase of false positives as we go from 10σ to 5σ by rerunning the same field as Section A.4.1 without any simulated objects and the timestamps randomly scrambled so that any detections were false positives. We show histograms comparing the false positives to true detections at the two thresholds in Figure A.12. We also calculated the precision of our final results where precision is defined as $\frac{True\ Positive}{True\ Positive\ +\ False\ Positive}$ and show the results in Figure A.13. While we do detect objects to over a magnitude fainter ($g \sim 25.4$) with the lower likelihood threshold we are overwhelmed by false positives at the fainter magnitudes and even at magnitudes at which we achieve high precision with a higher threshold. At a threshold of 5σ we are below 50% precision at $g = 24.1$ while we are never below 90% precision when using a 10σ threshold. This degradation at a brighter magnitude in the 5σ results happens because bright false positives with fewer observations will appear at a higher likelihood than a dim object with more observations (e.g., a greater number of the observations were in a masked area or off the edge of the CCD). Due to this false positive performance when running on science images we used the 10σ threshold in this work to show the effectiveness of the algorithm and code while also producing useful results. We discuss our future plans in the next section including what we will do to run the code at a lower detection threshold going forward.

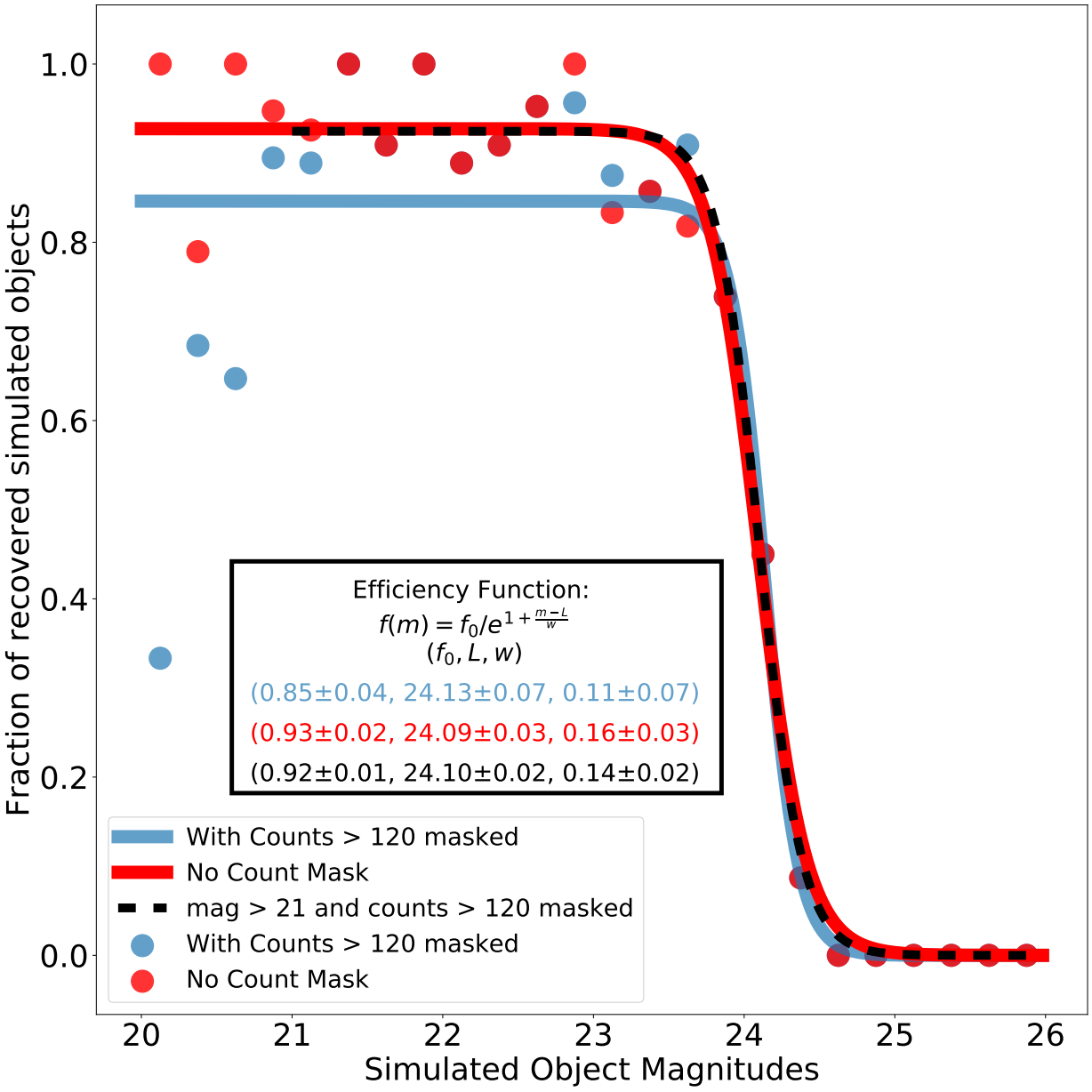

Figure A.11: Comparison of efficiency curves run on the same simulated object set with and without the count masking described in Section A.5.2. There is almost no effect on the depth of our recoveries.

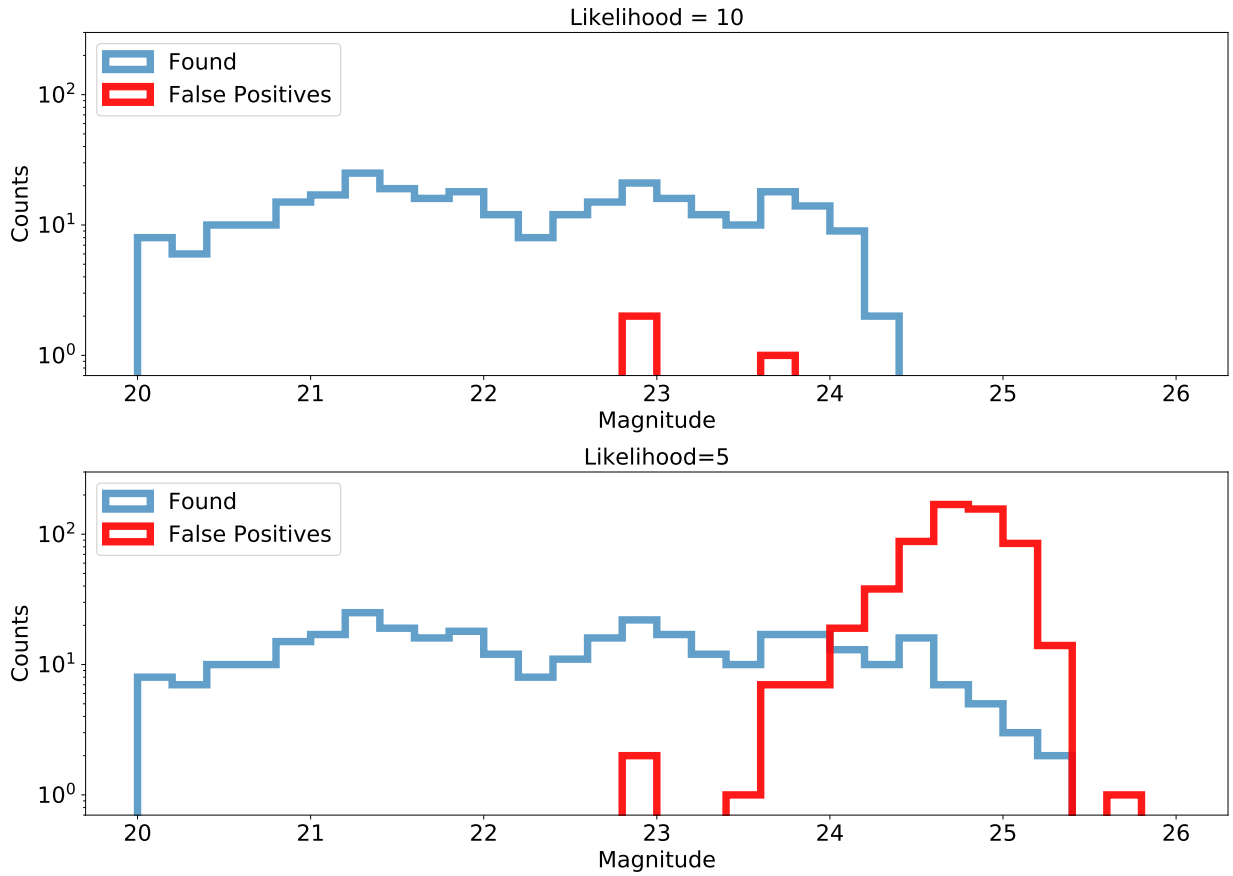

Figure A.12: Comparing the false positives at detection thresholds of 10σ versus 5σ in a field with simulated objects inserted. False positives overwhelm our filtering methods at the lower threshold when using science images.

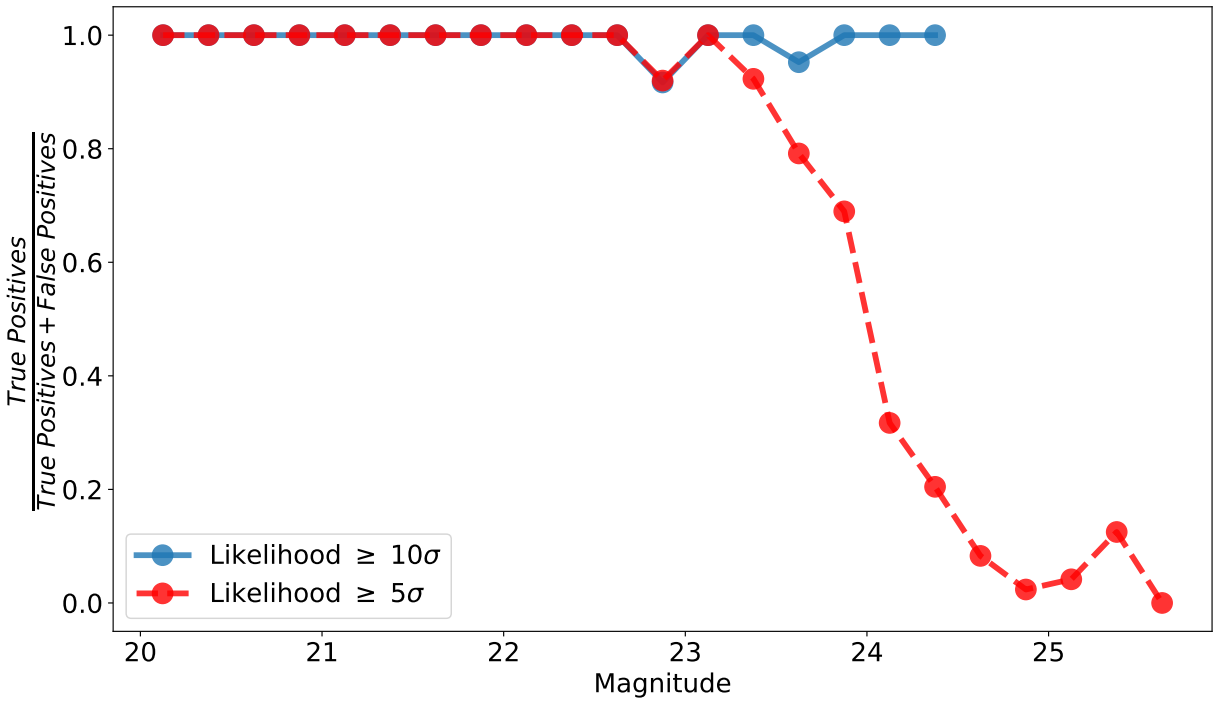

Figure A.13: Comparing the precision at detection thresholds of 10σ versus 5σ in a field with simulated objects inserted. We are very confident in our detections at 10σ with science images but would not be if we were to go to 5σ detections with the science images.

A.5.3 Future Work

There is ongoing work to decrease the detection threshold for sources in order to extend the searches ov er longer timescales thus allowing us to find fainter and more distant objects. Work with longer baselines and larger stacks of images would also allow us to push this detection limit lower and confidently go beyond the limits of current population studies of KBOs. We plan to use the code with difference images in the future which we hope will reduce the time spent filtering and also remove many of the artifacts from science images that lead to false positives above 5σ and also pushed us to add in additional masking procedures in this work. We also plan to move to a deep-learning-based postage stamp classifier that we hope will perform well at fainter magnitudes and plan to do a comparison versus our existing technique on the difference imaging search. We will also address working with non-linear trajectories and methods to find objects that move from one chip to another during the survey period. Finally, we are also working on enhancements to the GPU algorithm that will allow us to spend less time searching low likelihood trajectories thereby increasing search efficiency.

All of these improvements will help us explore deeper into the Kuiper Belt and enhance our ability to use stacks of images from long baseline surveys such as ZTF or LSST in the future.

A.6 Conclusion

In this paper we presented a new algorithm that uses the power of GPU processing to search for slow moving sources across a sequence of images. Our approach is capable of searching over 10^{10} candidate moving object trajectories in one minute. Applying these techniques to existing data we discovered 39 new KBOs and the recovered 6 KBOs already present in the Minor Planet Center catalogs. Finally, we used the results of our search to compare to the Brown (2001) Kuiper Belt inclination distribution and the Fraser et al. (2008) magnitude distribution. We found both of the models to be consistent with observed results indicating

that the recovered sample matches overall published characteristics of the KBO population. Combined with the high rates of detection efficiency recovered in tests, this indicates that our software provides a nearly complete recovery of an unbiased sample of moving objects down to the detection limit.

Our software, Kernel Based Moving Object Detection (KBMOD) is available to the public at `https://github.com/dirac-institute/kbmod`. This includes the GPU searching code as well as python analysis code we have used to process the search results. Development is continuing and can be followed at the Github repository hosting the code.

Table A.1: Detected KBOs in HiTS field with estimated orbit properties from HiTS data

MPC designation	Semi-Major Axis (AU)	Eccentricity	Inclination	g_mag	New Discovery?
2015 DZ248	67.33	0.42	17.60	23.68	Yes
2015 DT248	21.38	0.36	22.21	24.14	Yes
2015 DT249	43.67	0.20	25.76	23.74	Yes
2015 DY248	48.02	0.28	14.72	24.19	Yes
2015 DX248	37.61	0.02	24.67	24.24	Yes
2015 DV248	42.32	0.67	27.67	23.94	Yes
2015 DW248	48.52	0.62	31.42	23.02	Yes
2015 DU248	44.24	0.02	14.44	23.99	Yes
2015 DQ248	63.15	0.45	33.13	24.05	Yes
2015 DR248	41.61	0.31	15.81	23.95	Yes
2015 DS248	42.67	0.78	23.02	24.00	Yes
2015 DO248	46.90	0.27	22.65	24.08	Yes
2015 DP248	48.97	0.70	21.73	24.15	Yes

MPC designation	Semi-Major Axis (AU)	Eccentricity	Inclination	g_mag	New Discovery?
2015 DO249	48.99	0.29	19.88	24.35	Yes
2015 DP249	49.49	0.43	17.06	24.28	Yes
2015 DZ249	45.05	0.25	14.27	24.30	Yes
2015 DY249	38.14	0.03	25.72	24.41	Yes
2015 DN249	38.30	0.03	17.08	24.34	Yes
2015 DM249	44.39	0.14	12.05	23.74	Yes
2015 DX249	47.08	0.24	30.04	23.22	Yes
2015 DL249	44.18	0.27	13.83	24.23	Yes
2015 DK249	44.29	0.02	7.03	24.05	Yes
2015 DH249	46.85	0.02	20.71	24.63	Yes
2015 DJ249	42.81	0.19	12.99	23.44	Yes
2015 DW249	38.83	0.19	15.11	23.68	Yes
2015 DC249	41.67	0.16	20.77	23.62	Yes
2015 DD249	65.74	0.32	19.25	23.65	Yes
2015 DB249	40.31	0.27	24.58	23.56	Yes
2015 DU249	59.50	0.21	18.87	24.72	Yes
2015 DA249	38.40	0.15	19.97	23.61	Yes
2015 DF249	45.35	0.44	12.99	24.08	Yes
2014 BV64	50.20	0.31	15.43	22.10	No
2015 DE249	41.66	0.35	20.29	23.35	Yes
2015 DV249	54.12	0.15	11.79	24.00	Yes
2015 BF519	45.12	0.39	18.42	23.00	No

MPC designation	Semi-Major Axis (AU)	Eccentricity	Inclination	g_mag	New Discovery?
2015 DG249	64.80	0.33	15.44	24.19	Yes
2011 CX119	47.70	0.46	29.47	23.19	No
2015 DA250	45.12	0.29	34.11	23.89	Yes
2015 DB250	44.13	0.27	10.99	23.80	Yes
2014 XW40	58.03	0.32	16.43	23.39	No
2015 DQ249	28.07	0.44	43.91	23.93	Yes
2015 DR249	41.10	0.02	35.13	23.81	Yes
2014 XP40	78.02	0.62	11.53	22.33	No
2013 FZ27	57.12	0.25	13.54	22.73	No
2015 DS249	31.08	0.29	11.38	24.37	Yes