TERM PROJECT FINAL REPORT

TROY RAEN

Abstract. In this work I study how the errors in photoz estimates scale with the size of the training set for various machine learning algorithms. A 'photoz' is an estimate of the redshift of a particular object (usually a star or a galaxy) that uses photometric data. In this work I study how the errors in photo-z estimates scale with the size of the training set for two neural net architectures, a random forest algorithm, and a publicly available code developed specifically for photo-z's called GPz [1]. My code is available at `https://github.com/troyraen/photoz_errors`.

1. Introduction

It is well established that the light reaching our telescopes from distant galaxies is shifted toward the red end of the spectrum (relative to the frequency it was originally emitted at), and that the magnitude of this 'redshift' (denoted by z) increases with the galaxy's distance from us (e.g. see [6], [3], [5]). The combined measurements from many galaxies indicate that the universe itself, the space between galaxies, is expanding at a rate that increases with time. More precise calculations of this expansion rate, and several other quantities fundamental to cosmology, is being pursued, and they all depend strongly on our ability to make accurate redshift calculations for large numbers of galaxies.

The dataset used in this work is simulated and intended to mimic the data anticipated from the upcoming Large Synoptic Survey Telescope (LSST). LSST will collect data from large volumes of the sky and at rates several orders of magnitude above any other telescope to date. The community is making large efforts towards dealing with data at this scale, and one of these efforts is toward quick and accurate redshift calculations.

1.1. Dataset.

I use the dataset `Catalog_Graham+2018_10YearPhot` (see [3]) which consists of simulated telescope data: measurements of light in 6 frequency ranges (bins), plus errors on the measurements, for 3.8×10^6 galaxies. The dataset includes the true redshift for each galaxy, so this is a supervised, regression problem.

Figure 1 shows the true redshift distribution of the galaxies in the dataset. Projections along principal components that minimize the covariance in the data is shown in the table below. Figure 2 shows pairwise scatter plots of the features and the target along with correlation coefficients. I chose the two features with the highest absolute value correlations with the target redshift and show their scatter plot, colored by redshift, in Figure 3 to get a sense of the distribution.

PCA vectors

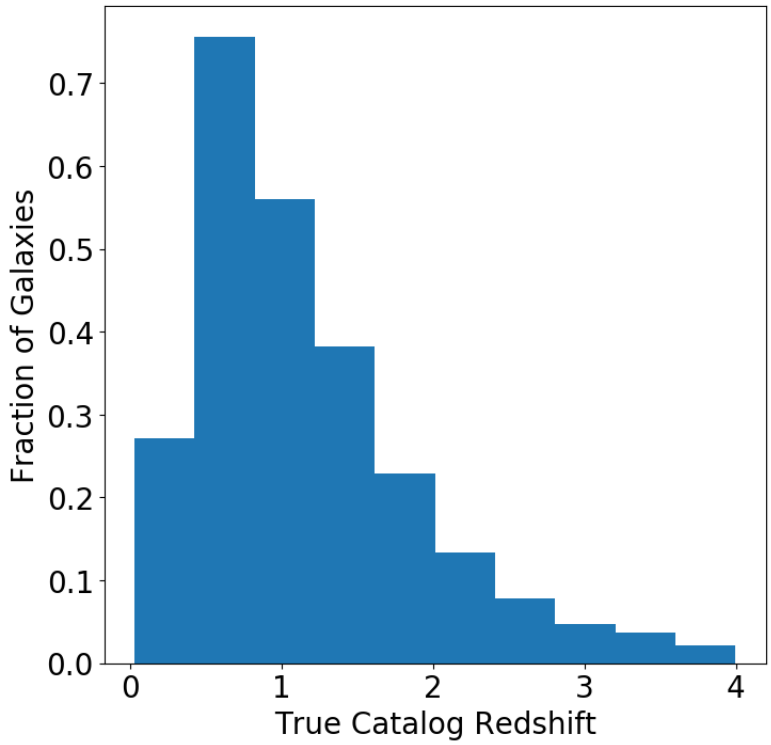
	pc1	pc2	pc3	pc4	pc5	pc6
u	0.7977	-0.5534	-0.2285	-0.0726	-0.004188	-0.001952
u-g	0.5969	0.7661	0.1783	0.1568	0.02073	0.0004168
g-r	0.08518	-0.1745	0.8555	-0.4118	-0.2461	-0.01691
r-i	0.01012	-0.2212	0.3991	0.4946	0.7049	0.2239
i-z	-0.006634	-0.1591	0.1573	0.6848	-0.4525	-0.5255
z-y	-0.003662	-0.04681	0.00884	0.2948	-0.4872	0.8206

1.1.1. *Features:*

Previous algorithms have had more success using a set of transformed features commonly called 'colors'. This transformation is motivated by physics, and is done by subtracting the measurements in adjacent bins, pairwise, bringing the 6 measurements down to 5 features. By including the original measurement from one bin (it doesn't matter which one, unless it carries an unusually large error), no information is lost in the transformation. So the final feature set includes 5 colors and 1 raw measurement for a total of 6 features.

We could also transform the measurement errors and include them as features. The simplest way to do this is to assume the errors are uncorrelated and add them in quadrature. Indeed, this is what is done in [3]. However, with one exception (see 2.1.3), I leave them out of the final dataset for two reasons. First, I'm not sure how the errors were calculated (this is simulated data), and it is possible that the information used to calculate the true redshift was also used to calculate these errors in a way that provides the algorithm with access to the true information, even in the test set. Second, there are multiple factors that could cause the errors to be correlated, so adding them in quadrature is not necessarily the best thing to do, especially since I don't know where they came from to begin with. A more thorough analysis of the errors in the dataset is an avenue for future work.

FIGURE 1. Histogram of true redshifts in the dataset, reproduced from [3]. True redshifts become more difficult to obtain as the redshift increases. Algorithms may have a more difficult time making accurate predictions at higher z because of the sparsity of training data.



1.1.2. *True Redshift (spec-z).* The calculation of the redshift from measurements of light generally depends on being able to find known features in the intensity as a function of frequency and measure how far they have been shifted along the spectrum. As a result, poor frequency resolution propagates to increased error in the redshift. This becomes important when we consider the two ways in which telescopes can take measurements: spectroscopy and photometry.

Spectroscopy records information about the amount of light coming in over a wide range of the frequency spectrum, at high resolution. Therefore, a redshift calculated from spectroscopic measurements is very precise and can be taken as the true redshift (sometimes called a 'spec-z').

Photometry essentially divides the spectrum into a small number of bins (usually on the order of 5) and records only aggregated information for each bin. Thus photometry is much cheaper to do and so we have (and LSST will be able to get) much more data of this type. This data can be collected quickly for large numbers of galaxies and used by ML algorithms to estimate the redshift (called a 'photo-z'). However, the low resolution necessarily leads to less accurate predictions. This provides further motivation to study how the errors scale with sample size for various algorithms.

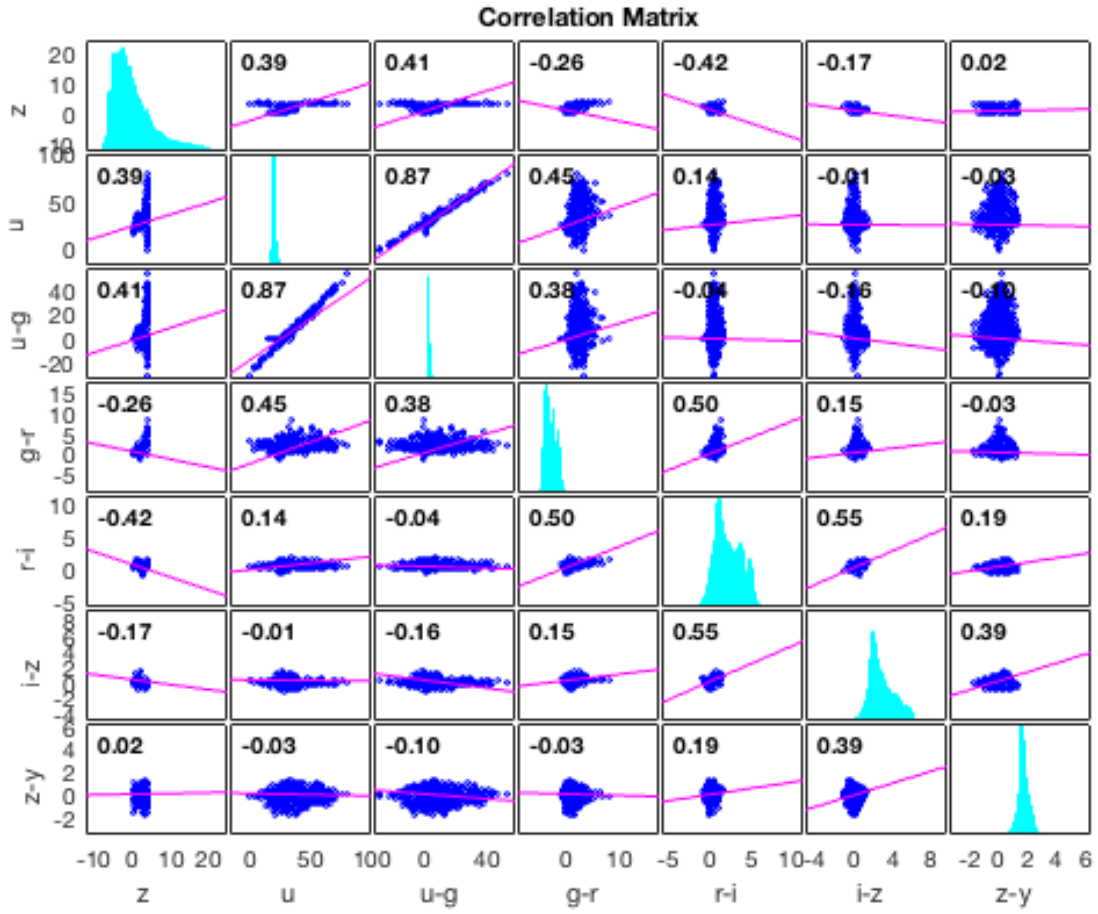
2. Methodology and Results

I follow [6] in evaluating the accuracy of photo-z estimates as a function of sample size. Various machine learning algorithms have been used to estimate photo-z's, with neural nets (NN) and random forest regressors (RF) showing the most success. See, for example, [5], [7], and [4]. I evaluate and compare the performance of four ML algorithms: 1) Neural Net composed of 2 hidden layers with 10 units each (NN_2x10); 2) Neural Net composed of 3 hidden layers with 15 units each (NN_3x15); 3) Random Forest regressor (RF) composed of bagged decision trees; and 4) GPz which is a publicly available code based on Gaussian Processes and developed specifically for photo-z's. See section 2.1 for details.

My main results (Figure 4) are in the form of curve fits to two statistics (detailed below) calculated on the algorithm predictions. Specifically I fit (for each statistic) the parameters {a,b,c} in the function $a + b \times N^c$, (where N is the training sample size) using the Python function `scipy.optimize.curve_fit()`. We are primarily interested in the value of the exponent, c. The traditional metric for evaluating photo-z estimates is the scaled difference

$$\Delta z \equiv \frac{photo_z - true_z}{1 + true_z}. \tag{1}$$

FIGURE 2. Pair-wise scatter plots with histograms along the diagonal. Correlation coefficients are printed on each plot.



I evaluate the following two statistics on the metric:

$$\text{NMAD} = 1.48 \times \text{median}(|\Delta z|) \tag{2}$$

$$\text{OUT10} = \frac{1}{N}\sum_{n=1}^{N}\left[|\Delta z_n| > 0.1\right] \tag{3}$$

NMAD is the normalized, median absolute deviation and OUT10 is the fraction of predictions for which $|\Delta z| > 10\%$. OUT10 is an important statistic because photo-z algorithms are prone to catastrophic errors in the predictions, due to both the physics involved and the inherently low resolution of photometry.

For the NN and RF cases I use the same test set of 10^5 randomly selected galaxies (separated from the training sets prior to training) for the predictions. The GPz code handles the train/validate/test dataset splits internally, but I set Ntest $= 10^5$ to maintain some consistency.

Because I aim to evaluate and compare the performance of different algorithms (rather than any specific instance of a trained model), I train 20 models for each algorithm and training sample size (N) and pool the results before calculating the statistics.

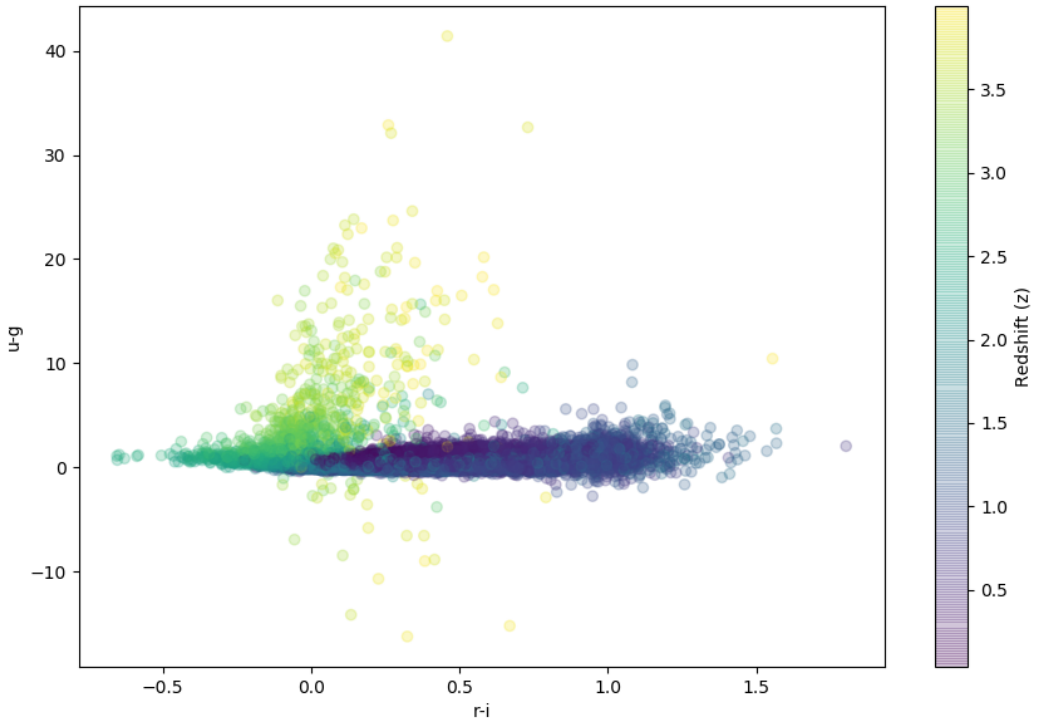
2.1. **Algorithms.**

2.1.1. *Neural Nets.* I train multi-layer neural networks using two different architectures: 2x10 with 2 hidden layers, each with 10 units; and 3x15 with 3 hidden layers, each with 15 units. Both are motivated by approaches in [5] (see sections 4.1.1 DESDM and 4.1.2 ANNZ).

I use the Matlab `fitnet()` function with backpropagation optimized using the Levenberg-Marquardt method. After running a few tests I set the parameters `epochs=500`, `max_fail=50`, `min_grad=1e-10` and use the `tanh` transfer function.

In addition, I did two smaller tests, each with 6 sample sizes and 5 runs per sample size:

FIGURE 3. Scatter plot of the two features with highest absolute value correlations with the redshift, colored by true redshift. Galaxies at low z tend to cluster near $u - g = 0$. At high z there is a much larger scatter in $u - g$ but the $r - i$ values tend to be lower. This is a random sample of 10000 galaxies to avoid saturating the plot. I verified that the plot looks qualitatively similar for different random samples.



(1) The predictions improve with the more complex network (3x15), so I ran a smaller test using 3 hidden layers with 50 units to see if the trend continued.
(2) RELU (`'poslin'` in Matlab) is a much more common transfer function in photo-z algorithms, so I tested this as well.

All NN results are shown in Figure SHOW PLOT

2.1.2. *Random Forest Regression.* I train random forest regression models using the Matlab `fitrensemble()` function. I did some preliminary runs with `OptimizeHyperparameters='auto'` and found the following "best" options:

Method	Bag
NumLearningCycles	495
MinLeafSize	1

Guided by these results I tested some settings and ultimately use `Method='Bag'` with `Learners='tree'`, `MaxNumSplits=Nsamples-1`, and `NumLearningCycles=500`, `Crossval='off'`. This generates a random forest model using 500 weak learner decision trees, each trained on a subset of data of size N generated via bootstrap resampling.

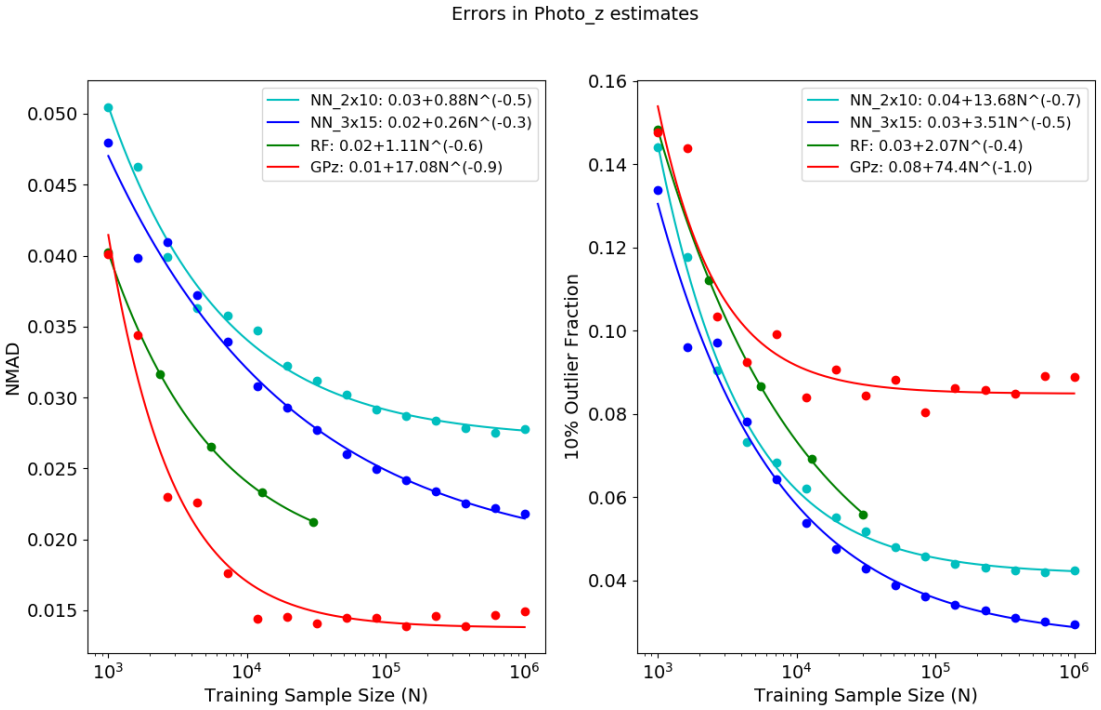
RF regressors predict a target value for each leaf by minimizing a loss function.

EXPLAIN RF REGRESSION

2.1.3. *Gaussian Process Regression using GPz.* GPz is a publicly available code developed specifically for photo_z estimates. The method is described in [2], and the specific code is introduced in [1] and available at `https://github.com/OxfordML/GPz`. Since we studied this type of technique only briefly in the course I will outline the approach in a little more detail than I have done for NN and RF.

A Gaussian Process (GP) is a non-parametric, non-linear, regression algorithm. It assumes output, y, is predicted by some function of the input, $\mathbf{x}$, plus Gaussian noise $\epsilon \sim N(0, \sigma^2)$:

FIGURE 4. Errors in photo-z estimates as a function of training set size for selected ML algorithms. Figure 5 shows the results of [6] for comparison.



$$y = f(\mathbf{x}) + \epsilon.$$

Then the conditional probability of y given f is Normally distributed as $p(y|f) \sim N(f, \sigma^2)$ and Bayes' Theorem can be used to write

$$p(f|y, \mathbf{X}) = \frac{p(y|f)p(f|\mathbf{X})}{p(y|\mathbf{X})} \tag{4}$$

The prior, $p(f|\mathbf{X})$, is modeled non-parametrically (except for hyperparameters) using kernels that model the density around each input point. GPz uses radial basis functions for these kernels. Standard GP models are computationally expensive since there is a kernel for each datapoint and the solution requires us to invert an NxN covariance matrix associated with the kernels. GPz dramatically reduces the complexity by using sparse kernels and maintains performance by optimizing hyperparameters (governing the shape and length scale) that are unique to each kernel rather than standard, global hyperparameters. This allows kernels to specialize on different regions of parameter space, and this flexibility is cited as a key reason for the success of GPz (see [2] for a detailed derivation).

GPz also has a large component that is focused on estimating the variance in the prediction due to two factors, the input noise and uncertainty due to the density of training points in a particular region. 'True' redshifts are more difficult to obtain at higher redshifts, and so the training data is not uniformly sampled (see Figure 1) I only use the point estimate of (4) in this work, so I will not describe the error calculations. However, this means that GPz requires the use of the measurement errors, so these models use more information than the NN or RF cases.

GPz performs the feature transformation internally, so the inputs to this algorithm are the raw measurements, including the errors. Additionally, it minimizes $|\Delta z|$ directly rather than the more standard mean squared error.

I ran several small tests (SHOW PLOT) with different GPz settings and found two that improved the predictions and used them in my final runs (Figure 4): increasing the maximum number of iterations, and using the errors as features (rather than including them in the prediction uncertainty). I chose to use the errors as features since the use of the errors is required one way or the other. I intended to do a more fair comparison with the NN and RF models by following the method used in [3] to add the error features to those datasets, but time constraints prevented this. In retrospect it would have been a better choice to use the errors in the opposite way for the GPz final runs.

3. Conclusions and Future Work

Overall, the NN models I tested perform the worst and

FIGURE 5. Reproduced from Newman et al., 2019 ([6])

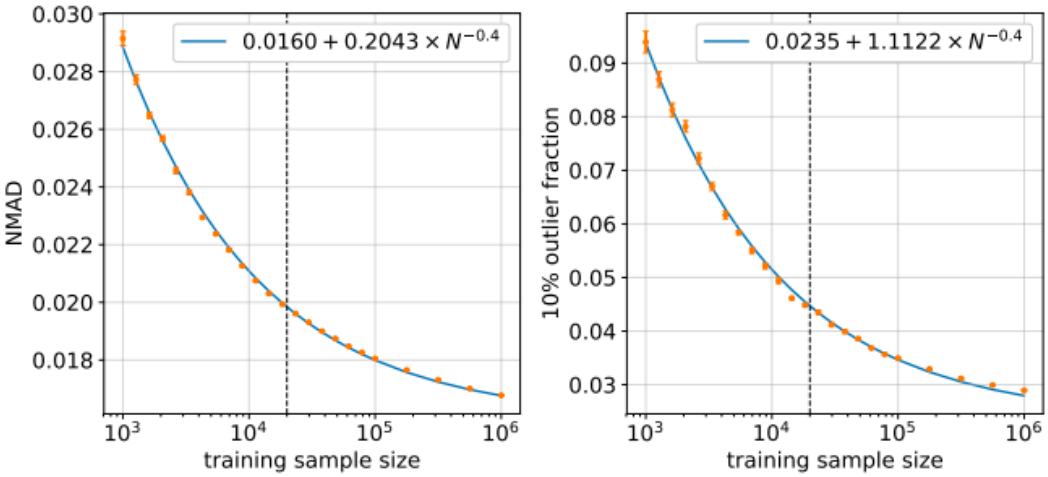


Figure 1: Orange points show photometric redshift errors and outlier rates versus the number of galaxies in the training set for galaxies with simulated LSST photometric errors. Photo-z's were calculated using a random forest regression algorithm. The left panel shows the photo-z error, quantified by the normalized median absolute deviation (NMAD) in $(z_{\rm phot} - z_{\rm spec})/(1+z_{\rm spec})$, as a function of training set size; similarly, the right panel shows the fraction of 10% outliers, i.e. objects with $|z_{\rm phot} - z_{\rm spec}|/(1+z_{\rm spec}) > 0.1$. A vertical dashed line shows the sample size for the baseline training survey from [5]. The blue curves represent simple fits to the measurements as a function of the training set size, N. This analysis uses a set of simulated galaxies from Ref. [11] that spans the redshift range of $0 < z < 4$, using a randomly-selected testing set of 10^5 galaxies for estimating errors and outlier rates.

This feature is also noted in [1] (small improvement after $\sim 10^4$).

For a more fair comparison between GPz and the other models, more effort should be devoted to handling the measurement error features more carefully. They could be added to the feature set for NN and RF, and/or they could be artificially set to zero in the GPz feature set.

TRY VARYING m, THE NUMBER OF BASIS FUNCTIONS [2] pg 4, but plot 5 shows delta z does not vary much with training size for a given number of basis fncs =¿ this could improve the predictions but should not change shape of scaling with N? Could bring down out10?

3.1. **Future Work.** Understand errors included in the dataset. Possibly use them as features.

- There is a lot of room here to tune the algorithms for a closer comparison with existing literature and to test other algorithms written specifically for photo-z (e.g. ANNz2 [7], TPz [4]). Initially I was going to test ANNz2, and I spent several days trying to get it to install properly before abandoning it in favor of GPz.
- Experiment with the number of basis functions used by GPz. Find a small number (of order 5) of basis functions that still produces good predictions (must quantify 'good'). Compare these fitted kernel density estimates, and where they lie in parameter space, with two things:
  (1) Current physical models that predict specific types of galaxies and what their photometry measurements should look like as a function of redshift. In other words, x-type galaxies at redshift z should live in R-region of parameter space.
  (2) Results of unsupervised, clustering methods which may provide insight on how many distinct types of galaxies there are as a function of redshift.
- Density estimation on the subset of the test data with out10 > 10% to see if there are localized regions of the parameter space that are not well predicted. These results could also be compared to (1) and (2) above to search for insight.

REFERENCES

[1] I. A. Almosallam, M. J. Jarvis, and S. J. Roberts. GPZ: Non-stationary sparse Gaussian processes for heteroscedastic uncertainty estimation in photometric redshifts. *Monthly Notices of the Royal Astronomical Society*, 462(1):726–739, 2016.

[2] I. A. Almosallam, S. N. Lindsay, M. J. Jarvis, and S. J. Roberts. A sparse Gaussian process framework for photometric redshift estimation. *Monthly Notices of the Royal Astronomical Society*, 455(3):2387–2401, 2016.

[3] M. L. Graham, A. J. Connolly, Ž. Ivezić, S. J. Schmidt, R. L. Jones, M. Jurić, S. F. Daniel, and P. Yoachim. Photometric Redshifts with the LSST: Evaluating Survey Observing Strategies. 1, 2017.

[4] M. C. Kind and R. J. Brunner. 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, 2013.

[5] C. Lidman, H. Lin, D. Burke, C. Cunha, N. Greisel, J. Zuntz, A. Fausti, K. Glazebrook, G. Bernstein, J. Gschwend, D. Gerdes, M. A. G. Maia, C. Sánchez, S. S. Allam, A. Dey, E. Sánchez, H. T. Diehl, D. Finley, M. Lima, D. Capozzi, R. Scalzo, A. Sypniewski, S. Jouvel, E. Fernández, I. Sadeh, S. Seitz, J. L. Marshall, J. de Vicente, J. Frieman, A. Walker, P. Pellegrini, I. Sevilla-Noarbe, A. Roodman, M. Carrasco Kind, M. J. Childress, B. X. Santiago, R. C. Nichol, F. J. Castander, A. Carnero, R. Miquel, P. Doel, F. Ostrovski, T. Davis, A. Kim, A. Evrard, F. Valdés, S. Desai, J. Estrada, O. Lahav, C. Bonnett, G. Tarle, F. B. Abdalla, D. L. DePoy, E. Gaztanaga, J. P. Bernstein, R. L. C. Ogando, D. L. Tucker, M. E. C. Swanson, W. Hartley, A. Amara, F. Yuan, K. Honscheid, S. A. Uddin, N. Kuropatkin, T. Abbot, D. Thomas, B. Flaugher, M. M. Rau, R. Brunner, E. Buckley-Geer, L. A. N. da Costa, M. Sako, K. Kuehn, M. Banerji, D. Atlee, P. Martí, M. Makler, and R. C. Smith. Photometric redshift analysis in the Dark Energy Survey Science Verification data. *Monthly Notices of the Royal Astronomical Society*, 445(2):1482–1506, 2014.

[6] J. A. Newman, J. Blazek, N. E. Chisari, D. Clowe, I. Dell'Antonio, E. Gawiser, R. A. Hložek, A. G. Kim, A. von der Linden, M. Lochner, R. Mandelbaum, E. Medezinski, P. Melchior, F. J. Sánchez, S. J. Schmidt, S. Singh, and R. Zhou. Deep Multi-object Spectroscopy to Enhance Dark Energy Science from LSST. 2019.

[7] I. Sadeh. ANNz2 - Photometric redshift and probability density function estimation using machine-learning. *Proceedings of the International Astronomical Union*, 10:316–318, 2015.