Monterey Bay Aquarium
Research Institute

Developing A Regional Neural Network Framework to Accurately Predict Ocean pH Using Glider Observations in Central California

Adam Gibbs, Amherst College

Mentor: Dr. Yui Takeshita

Summer 2021

Keywords: ***Machine Learning; Empirical Algorithm; Ocean Acidification; Carbonate Chemistry; Coastal Oceans***

Abstract. Due to frequent upwelling and highly dynamic surface waters, the coast of California is a hotspot for changing pH on the short- and long-term scales. Reductions in pH can damage ecosystems and fisheries, resulting in negative consequences for the West Coast economy and a desire to better understand changes in ocean pH. However, we do not have datasets with enough spatiotemporal variability in pH measurements to have a comprehensive understanding of their variability. Thus, global empirical algorithms have been developed to estimate pH from more commonly measured parameters, like temperature, pressure, salinity, and oxygen, to increase the spatiotemporal availability of pH data. Unfortunately, these algorithms do not perform well in shallow coastal waters due to a lack of training data in these areas and regional specificity. So, we developed a fully automated empirical algorithm development framework that trains both a single neural network model and a neural network ensemble that, relative to previously developed algorithms, can more accurately estimate pH from time, location, temperature, pressure, salinity, and oxygen within the Central California region. To train our algorithm we used a robust training set using open-source in-situ pH measurements from autonomous glider deployments from MBARI alongside discrete shipboard measurements from both MBARI and NOAA. When evaluated on an independent testing dataset comprised of three glider deployments our best single model achieved a RMSE of 0.01074 and our best ensemble achieved a RMSE of 0.01052; which is better than our benchmark algorithm, CANYON-B, that achieved a RMSE of 0.02248.

1 Introduction

Increased acidification of the ocean, caused by the growing presence of human activity, has been shown to negatively affect a range of organisms and ecosystem. Ocean acidification is defined as the reduction of pH of the ocean over time primarily as a result of the absorption of atmospheric carbon dioxide [1]. As pH gets lower, carbonate ion concentrations also get lower which stress organisms like corals, oysters, clams, and some plankton which use carbonate ions to form their shells [2]. This then leads to negative effects on our economy, like in the case of oysters in the Pacific Northwest. It was found that lower saturation states–a measure associated with low pH–led to a sharply increased mortality rate in larval oysters from oyster farms in the Pacific Northwest, which is an industry valued at $US278 million [3]. Studies have also shown that ocean acidification has led to increased shell dissolution in pteropods, which are organisms at the base of the oceanic food web that are often used to assess ecosystem health. Areas with high rates of ocean acidification are thus less suitable habitats for pteropods, ultimately threatening the stability of the food web [4]. These findings have put ocean acidificaiton on the minds of people ranging from scientists to legislators–especially those in areas of rapid ocean acidification. One of these areas is the coast of California, wherein CO_2 rich water brought in from the deep ocean via natural upwelling combines with anthropogenic atmospheric CO_2 and results in an increased rate of ocean acidification [5, 4]. Given the size of California's economy and the importance of the coastal California waters, studying ocean acidification is imperative.

To study the effects of ocean acidification, large amounts of observational carbon measurements, especially pH, are needed. However, reliable pH sensors were only recently developed and have not yet been widely implemented enough to meet the needs of scientists trying to study ocean acidification and its effects. To fill this observational data gap, empirical algorithms have taken advantage of easier to measure ocean parameters such as temperature, pressure, salinity, and oxygen and their covariance with carbon parameters to estimate these carbon parameters. These empirical algorithms, such as CANYON-B and LIPHR, have used the GLODAPv2 dataset of historical shipboard ocean carbon measurements for training and testing. CANYON-B was a Bayesian neural network approach that extended off the CANYON neural network approach. For estimating pH CANYON-B achieved an RMSE of 0.013 and biases reported in the 10/50/90th percentiles as $-0.012/0.000/0.012$ when tested on a random sample of global pH data. These were improvements over CANYON which achieved an RMSE of 0.019 and the study suggests that some improvement can be associated with the additional data added in GLODAPv2 [6]. LIPHR was able to achieve an RMSE of 0.002 compared to CANYON which had an RMSE of 0.009 when reproducing the dataset LIPHR was tested on (Carter et al 2018). Both these models used the same GLODAPv2 dataset to achieve the same goal with similar accuracy and allow scientists studying pH to fill observational gaps in pH measurements. However, both models perform worse at the sur-

face and in coastal water than at depth in open oceans. This is primarily due to air-sea flux which makes the surface highly dynamic and difficult to accurately model [6, 7]. This proves problematic as surface and coastal waters are critical to study since they are the areas that humans interact with most.

To address the shortcomings of these global empirical algorithms, new algorithms have been developed following a similar structure to estimate carbon parameters of much smaller regions. In 2020, CANYON-MED was developed by Barton et al to better estimate carbon parameters in the Mediterranean Sea. The dynamic waters of the Mediterranean Sea prevented the CANYON-B empirical algorithm from accurately modeling the patterns in the correlation between temperature, pressure, salinity, and oxygen and the various carbon parameters of interest. However, after training the algorithm specifically on data from the Mediterranean Sea, the algorithm was then able to more accurately predict the carbon parameters than CANYON-B. In the case of our interest, CANYON-MED was able to estimate pH of their testing dataset with an RMSE/MAE of $0.016/0.010$ compared to CANYON-B and CANYON which had RMSE/MAE of $0.020/0.014$ and $0.026/0.018$, respectively [8]. This approach suggests that developing a regional algorithm with more local data while following the CANYON neural network structure can provide improved estimates of carbon parameters–especially pH.

Our interests lie in being able to study ocean acidification in the Central California region. This region's waters are very dynamic due to frequent upwelling, causing CANYON-B and LIPHR to perform poorly when predicting pH and other carbon parameters in the Monterey Bay region. Takeshita et al discuss the performance of empirical algorithms for estimating pH in this area and found that CANYON-B performed better than LIPHR. Then when assessing the accuracy of CANYON-B using shipboard data from MABRI C3PO and NOAA West Coast Ocean Acidification (WCOA) cruises, they found that below 200m, CANYON-B had agreement within ± 0.01 and between 350m and 550m, very close agreement was observed, ± 0.005. However, as stated in the CANYON-B and LIPHR papers, the algorithms performed poorly at depths down to 200m–agreement of only ± 0.04 [9]. So, after observing the success of CANYON-MED in developing a regional empirical algorithm, we set out to and developed another regional empirical algorithm for the Central California region to predict pH from temperature, pressure, salinity, and oxygen data. We focused on pH, however, the framework we developed should be able to be extended to other carbon parameters such as partial pressure of $CO_2(pCO_2)$, dissolved inorganic carbon (DIC), and total alkalinity (TALK). CANYON-B is a suite of multiple neural networks each designed, using a similar structure, to predict a different carbon parameter. So, our algorithm framework should extend beyond just pH, but this paper focuses only on estimating pH.

Past empirical algorithms have been improved through both structure and the training data used. We used datasets from three years of pH spray glider transects deployed by MBARI which provide hundreds of thousands of labeled datapoints (here we will use the term *datapoints* to de-

scribe an input vector containing all our described input parameters and an associated pH output value) within the Central California region to train our empirical algorithm. This glider data was accompanied by much smaller datasets of labeled shipboard datapoints from MBARI and NOAA West Coast Ocean Acidification cruises. These datasets provide a much better training set in terms of both volume and seasonal representation compared to the subset of the CANYON-B and LIPHR training sets of datapoints in the Central California region. Both CANYON-B and LIPHR used the GLODAPv2 [10] dataset to train their algorithms which provides a better yearly representation as it has cruise data from many years, but represents very few months and only has a couple thousand datapoints within our region. The use of autonomous glider data to help us develop a more accurate model highlights the opportunity that the future of autonomous data collection provides in terms of developing better algorithms.

In terms of structure, we follow the format of CANYON, CANYON-B, and CANYON-MED in developing a neural network for our estimating algorithm. We developed both a single neural network that outperforms CANYON-B as well as a ensemble of neural networks formed in a similar structure to the ensemble made in CANYON-MED that also outperforms CANYON-B. The process of preprocessing our inputs are slightly different than the process used in CANYON-B and the best neural network structure was similar to that found by CANYON, CANYON-B, and CANYON-MED. The rest of the paper will further describe the data we used and our quality checking process (Section 2.1); the preprocessing steps we took (Section 2.2); the neural network architecture (Section 2.3); the ensemble architecture (Section 2.4); our training and testing process (Section 2.2.1); a description of our automated pipeline (Section 2.6); the results of our neural network, our ensemble, and CANYON-B when tested on our Central California testing dataset (Section 3); a discussion of our results (section 4); our conclusions (section 5); and the next steps we plan to take with the project (Section 6).

2 Methods

Our methods were designed to train a neural network using autonomous glider observations from an easy-to-use and repeatable pipeline. We follow standard machine learning practices to prepare our data, train our model, and evaluate our model. The specifics of the data we used and each step of the development process are outlined below.

2.1 Data Sources and Quality Control

To develop our algorithm we used glider observations, MBARI C3PO cruise data, and NOAA West Coast Ocean Acidification (WCOA) cruise data. Each data source has its own format and quality control markers, so we had to individually quality check and read in each data format individually.

This initial process involved minor manual adjustments in addition to automated scripts for each data type to allow our development pipeline to compile all data types into one large training and one large testing dataset. The individual data sources and their quality checking are described in the following sections.

2.1.1 MBARI C3PO Cruises

We use data from three different MBARI C3PO cruises that collected pH measurements in the Central California region in May 2019, July 2019, and May 2021. From these files we were able to collect 748 complete datapoints with acceptable measurements for temperature, pressure, salinity, oxygen, and pH. These data files were fully quality checked by MBARI prior to our use, however, we performed our own range and spike tests. We used 7.3-8.5 as our pH range and the spike test marks the pH measurement V_2 as bad if the following is true,

$$|V_2 - median(V_0, V_1, V_2, V_3, V_4)| > 0.04$$

as recommended by Johnson et al (*equation* 23) [10]. We also checked all the quality flags for missing data points ensuring that all measurements of -999 are marked with a quality flag 8 for *missing data*. These quality checks were performed using code via a python script and the data with our updated quality checks were saved to a new file. We also manually removed all comments and empty lines in each csv data file before running the quality check script.

2.1.2 WCOA Cruises

Our final research ship cruise data file came from the 2016 WCOA cruise performed by NOAA. We were only able to use this one cruise because all other WCOA cruises uploaded do not have direct pH measurements. We avoided using calculated pH values from Dissolved Inorganic Carbon (DIC) and Total Alkalinity (TALK) measurements due to subtle differences in measured and calculated pH values. From this cruise we added 48 datapoints of acceptable measurements of our inputs and pH to our overall dataset. For quality checking we used the same range and spike tests and missing data checks as specified for the MBARI C3PO cruises and performed similar manual removals of comments from the data file before running the WCOA quality check script.

2.1.3 MBARI Spray Glider Deployments

Spray glider deployments were our main source of data for developing our algorithm. MBARI equipped spray gliders with pH sensors (CITATION???) and has collected pH measurements for the previous three years. Glider deployments extend perpendicularly out into Monterey Bay and the Pacific Ocean from MBARI and each deployment travels a different distance offshore. The

furthest deployments extend THISMANYKM kilometers offshore. After quality checks we added about 1 million datapoints of acceptable measurements to our overall dataset. The glider data is quality checked in real time as the data is sent back to MBARI during the deployments and then goes through some post-processing steps after the deployment. Further quality checks are thus necessary and we performed them following the same process as the ship data using a python script with the same range check, spike test, and missing value flag check. MBARI spray gliders follow the same data format as BGC-Float data files and so no manual data manipulation was required for our scripts to easily read in the data.

2.2 Data Preparation

Our initial quality checking and minor data manipulation was done so that our data preparation phase which creates training and testing datasets to develop our algorithm could be done easily and without any manual manipulation. After the quality checks, all data files are put into specified directories (folders on our computer), described in the next section, where they are read in by our code and fully processed as described in the following sections.

2.2.1 Training and Testing Data Separation

We separated our data files by deployments and cruises to create our separate training and testing datasets. To evaluate our algorithm we need to set aside a subset of our data that our algorithm will never be exposed to during the training phase. This subset of data is then used to evaluate the performance of our algorithm on data it has never seen before and is designated the testing set. All data not in this subset will be used to train our algorithm during the training phase and is designated the training set.

We placed all ship data in our training set as they are our highest quality measurements and we want our algorithm to learn from our best measurements. We then chose three deployments from different years and months to use as our testing dataset. The remaining six deployments were used in addition to the ship data to create the training set. We split training and testing sets by deployments and cruises and not random selection from the entire dataset due to the density of the glider data. Gliders collect data at such a high frequency that if randomly selected, any datapoint in the testing set will be very similar to an input in the training set. This violates the assumption that the testing set is independent from the training set and provides a fair evaluation of the algorithm after training.

We achieved this split by placing our quality checked training and testing data files in the specified *training and testing directories*. From there data files in both directories go through the same data cleaning and preprocessing pipeline. This pipeline is the same process any input to this algorithm would have to go through when being used to make a pH estimation and is outlined in

the next two sections.

2.2.2 Data Cleaning and Compilation

The first step in the cleaning and preprocessing section of our pipeline is to remove all datapoints marked as *bad* and then compile all remaining *acceptable* datapoints from each file into one dataset. This process is simple and requires checking the quality flags for each input (temperature, pressure, salinity, and oxygen) and the pH measurement. All measurements within a datapoint much be marked as *acceptable* for the datapoint to be kept in the dataset. Once all *bad* datapoints are removed from the data files, the *acceptable* datapoints are compiled into one large file where they are then shuffled. Shuffling the inputs helps the algorithm learn the desired mapping more effectively. When neural networks are trained, the internal weights and biases values are updated after a specified number of datapoints are given to it. By feeding the algorithm random datapoints we ensure a more representative sample of inputs are seen before the next weights and biases values update. Once this step is complete, the data is considered cleaned and ready to go through data preprocessing as the final preparation before the training phase.

2.2.3 Data Preprocessing

After the data is cleaned, the inputs go through a couple final transformations before they are ready to be used to train our algorithm. The date input transformation is the most important transformation in this step. Suazede et al included the day-of-year and year as separate date inputs to the CANYON algorithm while Bittig et al used the decimal year as the date input to CANYON-B [11, 6]. We follows Bittig et al and their CANYON-B approach and used the decimal year as our date input. This transformation converts the text date value in our data file into a numerical input that our algorithm can process. The decimal year was the chosen numerical transformation as it allows the algorithm to both track any yearly seasonal patterns and long term trends on the yearly scale. We then chose to not transform the location or pressure inputs. Prior algorithms used sine and cosine transformations for latitude and longitude to capture the spherical topography of the earth as they are global algorithms [6, 11]. Our algorithm has location inputs from a much smaller grid such that the linear magnitude changes of the non-transformed latitude and longitude inputs better represented our data and its patterns. Similarly, prior algorithms applied a transformation to the pressure inputs to account for the smaller rate of change in pressure the deeper you go in the ocean. Since our algorithm is mostly focused on accurate performance in shallow water and all our datapoints are from measurements taken above 1000m depth, this transformation had little impact on our data. So, of our seven inputs, we only transformed our date input to the decimal year despite other transformations being included in previously developed algorithms.

Since neural networks converge quicker and more effectively when given inputs in a small

range we did normalize all our data before feeding it into the neural network. This was performed using a normalization layer in our neural network. In this layer, inputs are divided by the standard deviation of that input and then added to the negative of the mean of that input so that the range of each input has a standard deviation of 1 and a mean of 0. This is performed individually on each input and the mean and standard deviation used is that of the training data inputs. All future inputs will be transformed via the mean and standard deviation of the training inputs. Although this preprocessing step is technically performed within the neural network as a normalization layer, we consider it a part of the preprocessing section as it is not involved is the actual estimation itself. Normalization is the final step in the cleaning and preprocessing pipeline before data is fed to the neural network for training, testing, or estimation.

2.3 Neural Network Architecture

We tested many different neural network structures and found the best architecture was a neural network with two hidden layers, 48 neurons in the first hidden layers, 24 neurons in the second hidden layer, and ReLU activation functions for each neuron in each hidden layer. This was the optimal sturcture for our testing set and fell into a pattern of our top performing models with all had two hidden layers with about 48-64 neurons in the first hidden layer and 16-32 neurons in the second hidden layer. More than two hidden layers and higher neuron counts in each hidden layer always led to overfitting of the training data and poor performance on the testing dataset. This structure agreed with the results of CANYON and CANYON-B which each found that the best neural network structure had two hidden layers with more neurons in the first hidden layer than the second [11, 6]. Previous algorithms used the tanh activiation function whereas our algorithm used the ReLU activation function. The tanh activation function allows neural networks to more easily learn nonlinear mappings, but in our case, it appeared to lead to overfitting with the models we trained. Future work would test the impact of different activation functions on the generalizability of neural networks for estimating pH. For all results explained in the paper, we refer to our aforementioned best trained model with two hidden layers with 48 and 24 neurons, repectively, and ReLU activation functions.

2.4 Ensemble Architecture

We were able to train 27 individual neural networks that outperformed CANYON-B on our testing dataset, so, we took our five best models and constructed an ensemble of neural networks to further improve our estimating performance. An ensemble works but feeding a single input to multiple models and taking the weighted average of the models' estimations and using that as the final estimate. For our ensemble we used weights of 0.4,0.2,0.2,0.15, and 0.15 for our best model through 5th best model in that order. The choice of 5 models was arbitrary as were the weights.

Further testing would be required to more surely claim the best ensemble architecture. Only basic testing was performed to test the potential for an ensemble to improve performance over that of our best single model.

2.5 Training and Testing Phases

To construct and train our algorithm we used the TensorFlow's Keras deep learing library for Python. Models were constructed as described in the previous *Architecture* sections using the *Keras.Sequential* model framework and trained using the built in **.fit()** method. Each model was trained for 50 epochs (number of passes through the entire dataset) using the default batch size (number of datapoints estimated while training before weights and biases values are updated). Training is performed on the training dataset built from the data files in the training directory after they go through the data cleaning and preprocessing pipeline. Testing is done automatically for every model that is trained. Each model estimates the entire dataset as well as the subset of the training set consisting of all pH measurements from depths less than 200m. Error metrics and visualizations are calculated and created from there.

2.6 Automated Pipeline and Transferable Models

Given data in the correct format our pipeline can clean the data, preprocess the data, train a model, and evaluate that model. This pipeline allows for multiple models to be trained easily, but with flexibility. The structure of the model and exact inputs to the model are determined by a series of variables at the top of the Python code that runs the pipeline. This automation means with basic knowledge of deep learning, one could quality check data, populate the training and testing directories, and develop a model to estimate pH based on their own data. The models developed are saved via the specified TensorFlow format which stores the weights for each neuron within the model. This means models trained can be transferred between the Python, MatLab, and R coding languages (and any other coding language with a deep learning library compatible with TensorFlow). This flexibility further allows more scientists to be able to utilize neural networks in their research.

3 Results

Our best ensemble and our best single model were able to estimate the testing set with less than half the error as CANYON-B when estimating the same dataset. The testing set consisted of three separate glider deployments and we evaluated a model's performance estimating this testing set using the standard regression error metrics mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE). Each error metric evaluates the model slightly differently,

however, the error metrics agree almost all time when comparing any two models so we focused in on RMSE as our evaluating metric and the rest of the paper will use RMSE as the primary comparison even though all error metrics will be presented. After evaluating each model on the entire testing dataset, we also calculated the error metrics for the model when only estimating the pH measurements from 0m to 200m depth. To better be able to evaluate the algorithms we developed, we evaluated CANYON-B estimating the testing dataset and compared each model and ensemble we constructed against the performance of CANYON-B. We chose CANYON-B as our benchmark as it outperforms CANYON, was identified by Takeshita et al as the best empirical algorithm for estimating pH in the Central California region, and is the model we modeled the foundation of our approach off of. In the next three sections we will present the results of the CANYON-B, our best single model, and our best ensemble on the testing set.

3.1 CANYON-B Performance

When estimating the entire testing dataset, CANYON-B has a MAE of 0.01694, a MSE of 0.00051, and a RMSE of 0.02248 (Table 2). This error comes from a consistent nonlinear underestimation of pH at all depths, but especially at depths less than 300m as highlighted by figures 7 and 8. The error increases when CANYON-B is used to estimate just the testing datapoints from depths less than 200m. When estimating these datapoints CANYON-B has a MAE of 0.02444, a MSE of 0.00089, and a S-RMSE of 0.02983 as shown in table 3. Figure 11 shows the bias in estimations from CANYON-B at shallow depths and the increased scatter towards the surface. This plot also shows tails extending down from 500m to 1000m with increasing error. This same data can be visualized as a 2D-histogram, figure 9, which shows the nonlinear underestimation. Deeper than 300m CANYON-B has a slight underestimation, but the error remains low. However, when zooming in on the top 200m as in figure 12 you can see the hook shape that forms from the majority of estimations by CANYON-B. The 2D-histogram also shows the low variance in CANYON-B estimates. At depths less than 75m we see scatter, but otherwise the estimates from similar inputs result in similar outputs. CANYON-B performed decently when estimating our testing set, but with clear area for improvement. These results formed the basis for evaluating our best single model and ensemble in the next two sections.

3.2 Trained Models

We were able to train 27 different neural networks that performed better than CANYON-B when estimating our testing dataset. Each of these models are a single neural network and not an ensemble. The error metrics 10 best models are shown in table 1 along with CANYON-B (Note this table is sorted by RMSE error calculated after estimating the shallow test dataset). Models are labeled by their sturcture in the format (Neurons 1, Neurons 2, ...) where each number within the

parenthesis represents the number of neurons in that hidden layer. So, our best model which has 48 neurons in the first hidden layer and 24 neurons in the second hidden layer would be written as (48,24). The models with the 10 lowest error metrics all have two hidden layers with the number of neurons in the first hidden layer being greater than or equal to the number of neurons in the second hidden layer. Some models with three hidden layers were developed and performed better than CANYON-B but worse than models with only two hidden layers. These models had lower error metrics when estimating the training dataset which points toward overfitting as the source of their worse performance on the testing set. The results from the best single model we trained are highlighted in the following section.

3.3 Best Single Model Performance

Our best single model performs more than twice as well as CANYON-B when estimating both the entire dataset and the datapoints from less than 200m depth. For the entire dataset, our best single model has a MAE of 0.00822, a MSE of 0.00012, and a RMSE of 0.01074–a reduction of 0.00872 0.00039, and .01174 in MAE, MSE, and RMSE, respectively, when compared to the performance of CANYON-B (Table 2. The reduction in error metrics is even greater at shallow depths as our model has a MAE of 0.01009, 0.01435 less than CANYON-B; a MSE of 0.00017, 0.00072 less than CANYON-B; and a RMSE of 0.01314, 0.01669 less than CANYON-B (Table 3. Unlike CANYON-B, our follows no clear bias as shown in figures 7 and 10 and we see a stronger correlation at lower pH values roughly between 7.6 and 7.85 than at higher pH values, roughly above 8. This pattern is similar to CANYON-B which has more scatter at a higher pH than at lower pH measurements. When focusing in on the top 200m of datapoints, we see our model has more overestimations than underestimations, but there is an even scatter in either direction of 0 of about $\pm 0.05 \Delta pH$ near 0m and $\pm 0.025 \Delta pH$ near 200m (Figure 11). Figures 9 and 12 are 2D-histgrams that show where the majority of datapoints lie on the error vs depth figures. Between 100m and 200m our model performs very well and most of the estimations have an error of 0. Above 100m and between 200m and 500m, our model mostly overestimates pH, but by less than 0.025. We can also see the scatter is quite low at the surface and those measurements with larger errors are more of outliers than significant patterns.

3.4 Best Ensemble Performance

Using the many models we developed that perform better than CANYON-B, we created an ensemble that was able to outperform our best single model in terms of accuracy and variance. When estimating the entire dataset, our best ensemble had a MAE of 0.00809, 0.00013 less than our best single model; a MSE of 0.00011, 0.00001 less than our best single model; and a RMSE of 0.01052, 0.00012 less than our best single model (4). Then for the shallow dataset, the ensem-

ble had a MAE of 0.00989, a MSE of 0.00016, and a RMSE of 0.01282–reductions of 0.00020, 0.00001, and 0.00032, repectively (5). The error vs estimation plots (figures 13 and 16) and error vs depth plots (figures 14 and 17) for our ensemble look very similar to those of our single model (figures 7 and 8). This follows from our error metrics which are very similar, but tells us that the ensemble tracks the same patterns as the single model, just with slightly more accuracy. The 2D-histograms, however, show decreased variance in the estimations. In figures 15 and 17 we can see more estimations fall within the same 2D-histogram bins. Note the scale on figures15 and 17 reach above 600 and 140 estimations per bin while figures 9 and 12 only reach about 500 and 140 estimations per bin. So similar inputs given to the ensemble are more likely to have similar pH estimates provided more certainty in the estimates given by the ensemble than our single best model.

4 Discussion

Using glider data collected over the last three years, we were able to develop many individual neural network models and neural network ensembles to more accurately estimate pH in the Central California region. This supports the idea that autonomous data has high potential for improving the development of algorithms to aid in oceanography research. Our models outperformed the previously used algorithm for Central California, CANYON-B, by reducing the nonlinear bias in estimations rather than decreasing variance in estimates. This suggests our increase in training data was the main driver of accuracy improvements rather than our structure better learning the covariance between temperature, pressure, salinity, and oxygen and pH.

4.1 Improvements of Best Model Over CANYON-B

Our best single model was able to perform noticeably better than CANYON-B across all depths but especially for pH measurements less than 200m. Since the internals of neural networks can be complicated to assess, it is hard to determine exactly why our model was able to better map our inputs to desired pH measurements. However, data limitation and neural network structure play into the source of more accurate estimates and we will discuss that further in the following sections.

4.1.1 Data Limitation

Narrowing the range of the location inputs of our algorithm to Central California means that we had hundreds of thousands of datapoints more than CANYON-B within the region (figures 2 and 3). This is the main difference between the training process of CANYON-B and our model as the neural network structure and input transformations were otherwise very similar. Analyzing the

results of CANYON-B when estimating the testing set, CANYON-B showed a consistent underestimation of pH values at the higher end of our pH range. The increase in error for pH measurements taken closer to the surface was not surprising, but the consistent underestimation was. It is important to note that the underestimation was nonlinear and not exactly a systematic bias as shown by the 2D-histograms (figures 9 and 12). The 2D-histograms also show the low variance in estimations from CANYON-B. For each depth, CANYON-B estimates tend to clump together and form high estimation per bin values which is evidence that similar inputs are more likely to lead to similar outputs through CANYON-B. This is expected as CANYON-B is actually comprised of a large variable ensemble that usually includes between 20 and 30 different models making estimations. These factors suggest that the nonlinear bias error comes from external factors causing bias captured by time and location inputs rather than failure to properly learn the mapping from temperature, pressure, salinity, and oxygen to pH. Thus, narrowing the focus of our model to only the Central California region and having many more datapoints to train the algorithm on, allows our model to better extract these external biases from the time and location inputs. On a global scale, it is harder to represent that many biases through three time and location inputs whereas on our little 10°x10° grid in Central California and with data with good seasonal and yearly representation, our model can learn and correct for these biases. Claiming that these external biases need to be captured in the time and location inputs does create a difficult task for the model as there could be many factors influencing seasonal biases in surface pH measurements. So, moving forward, additional ocean parameters may be added as inputs to try and address this missing information and improve the ability of empirical algorithms to capture these biases external to temperature, pressure, salinity, and oxygen.

4.1.2 Structure

The structure of our model follows closely to that of CANYON-B, but with enough changes to have some impact on our results. We performed the same date transformation as CANYON-B, but excluded location and pressure transformations. These transformations were more necessary for CANYON-B as they had larger ranges of location and pressure inputs. Had we transformed our inputs, the differences between our datapoints would have decreased even more than it already was since all of our inputs where from a small dense range. Therefore, excluding these transformations likely improved our model as it allowed our model to more easily differentiate between each datapoint input. The most influential structural change came from the use of the rectified linear unit (ReLU) activation function rather than the hyperbolic tangent (tanh) activation function. The tanh activation function is nonlinear and usually allows for neural networks to better map nonlinear functions, however, in our models it often times lead to overfitting. The ReLU activation function is a linear function and the simplest activation function (beyond the identity function) used for neural networks. The influence of the activation function on the the neural network performance has

not been thoroughly tested at this point and will be explored in more depth moving forward. The only significant difference in structure being in the activation function is evidence that the structure of our neural network had less to do with the improved performance of our model compared to the presence of more data.

4.2 Improvements of our Ensemble Over Best Model

Our ensemble had minor improvements–changes in the fourth decimal place–over our best single model in terms of error metrics but achieved less variance in its estimates compared to our single best model. The structure of each individual model and the data they were trained on were all the same, attributing all the improvement to the way the ensemble functions. By taking the weighted average of five models, minor changes in inputs have less influence on the final estimation as the minor changes are likely to push the estimations of the different models in different directions unless the changes are significant to the mapping. The weighted aspect of the average allows for the best models to have more influence over the final estimate as they have proven to estimate pH the best on our testing dataset. By weighting the best model more than the rest, we can better ensure there will be less sacrifice in accuracy for a gain in certainty in our estimates. The improvement of CANYON-B over CANYON came primarily from its development as an ensemble and CANYON-MED followed the structure of CANYON-B and developed an ensemble to improve their estimates whose structure we closely mimicked ([6, 8]). To develop a better ensemble, more models should be trained with similar structures to the models currently included in our ensemble. These models can then be added to the ensemble to further decrease variance in estimates while continuing to prevent loss in accuracy. A research ready algorithm framework will certainly include an ensemble structure for final estimations.

4.3 Impact of Using Glider Observations

The use of glider observations to construct the bulk of our training dataset allowed us to develop an algorithm that performs better than CANYON-B in the Central California region. Section 4.1.1, Data Limitation, highlights the evidence that the training data was more influential in creating a more accurate algorithm than the structure of our algorithm. This is important to note as it presents the opportunity to develop many more regional algorithms in areas of high research interest for various ocean parameters. As autonomously collected data becomes more prevalent through gliders, floats, etc., we have the opportunity to develop more accurate regional algorithms for predicting ocean parameters. Using autonomously collected data does come with some caveats in terms training machine learning algorithms. For example, the high density of the data prevents us from randomly selecting testing datapoints as they are not truly independent of the training data. However, this can be overcome by splitting sets by deployment or in disjoint large chunks of the

overall dataset. Further, using the glider data to train and test our algorithm we noticed tails in our error vs depth scatter plots that extend down from 500m (figure 8). Each tail is either the estimation of one ascent within a glider deployment or a conglomerate of multiple ascents. Sensors are calibrated for specific conditions, including depth, before deployments and it is important to note when using data from these deployments which conditions the sensors perform best at. In our case, it may be best to only use glider data from the top 500m. This increases the importance of careful quality checking done before training and testing any models. Despite the need to be cognizant of these tendencies when using autonomously collected data to develop algorithms, autonomous data provides a huge opportunity to develop better algorithms moving forward as having enough good measurements is such a pivotal part of the development process.

5 Conclusions

We were able to use glider observations to train a neural network to more accurately predict pH in the Central California region than previously developed empirical algorithms designed to predict ocean pH. The success of using glider observations shows the power that autonomous data has to push ocean science further ahead, especially in the realm of algorithm development. The use of glider observations within a localized region, in addition to prior work like CANYON-MED, further shows that we can outperform global algorithms given enough data within that localized region. And our ability to create an automated pipeline to develop this algorithm also presents the idea that these very specific regional algorithms can be accessible to any scientist regardless of machine learning background or knowledge. So, we hope that this project can be a step in the direction of providing access to neural network based empirical estimation algorithms to any scientist who needs them to do their research.

6 Future Work

After seeing the impact of using glider observations to train a better algorithm and our ability to create an automated pipeline to train and test a neural network model to estimate pH, we hope to further develop our pipeline to allow any researcher to develop a similar algorithm using their own data. As mentioned in the previous section, autonomous data and open-access provides the opportunity to improve on previously developed algorithms. By creating an easy-to-use framework accessible by anyone, any researcher can develop their own algorithm for their own specific use. To achieve this goal we hope to create similar frameworks for other ocean parameters and then combine the frameworks to one user interface where given user inputted training and testing data, a neural network would be trained and evaluate and save for the user to use later. The goal is for a researcher at the graduate level and above could use this framework. With large amounts

of ocean data becoming more readily available, it is important that tools like these are developed so researchers can take advantage of the power of tools like machine learning to advance our understanding.

7 Acknowledgements

I would like acknowledge the work of my mentor Yui Takeshita for his help on this project and with general assistance learning about becoming a researcher. I would also like to thank George Matsumoto, Megan Bassett, and Lyndsey Claassen for their work running the MBARI summer internship program as well as MBARI for the internship funding and opportunity this summer. I'd lastly like to thank the California State University - Monterey Bay REU for my personal funding this summer and the professional development and growth opportunities they gave me this summer while completing my project.

References

[1] N. O. a. A. A. US Department of Commerce, "What is Ocean Acidification?."

[2] N. O. a. A. A. US Department of Commerce, "Ocean Acidification: Saturation State Dataset | Science On a Sphere."

[3] A. Barton, B. Hales, G. G. Waldbusser, C. Langdon, and R. A. Feely, "The Pacific oyster, Crassostrea gigas, shows negative correlation to naturally elevated carbon dioxide levels: Implications for near-term ocean acidification effects," *Limnology and Oceanography*, vol. 57, no. 3, pp. 698–710, 2012. _eprint: https://aslopubs.onlinelibrary.wiley.com/doi/pdf/10.4319/lo.2012.57.3.0698.

[4] N. Bednaršek, R. A. Feely, J. C. P. Reum, B. Peterson, J. Menkel, S. R. Alin, and B. Hales, "Limacina helicina shell dissolution as an indicator of declining habitat suitability owing to ocean acidification in the California Current Ecosystem," *Proceedings of the Royal Society B: Biological Sciences*, vol. 281, p. 20140123, June 2014. Publisher: Royal Society.

[5] N. Gruber, C. Hauri, Z. Lachkar, D. Loher, T. Frölicher, and G.-K. Plattner, "Rapid Progression of Ocean Acidification in the California Current System," *Science (New York, N.Y.)*, vol. 337, pp. 220–3, June 2012.

[6] H. C. Bittig, T. Steinhoff, H. Claustre, B. Fiedler, N. L. Williams, R. Sauzède, A. Körtzinger, and J.-P. Gattuso, "An Alternative to Static Climatologies: Robust Estimation of Open Ocean CO2 Variables and Nutrient Concentrations From T, S, and O2 Data Using Bayesian Neural Networks," *Frontiers in Marine Science*, vol. 5, 2018. Publisher: Frontiers.

[7] B. R. Carter, R. A. Feely, N. L. Williams, A. G. Dickson, M. B. Fong, and Y. Takeshita, "Updated methods for global locally interpolated estimation of alkalinity, pH, and nitrate," *Limnology and Oceanography: Methods*, vol. 16, no. 2, pp. 119–131, 2018. _eprint: https://aslopubs.onlinelibrary.wiley.com/doi/pdf/10.1002/lom3.10232.

[8] M. Fourrier, L. Coppola, H. Claustre, F. D'Ortenzio, R. Sauzède, and J.-P. Gattuso, "A Regional Neural Network Approach to Estimate Water-Column Nutrient Concentrations and Carbonate System Variables in the Mediterranean Sea: CANYON-MED," *Frontiers in Marine Science*, vol. 7, 2020. Publisher: Frontiers.

[9] Y. Takeshita, B. D. Jones, K. S. Johnson, F. P. Chavez, D. L. Rudnick, M. Blum, K. Conner, S. Jensen, J. S. Long, T. Maughan, K. L. Mertz, J. T. Sherman, and J. K. Warren, "Accurate pH and O2 Measurements from Spray Underwater Gliders," *Journal of Atmospheric and Oceanic Technology*, vol. 38, pp. 181–195, Feb. 2021. Publisher: American Meteorological Society Section: Journal of Atmospheric and Oceanic Technology.

[10] K. S. Johnson, J. N. Plant, and T. L. Maurer, *Processing BGC-Argo pH data at the DAC level*. Ifremer, 2018. Medium: pdf Version Number: 1.0.

[11] R. Sauzède, H. C. Bittig, H. Claustre, O. Pasqueron de Fommervault, J.-P. Gattuso, L. Legendre, and K. S. Johnson, "Estimates of Water-Column Nutrient Concentrations and Carbonate System Parameters in the Global Ocean: A Novel Approach Based on Neural Networks," *Frontiers in Marine Science*, vol. 4, 2017. Publisher: Frontiers.

8 Tables

List of Tables

Error Metrics - 10 Best Trained Models & Canyon-B

Model	MAE	RMSE	S-MAE	S-RMSE
(48,24)	0.00822	0.01074	0.01009	0.01314
(32,32)	0.0092	0.01202	0.01023	0.01385
(64,48)	0.00858	0.01128	0.01039	0.01397
(64,64)	0.00881	0.01135	0.1112	0.01407
(48,32)	0.01003	0.01251	0.01119	0.01422
(64,32)	0.00915	0.01239	0.01117	0.01514
(24,24)	0.01182	0.01418	0.01222	0.01519
(64,16)	0.00946	0.01262	0.01129	0.01523
(48,42)	0.00801	0.01174	0.01096	0.01535
(32,16)	0.01132	0.01429	0.01156	0.01552
CANYON-B	0.01694	0.02248	0.02444	0.02983

Table (1) *Error metrics from the ten best neural networks trained when estimating the testing set, plus CANYON-B. Models are labeled by structure where each number within the parenthesis represents the number of neurons in that hidden layer. "s-" indicates the error metric for the model when only datapoints from less than 200m are considered.*

Our Model Error Metrics

Model	MAE	MSE	RMSE
Our Model	0.00822	0.00012	0.01074
CANYON-B	0.01694	0.00051	0.02248

Table (2) *Error metrics for our best single neural network and CANYON-B when estimating the entire testing set. Our best model has 48 neurons in the first hidden layer and 24 neurons in the second hidden layer with ReLU activation functions.*

Our Model Error Metrics (Depths <200m)

Model	MAE	MSE	RMSE
Our Model	0.01009	0.00017	0.01314
CANYON-B	0.02444	0.00089	0.02983

Table (3) *Error metrics for our best single neural network and CANYON-B when estimating the shallow datapoints (defined as datapoints with measurements from less than 200m deep) of our testing set. Our best model has 48 neurons in the first hidden layer and 24 neurons in the second hidden layer with ReLU activation functions.*

Our Ensemble Error Metrics

Model	MAE	MSE	RMSE
Our Model	0.00822	0.00012	0.01074
Our Ensemble	0.00809	0.00011	0.01052
CANYON-B	0.01694	0.00051	0.02248

Table (4) *Error metrics for our best ensemble and CANYON-B when estimating the entire testing set. Our best ensemble consists of our five best models weighted as 0.4, 0.2, 0.2, 0.15, 0.15, respectively.*

Our Ensemble Error Metrics (Depths <200m)

Model	MAE	MSE	RMSE
Our Model	0.01009	0.00017	0.01314
Our Ensemble	0.00989	0.00016	0.01282
CANYON-B	0.02444	0.00089	0.02983

Table (5) *Error metrics for our best ensemble and CANYON-B when estimating the shallow datapoints (defined as datapoints with measurements from less than 200m deep) of our testing set. Our best ensemble consists of our five best models weighted as 0.4, 0.2, 0.2, 0.15, 0.15, respectively.*

9 Figures

List of Figures

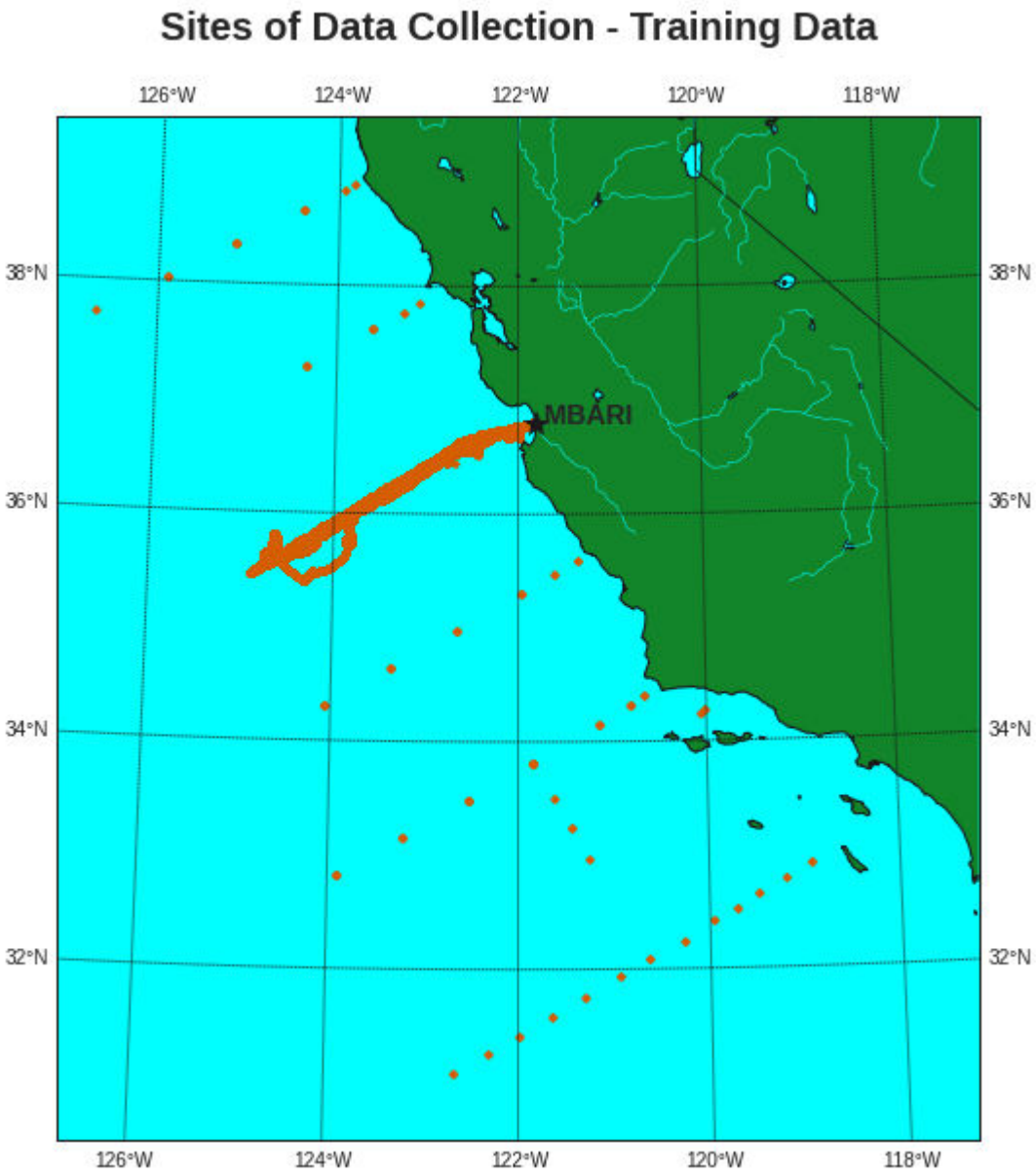

Figure (1) *Spatial Visualization of the training dataset for our models. Each orange dot represents a location where we have data from. The thick orange line formed extending out of Monterey Bay from MBARI is the site of glider transects. The rest of the sparse orange dots show measurements from MBARI and NOAA research ship cruises.*

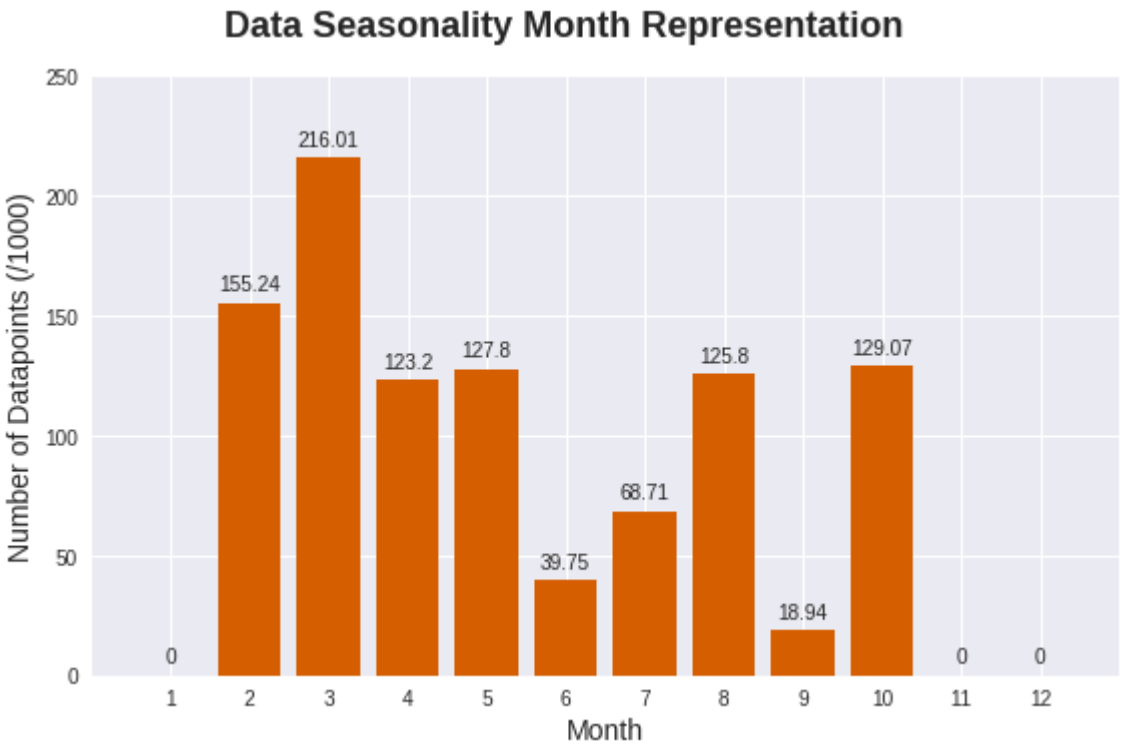

Figure (2) *Temporal visualization of the training dataset for our models. Each bar represents the number of datapoints from that month, scaled in 1000s. We have excellent seasonal representation in terms of both volume and number of months represented.*

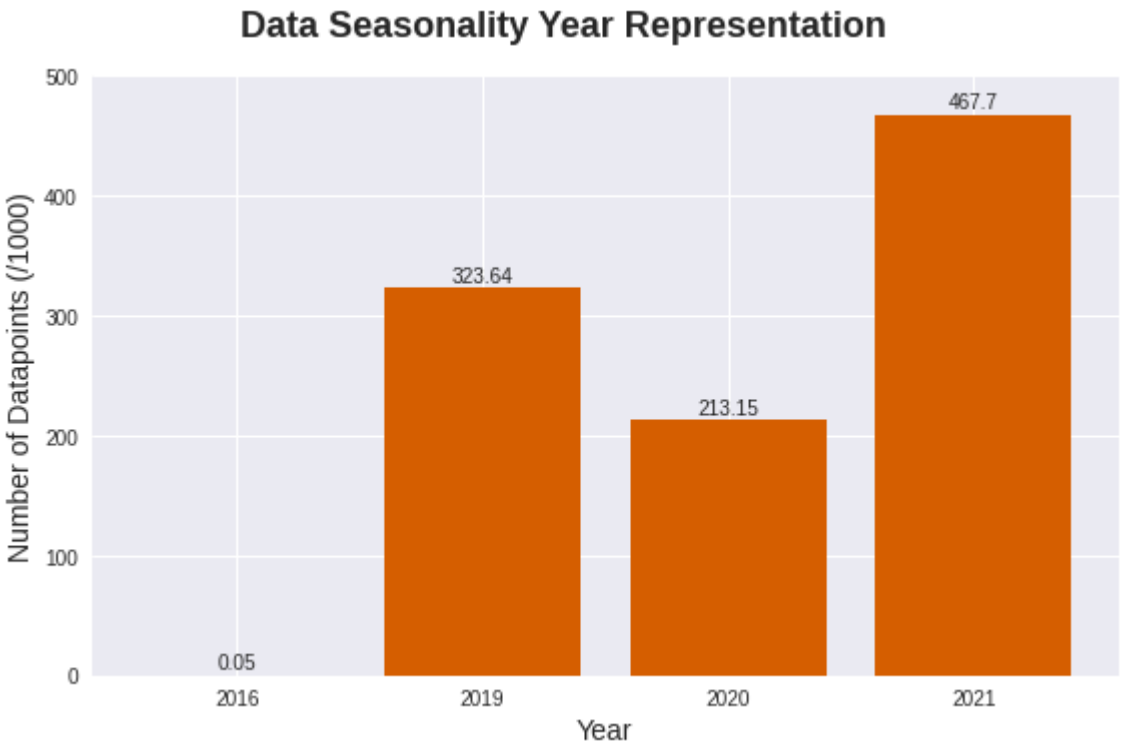

Figure (3) *Temporal visualization of the training dataset for our models. Each bar represents the number of datapoints from that year, scaled in 1000s. We have good yearly representation in terms of number of years represented and excellent representation in terms of volume.*

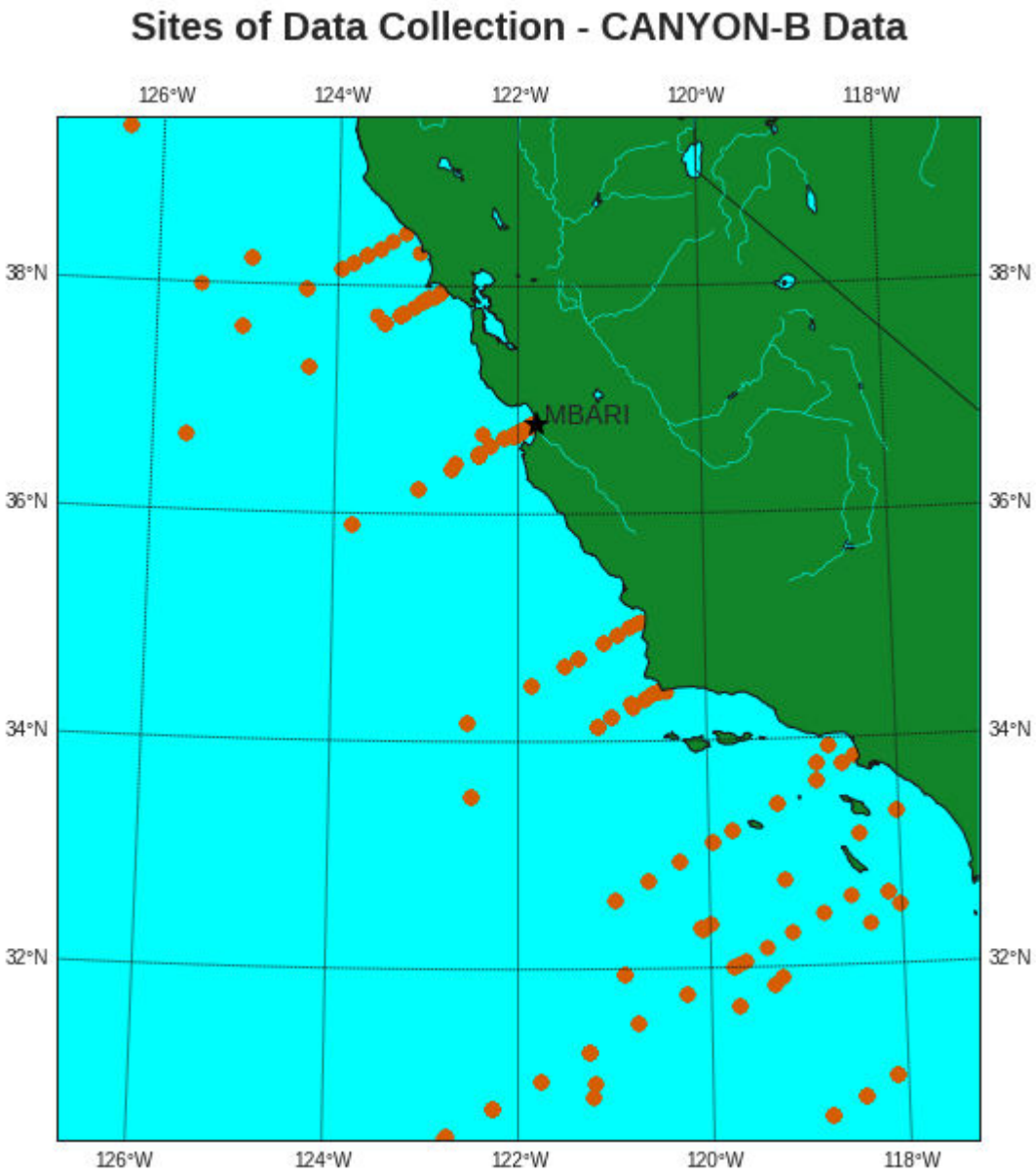

Figure (4) *Spatial Visualization of the training dataset used for CANYON-B from the GLODAPv2 dataset. Each orange dot represents a location where there is GLODAPv2 data with both DIC and TALK measurements. CANYON-B used calculated pH values from DIC and TALK as that provided more pH values than direct pH measurements themselves.*

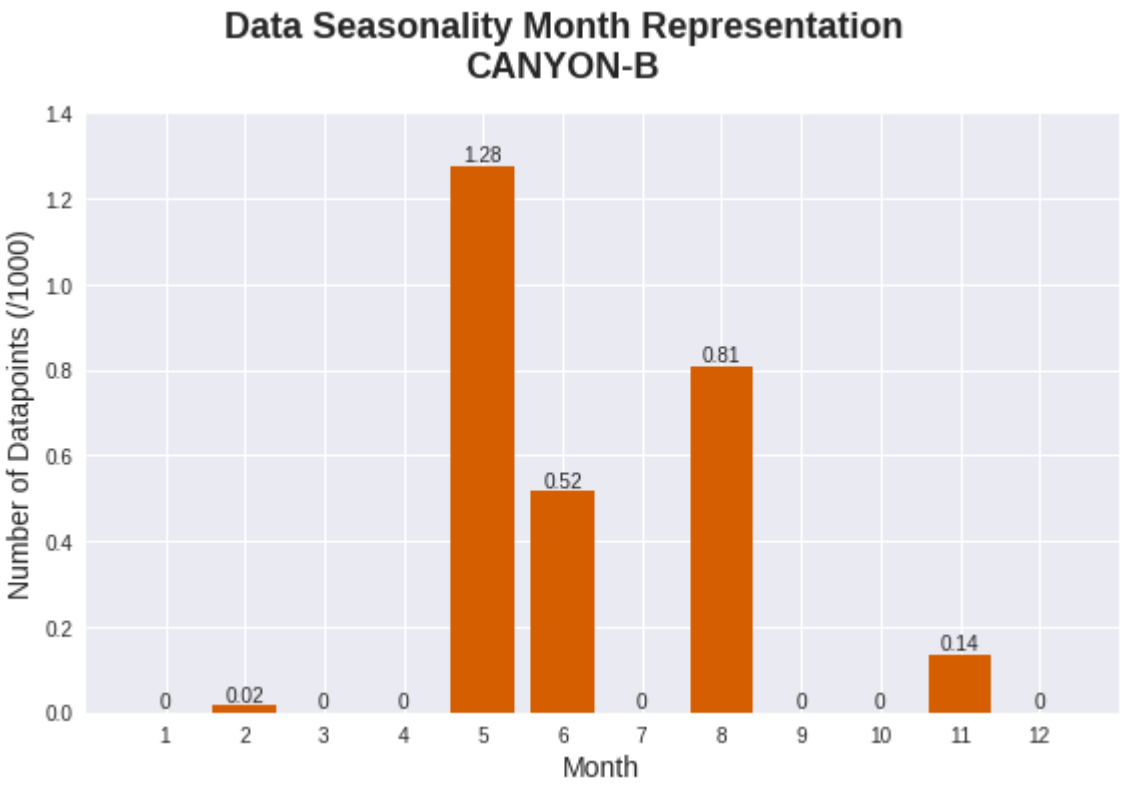

Figure (5) *Temporal visualization of the training dataset used for CANYON-B from the GLODAPv2 dataset. Each bar represents the number of datapoints from that month, scaled in 1000s. Within the Central California region, the CANYON-B training dataset does not have good monthly representation with only a couple thousands datapoints and 5 months represented (two of which have very little representation).*

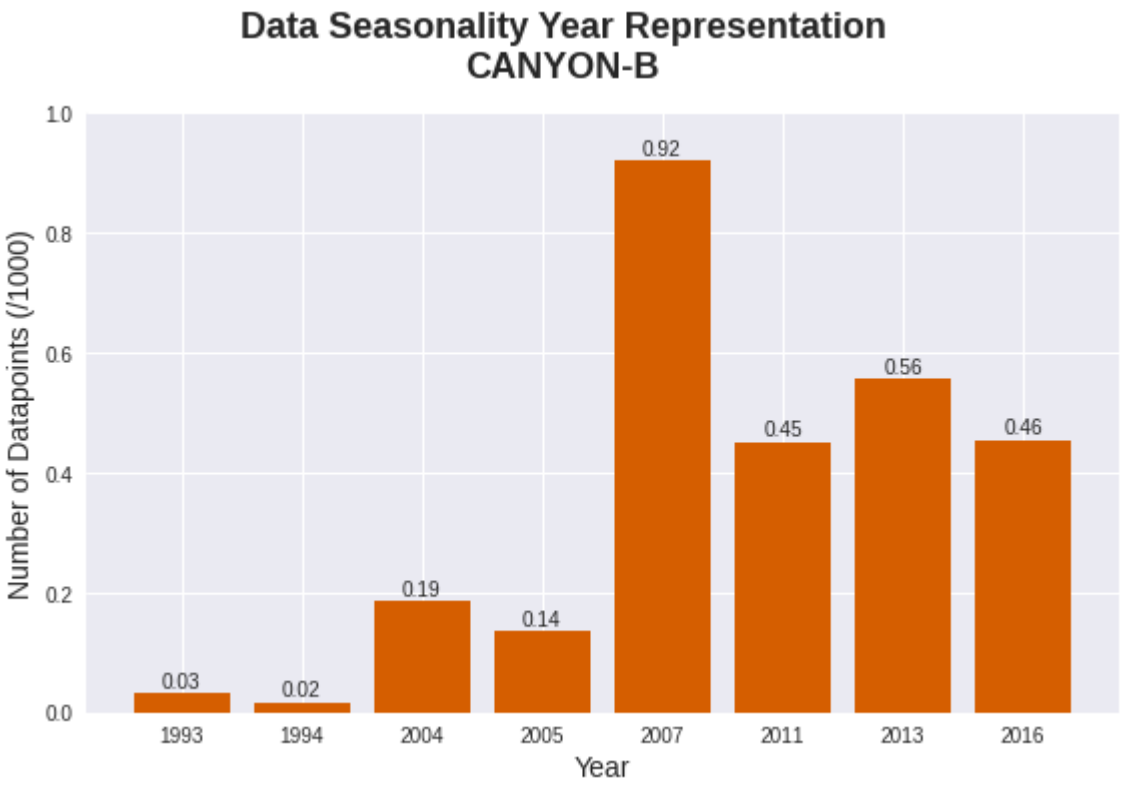

Figure (6) *Temporal visualization of the training dataset used for CANYON-B from the GLODAPv2 dataset. Each bar represents the number of datapoints from that year, scaled in 1000s. Within the Central California region, the CANYON-B training dataset has good yearly representation with datapoints from 1993 to 2016. However, it still only has a couple thousand datapoints so the representation in terms of volume is weak.*

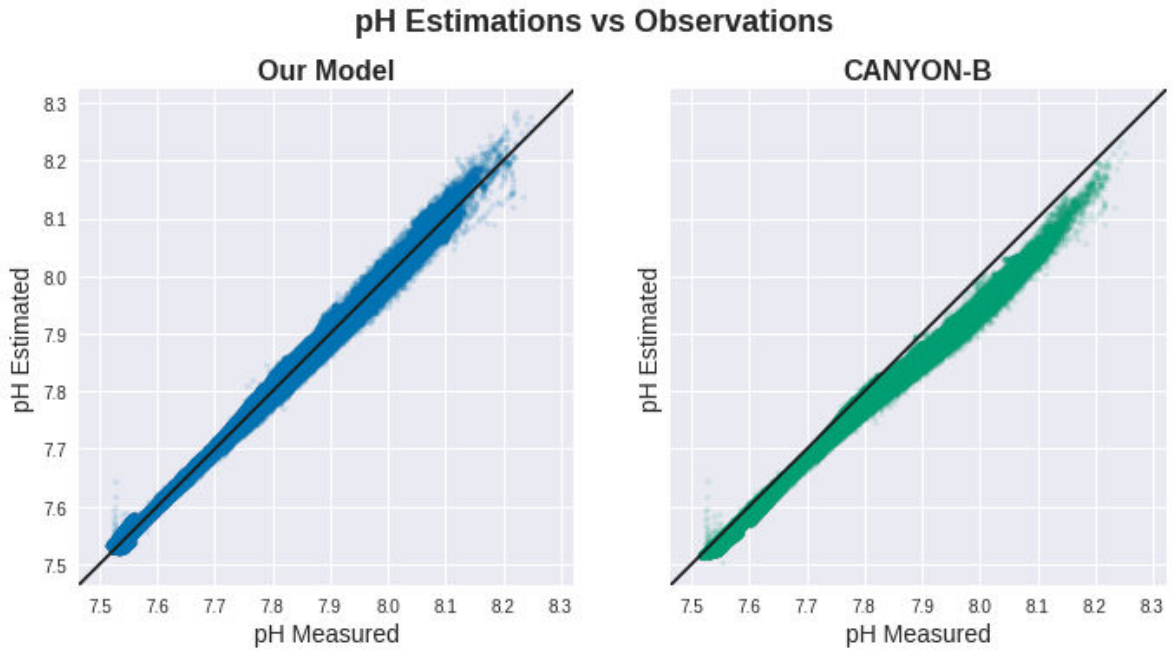

Figure (7) *This figure plots the pH estimations from our model and CANYON-B vs the measured pH for the same inputs. A 1-1 line is plotted to show where "perfect" estimations lie. This figure shows that our model is better able to reduce bias in its estimations compared to CANYON-B. It can also be seen that both models have more accuracy and less variance in their estimations at lower pH values than higher pH values.*

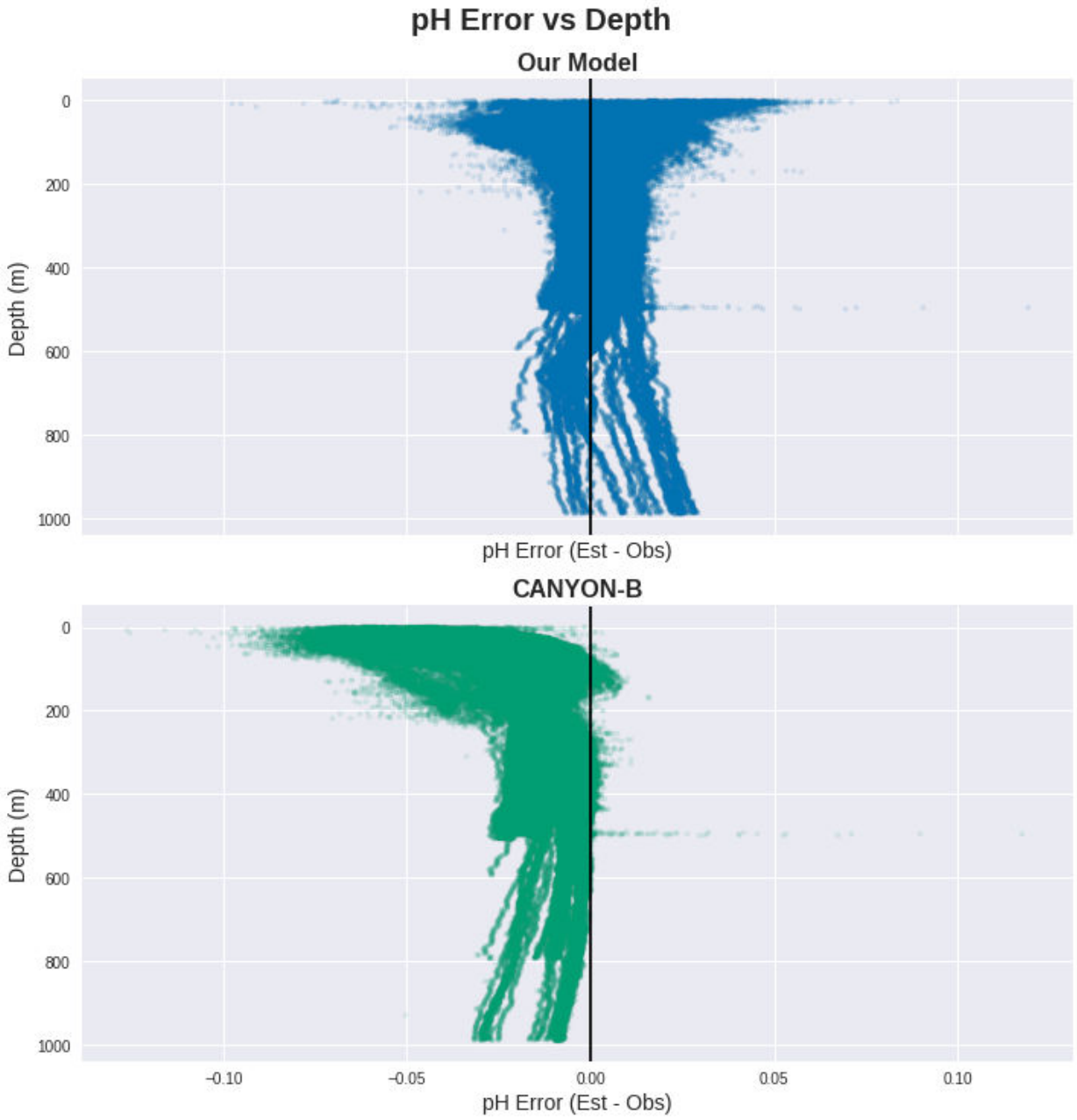

Figure (8) *This figure plots error vs the depth of the measurements for both our best single model and CANYON-B. In both plots, increased scatter and error can be seen at shallower depths. Our model stays centered on the zero-error line whereas CANYON-B hooks toward underestimations at around 200m. In both plots tails extending down from 500m are visible and each tail is created from estimations of ascents from glider deployments. The tails in CANYON-B clump together more because its ensemble structure results in less variance in its estimations than our single neural network.*

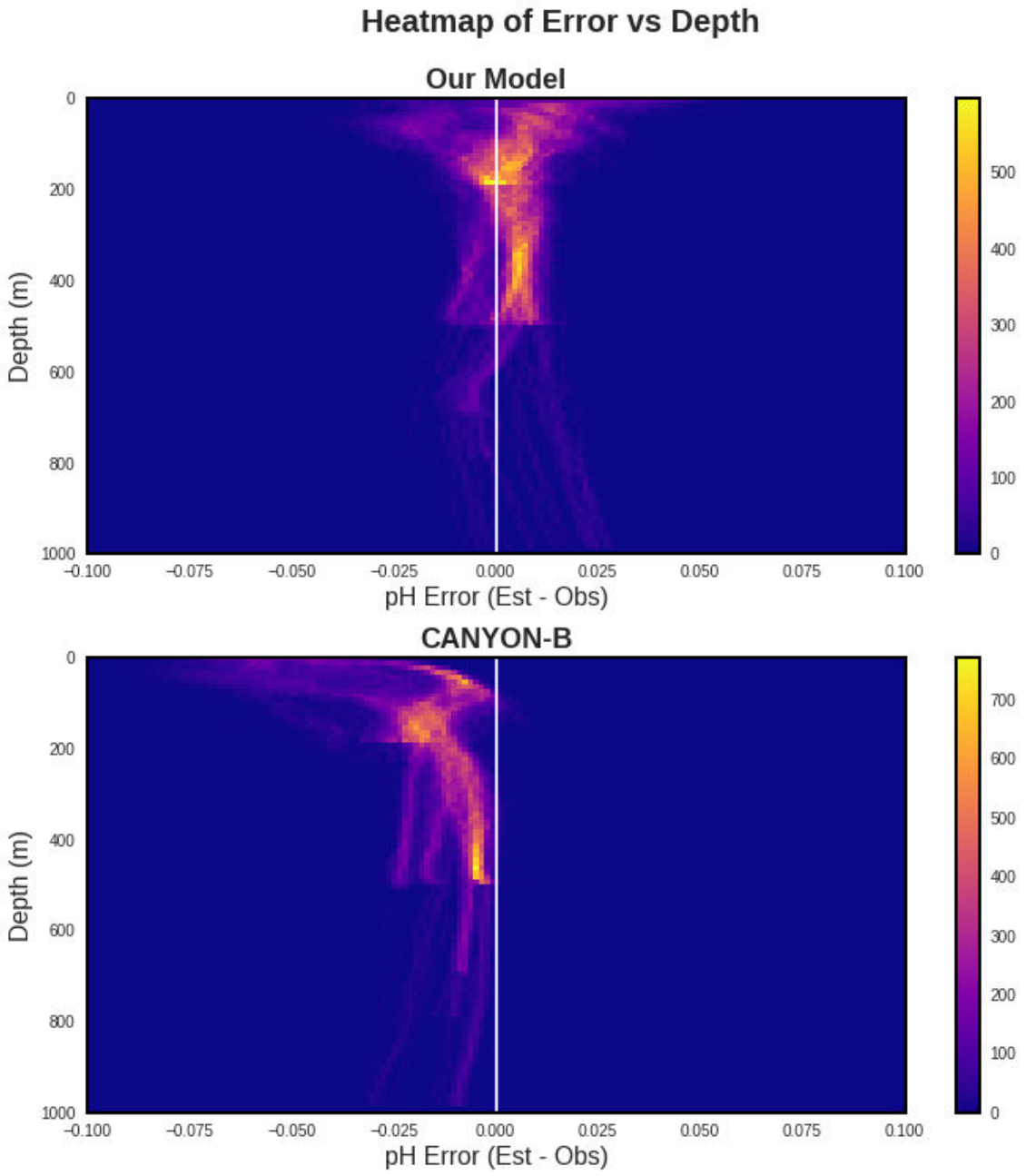

Figure (9) *This figure is a 2D-histogram of the error vs depth plot. This plot shows where the majority of estimations lie within the scatter plot. For our model you can see a clump of estimates around 0 error at 200m and slight over-estimations above and below the 200m depth line. With CANYON-B you can also see a clump at 200m but with an error of about -0.025 and then further under-estimation at shallower depths. The 2D-histogram also highlights the lower variance in CANYON-B estimates as its estimates are more clumped together on the plot.*

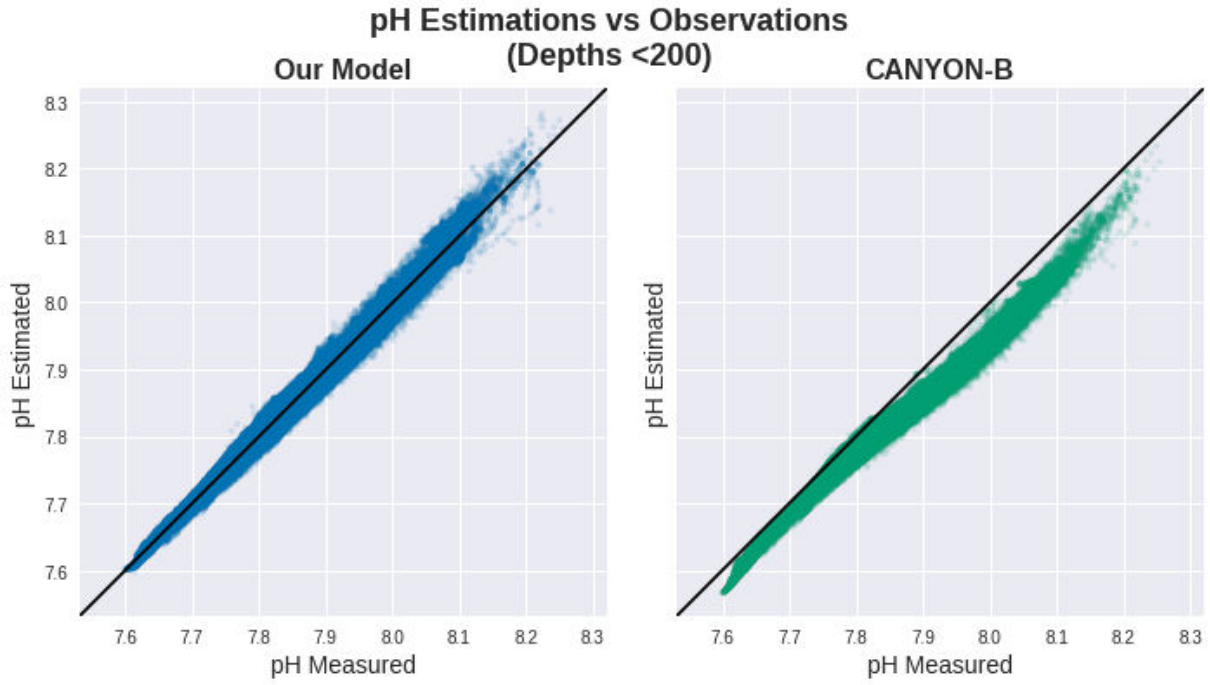

Figure (10) *This figure plots the pH estimations from our model and CANYON-B vs the measured pH for the same inputs but only for estimations for datapoints from less than 200m. A 1-1 line is plotted to show where "perfect" estimations lie. This figure shows that our model is better able to reduce bias in its estimations compared to CANYON-B. It can also be seen that both models have more accuracy and less variance in their estimations at lower pH values than higher pH values.*

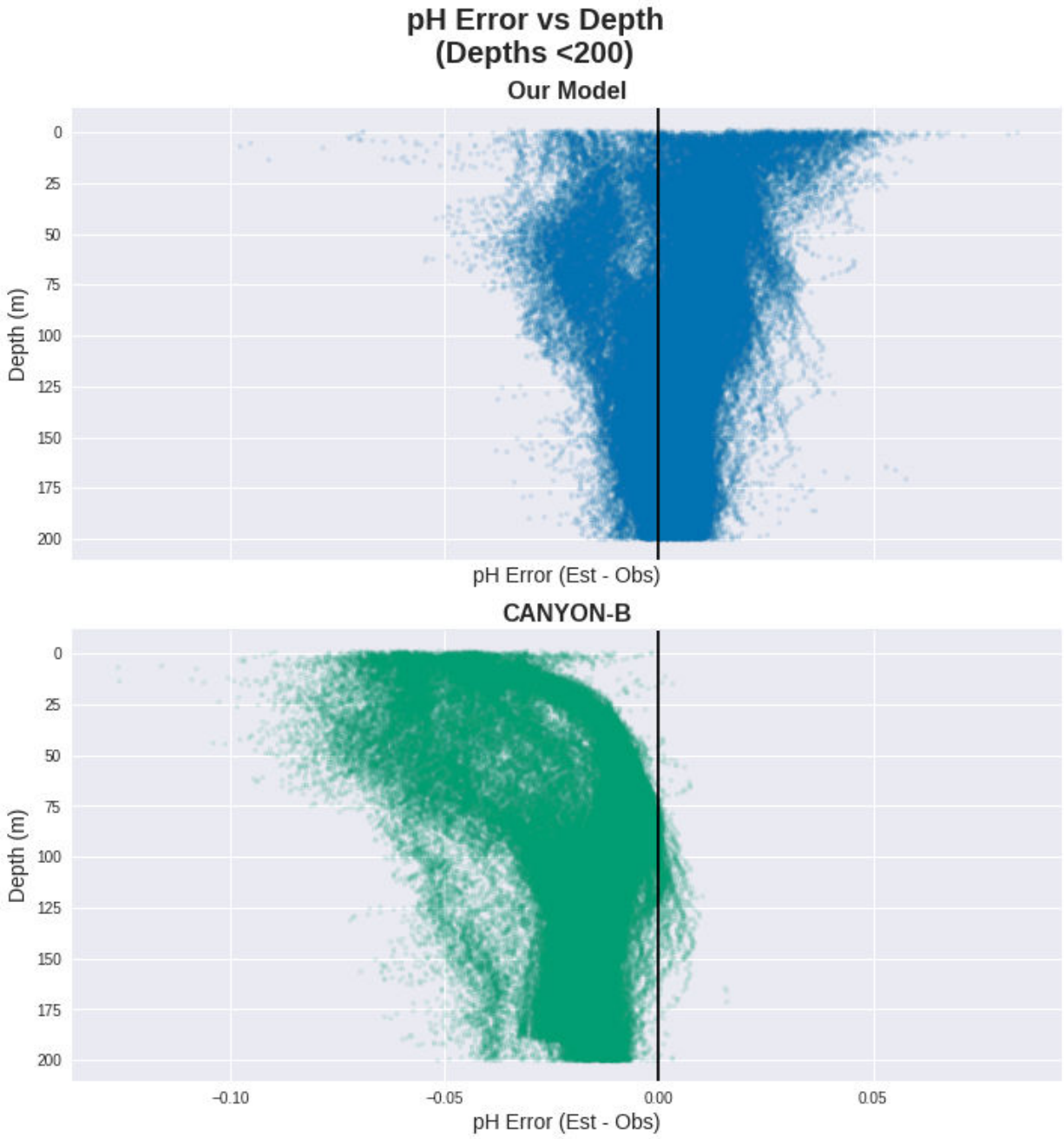

Figure (11) *This figure plots error vs the depth of the measurements for both our best single model and CANYON-B for all datapoints from less than 200m. In both plots, increased scatter and error can be seen at shallower depths. Our model stays centered on the zero-error line whereas CANYON-B hooks toward underestimations in the entire top 200m.*

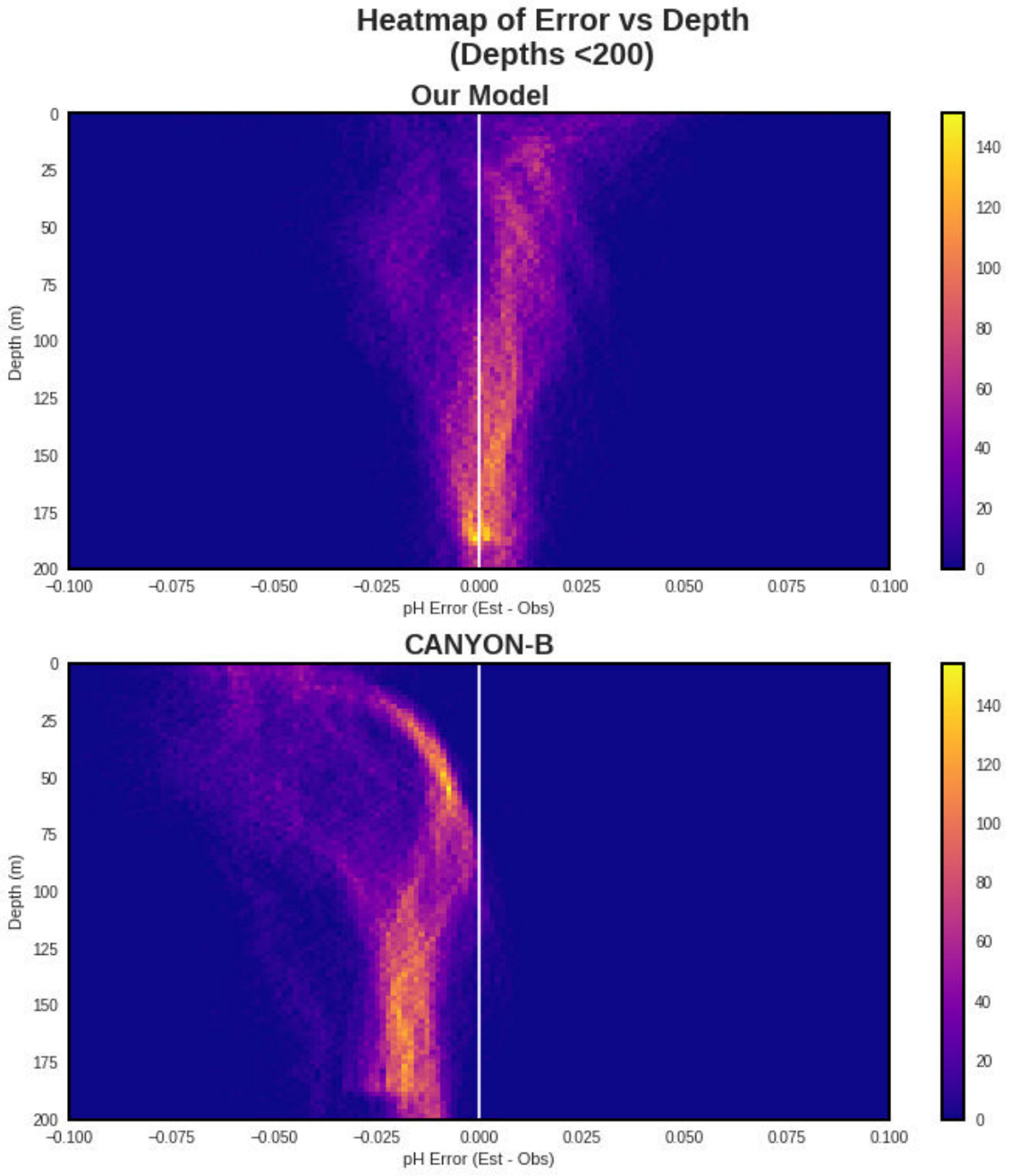

Figure (12) *This figure is a 2D-histogram of the error vs depth plot. This plot shows where the majority of estimations lie within the scatter plot. For our model you can see a clump of estimates around 0 error at 200m and subtle line extending upwards with a slight overestimation trend. CANYON-B shows almost every estimation is an underestimation and forms a sharp hook toward further underestimation at 0m.*

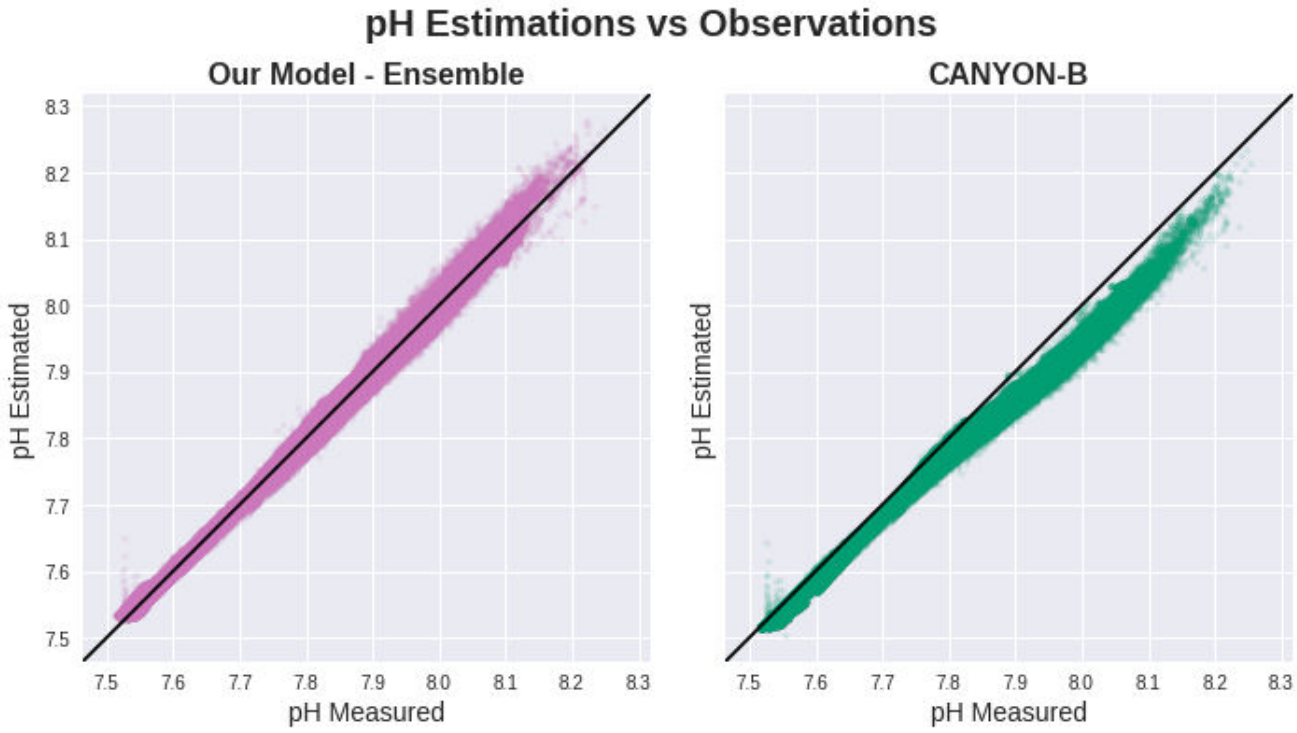

Figure (13) *This figure plots the pH estimations from our ensemble and CANYON-B vs the measured pH. A 1-1 line is plotted to show where "perfect" estimations lie. This figure shows that our ensemble is better able to reduce bias in its estimations compared to CANYON-B. It can also be seen that both models have more accuracy and less variance in their estimations at lower pH values than higher pH values.*

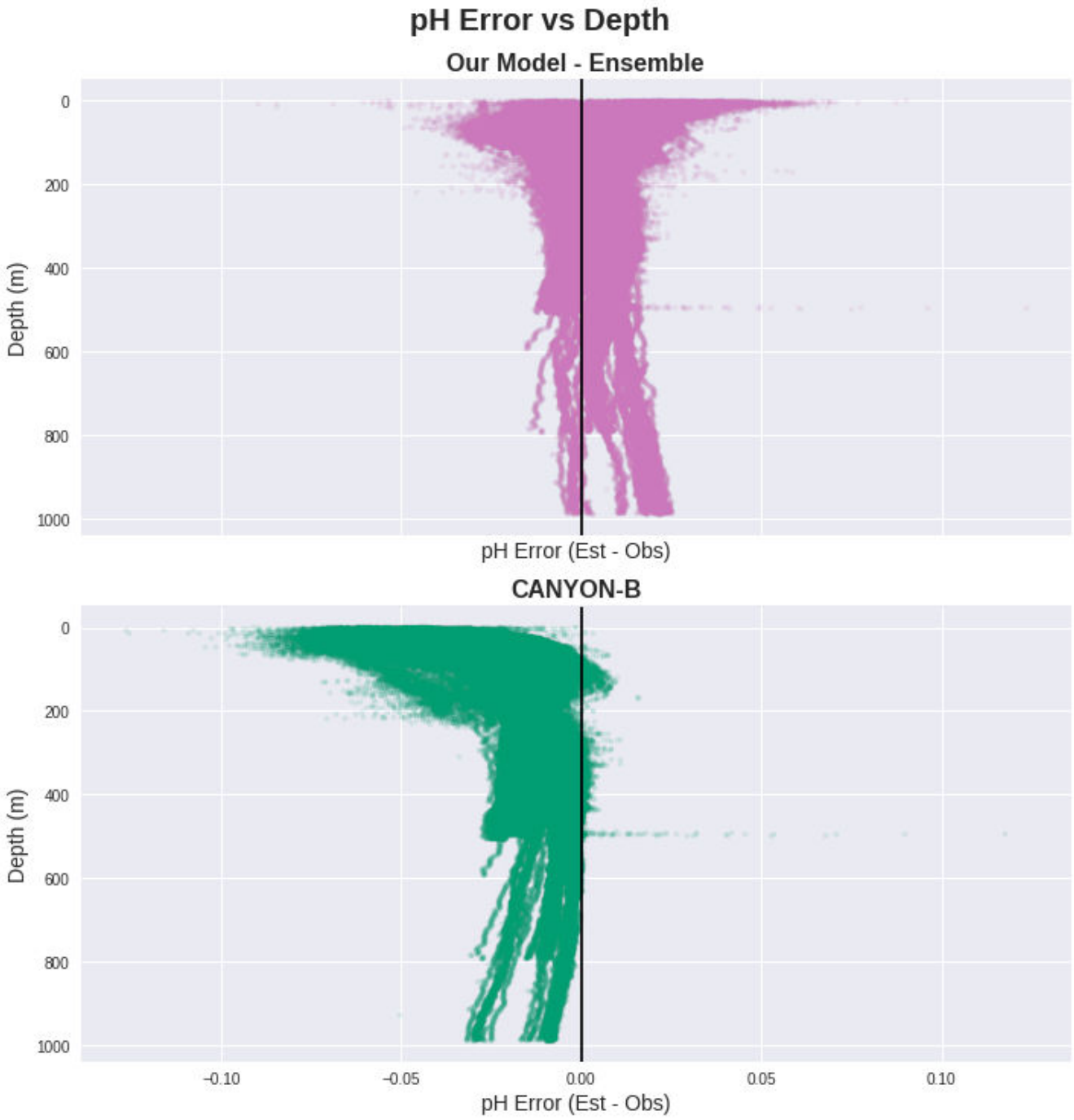

Figure (14) *This figure plots error vs the depth of the measurements for both our ensemble and CANYON-B. In both plots, increased scatter and error can be seen at shallower depths. Our model stays centered on the zero-error line whereas CANYON-B hooks toward underestimations in the entire top 200m. Our ensemble follows very similar trends to our single model in this plot.*

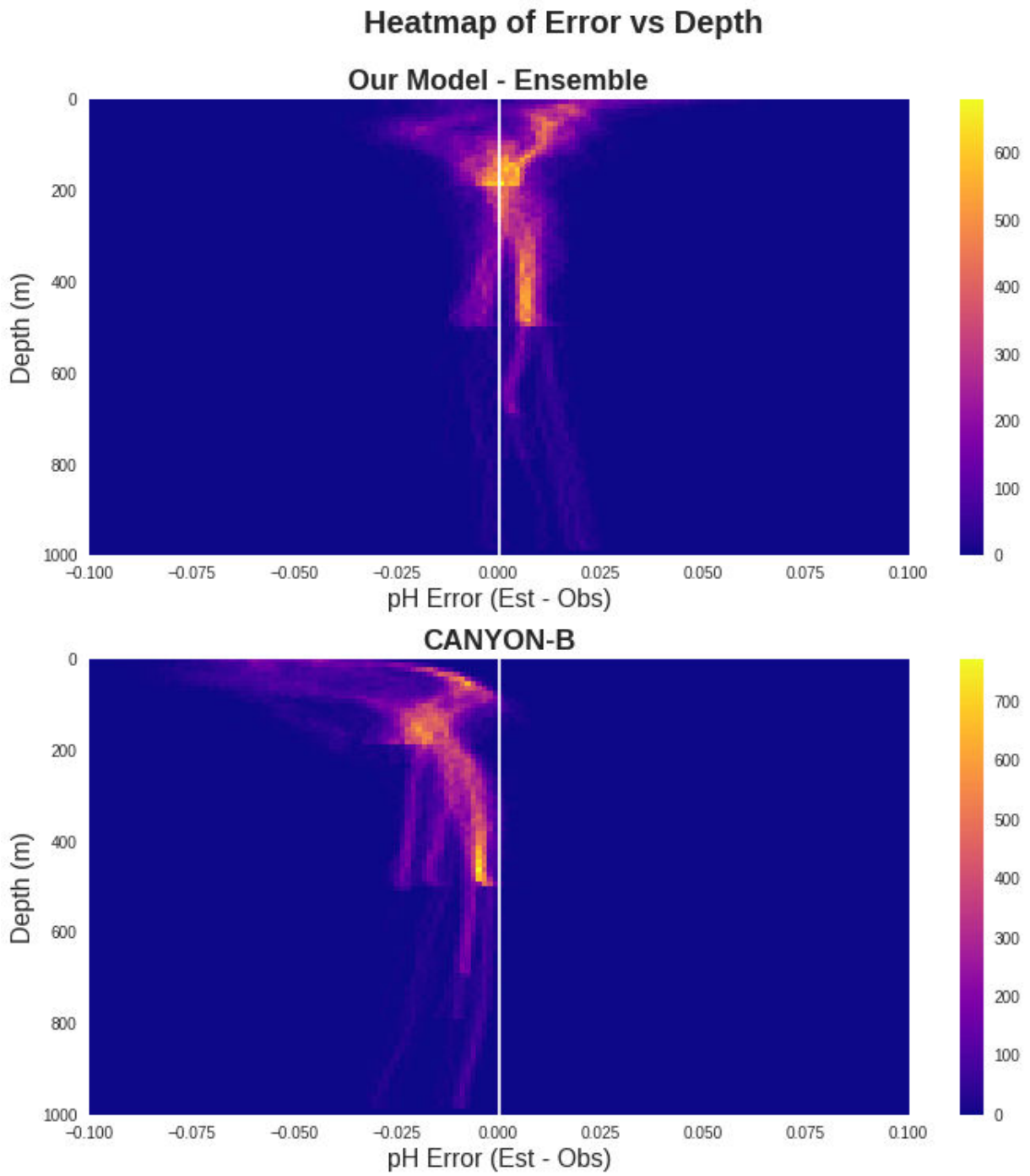

Figure (15) *This figure is a 2D-histogram of the error vs depth plot. This plot shows where the majority of estimations lie within the scatter plot. For our model you can see a clump of estimates around 0 error at 200m and sharp line extending upwards with a slight overestimation trend. This sharper line is from the decreased variance compared to the results of the single neural network. CANYON-B shows almost every estimation is an underestimation and forms a sharp hook toward further underestimation at 0m.*

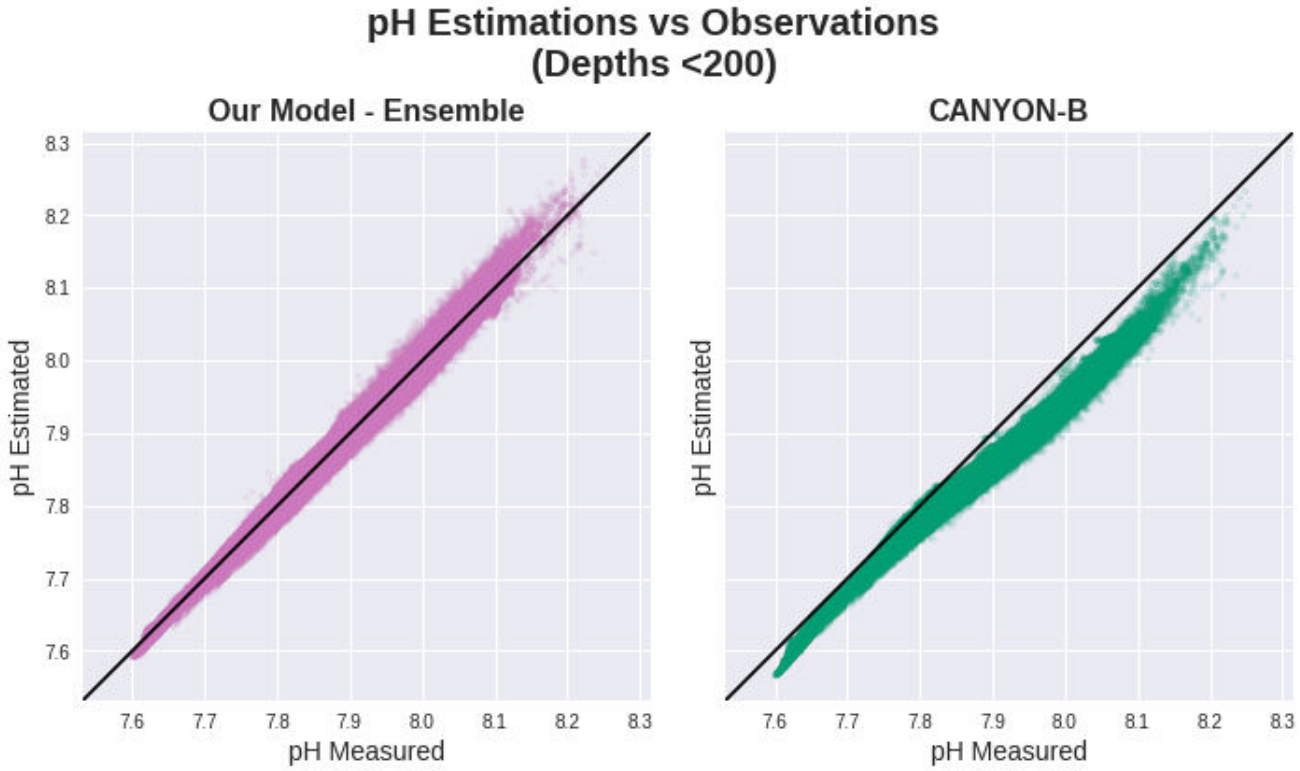

Figure (16) *This figure plots the pH estimations from our ensemble and CANYON-B vs the measured pH for the same inputs but only for estimations for datapoints from less than 200m. A 1-1 line is plotted to show where "perfect" estimations lie. This figure shows that our ensemble is better able to reduce bias in its estimations compared to CANYON-B. It can also be seen that both models have more accuracy and less variance in their estimations at lower pH values than higher pH values.*

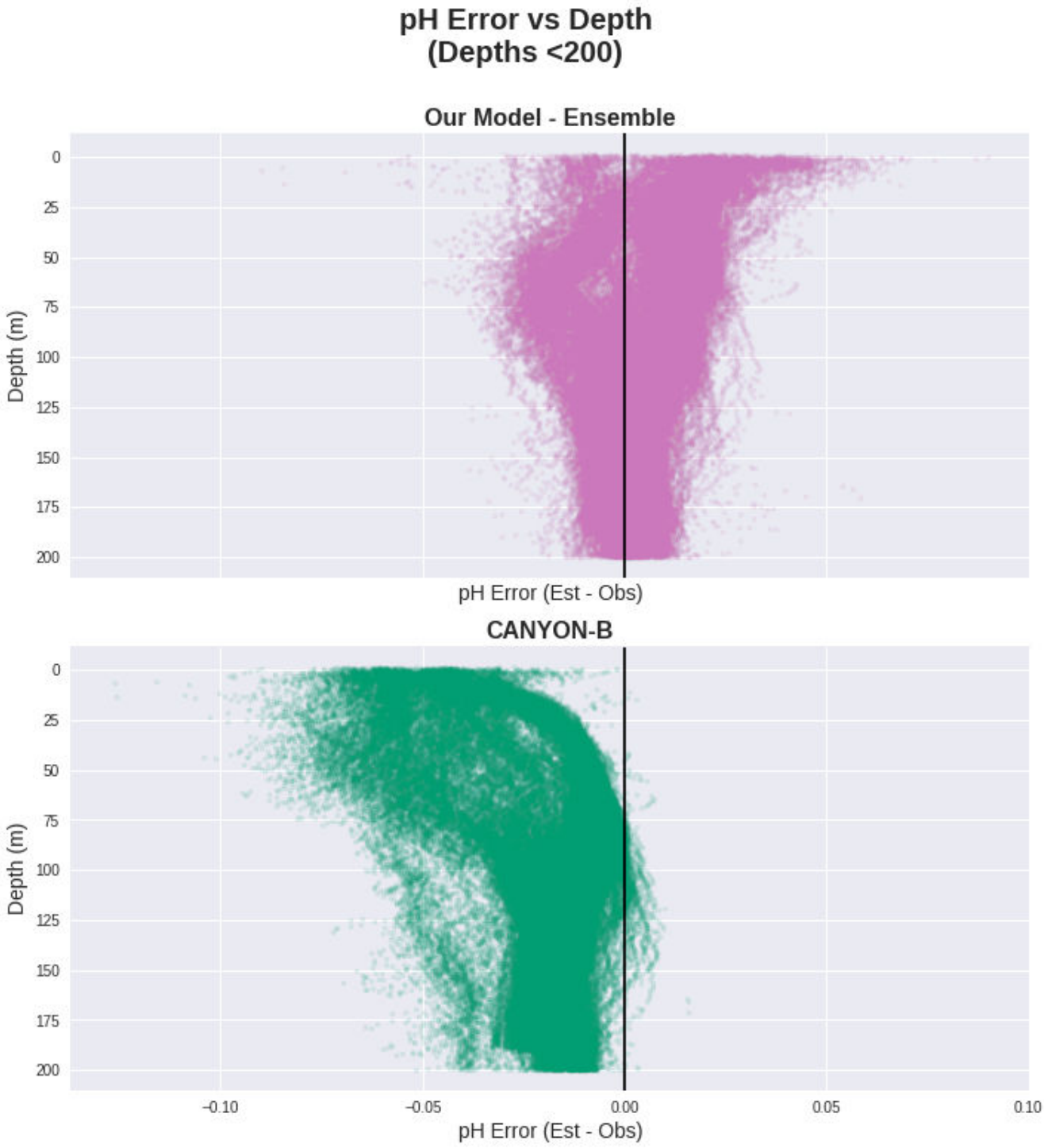

Figure (17) *This figure plots error vs the depth of the measurements for both our ensemble and CANYON-B for datapoints from less than 200m deep. In both plots, increased scatter and error can be seen at shallower depths. Our model stays centered on the zero-error line whereas CANYON-B hooks toward underestimations in the entire top 200m. Our ensemble follows very similar trends to our single model in this plot.*

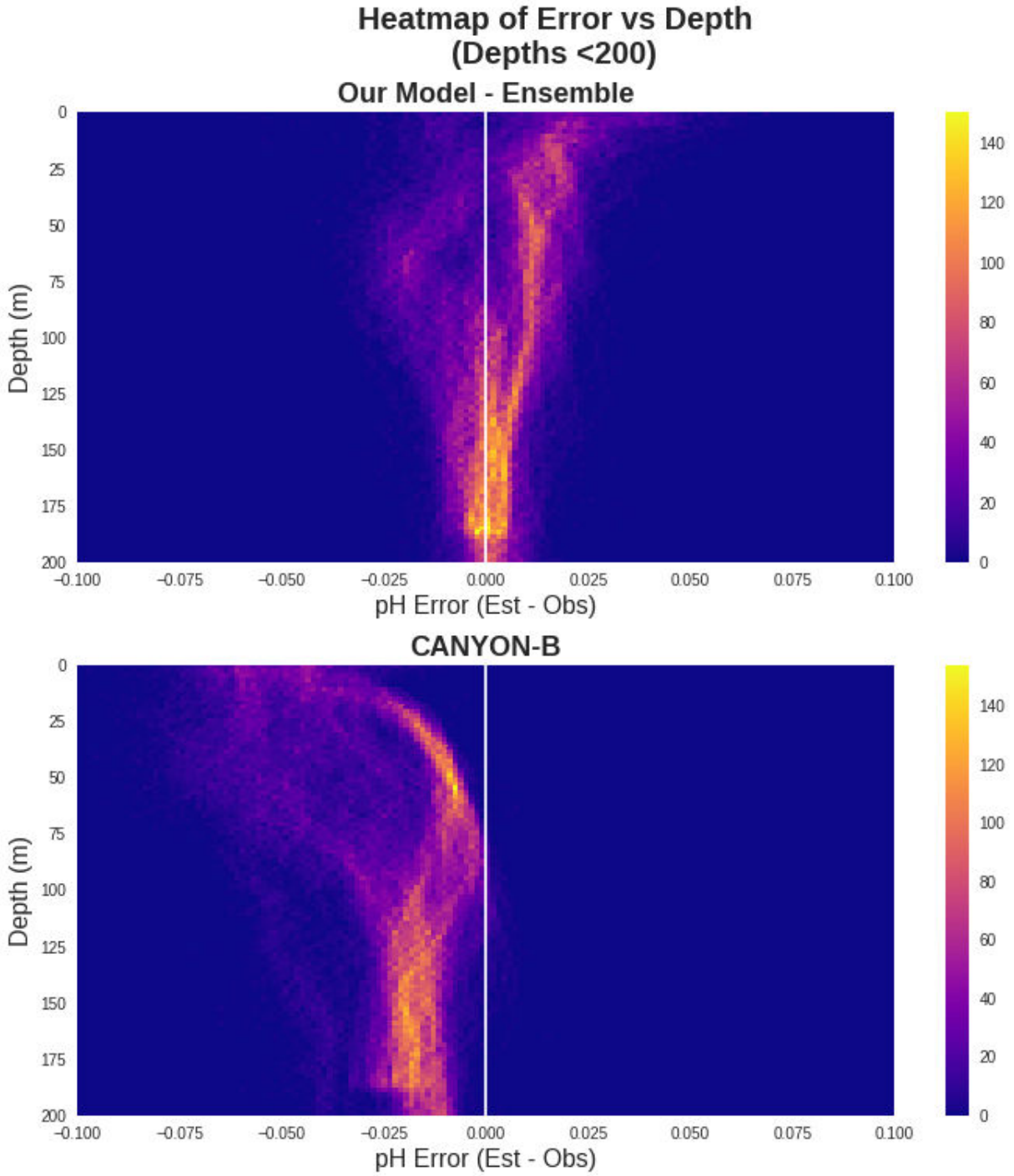

Figure (18) *This figure is a 2D-histogram of the error vs depth plot for datapoints measured from less than 200m. This plot shows where the majority of estimations lie within the scatter plot. For our model you can see a clump of estimates around 0 error at 200m and subtle line extending upwards with a slight overestimation trend. CANYON-B shows almost every estimation is an underestimation and forms a sharp hook toward further underestimation at 0m.*