PyESPERv1.0.0: A Python implementation of empirical seawater property estimation routines (ESPERs)

Larissa M. Dias[1,2], Brendan R. Carter[2]

[1]Cooperative Institute for Climate, Ocean, and Ecosystem Studies, University of Washington, Seattle, 98105, USA
[2]NOAA Pacific Marine Environmental Laboratory, Seattle, 98115, USA

Correspondence to: Larissa M. Dias (lmdias@uw.edu)

Deleted: 01

Deleted: 1

Abstract. This project presents a Python language implementation of locally interpolated regression (LIR) and neural network (NN) algorithms from empirical seawater property estimation routines (PyESPERv1.0.0). These routines estimate total alkalinity, dissolved inorganic carbon, total pH, nitrate, phosphate, silicate, and oxygen from geographic coordinates, depth, salinity, and 16 combinations of 0 to 4 additional predictors (temperature and biogeochemical information), and were previously available only in the MATLAB programming language. Here we document modifications to reduce discrepancies between the implementations, calculate the disagreements between the methods, and quantify Global Ocean Data Analysis Project (GLODAPv2.2022) reconstruction errors with PyESPER. While the PyESPER routine based on neural networks (PyESPER_NN) faithfully reproduces the corresponding MATLAB routine estimates of properties that do not require anthropogenic carbon change information, PyESPER_LIR and—to a lesser extent—PyESPER_NN estimates for total pH and dissolved inorganic carbon do not exactly reproduce the MATLAB routine estimates due to differences in interpolation and extrapolation methods between the programming languages. While the MATLAB and Python LIR-based estimates are not identical, we show that they are similarly skilled at reproducing the GLODAPv2.2022 data product and are thus comparable. This project increases the accessibility of ESPERv1.01.01 algorithms by providing users with code in the freely available Python language and enables future ESPER updates to be released in multiple coding languages.

Deleted: This project produced

Deleted:

Deleted: 1

Deleted: 1

1 Introduction

Ship-based biogeochemical data, as compiled within the Global Ocean Data Analysis Project (GLODAP; Lauvset et al., 2022) have high precision and accuracy, but are seasonally biased and spatially sparse (Hauck et al., 2023). International efforts to deploy biogeochemical (BGC) profiling floats with broad spatial coverage and high temporal resolution (10 days) are ongoing (Bittig et al., 2019), with potential to greatly augment available ocean carbon cycle and biogeochemical data. These data can then support a wide variety of research topics and management applications (e.g., warming, acidification, eutrophication, deoxygenation, fisheries, and ecosystem studies). This strategy leverages the high precision and accuracy of ship-based measurements to calibrate and validate the BGC float sensors periodically throughout a float deployment. To do this, machine learning and regression algorithms—which take advantage of the strong regional correlations between seawater properties in the open ocean, and especially the ocean interior (Bittig et al., 2018; Carter et al., 2017, 2021)—are used to map the ship-based information onto "reference depth" portions of the float profiles.

The empirical seawater property estimation routines (ESPERv1.01.01, henceforth referred to as ESPERs), originally written in MATLAB programming language, aim to help realize the full potential of BGC float data by using machine learning techniques and regression strategies to predict total alkalinity (TA), dissolved inorganic carbon (DIC), pH on the total scale (pH_T), phosphate, nitrate, silicate, and oxygen from commonly measured physical and BGC parameters (Carter et al., 2021).

The algorithms are used to calibrate float profiles (Maurer et al., 2021). In addition, since two carbonate system property measurements are necessary to fully quantify the carbonate system in seawater (Zeebe and Wolf-Gladrow, 2001) and BGC floats only have the capability to measure pH_T, these algorithms have the potential to provide (calculated) TA or DIC as a secondary constraint for the marine carbonate system. This method also offers an alternative to using models to estimate variables for carbonate chemistry calculations when nutrient information is unavailable, which potentially has high error values. ESPERs have also been used to map ship-based information across spatial and temporal scales for other applications including estimation of TA for adjustments of pH and fugacity of CO_2 (fCO_2) to in situ conditions for data products (Jiang et al., 2021), and estimation of TA and seawater properties necessary for estimation of ocean acidification indicators (Jiang et al., 2020; Sharp et al., 2024). Recent research has also shown that similar machine learning estimation algorithms have potential for the development of four-dimensional data products such as the Gridded Ocean Biogeochemistry from Artificial Intelligence – Oxygen (GOBAI-O_2; Sharp et al., 2023) and the Mapped Observation-Based Oceanic DIC (MOBO-DIC; Keppler et al., 2020).

1.1 Importance

Tanhua et al. (2021) and others have argued that researchers should utilize workflows that produce findable, accessible, interoperable, and reusable (FAIR) data products. ESPERs are publicly available (findable) on Zenodo, with updates published to GitHub, free (accessible), and provide the option for users to cite a digital object identifier (DOI) for each version (reusable). However, until now ESPERs were only available in the proprietary MATLAB programming language, which posed a barrier to accessibility and interoperability that we aim to address. Future updates may include even more accessible features such as a user interface.

1.2 Goals

This project aimed to create a freely available Python implementation of ESPERs (PyESPERv1.0.0, henceforth referred to as PyESPERs; Carter et al., 2021; Dias and Carter, 2025) that is equivalent to the MATLAB version within ±2 × Estimate Uncertainties (σ) for all estimated biogeochemical properties (TA, DIC, pH_T, nitrate, phosphate, silicate, and oxygen). PyESPER code is freely available at Zenodo and updates will be made available at the GitHub repository (*see* Sect. "*Code availability*").

Deleted: 1.

Deleted: 1

Deleted: 1

2 Methods

ESPER algorithms were translated into Python coding language, while associated files were either translated into Python or read by Python as MATLAB files. Some original methods were required to allow interpolations to be similar in Python to those of MATLAB ESPERs.

2.1 ESPERs

ESPERs allow estimation of biogeochemical seawater properties using coordinates, depth, salinity, and other optional inputs from a single function call. While sharing a similar set of equations and required input data, ESPERs have two variants that use locally interpolated regressions (ESPER_LIR) and neural networks (ESPER_NN), respectively, along with a mixed estimate (ESPER_Mixed) that is the mean of estimates from the two functions (Carter et al., 2017). There are a couple of reasons to maintain the separate ESPER LIR, NN, or Mixed options, from an end-user perspective, and these reasons are also true for PyESPERs.

1. ESPER_LIRs predate the ESPER_NNs and have been used as a standalone data product for various research purposes (*see* Carter et al., 2016; Carter et al., 2018). Long-term users of these LIRs have previously expressed desire for consistency between versions (e.g., when depth was taken out as predictor for pH_T), and some of them already use CANYON-B (Bittig et al., 2019) as a neural net option for comparison. Therefore, these users who desire consistency would most likely prefer to use ESPER_LIR.
2. ESPER_LIRs are more transparent than ESPER_NN, as it is simple to parse apart coefficients at the gridded locations and to see how the equations are a result of these. ESPER_LIRs also rely on a grid, which may appeal to some users.
3. ESPER_NNs provide improved estimates on average than ESPER_LIRs and behave more like a mapping product in that 3D coordinates are predictors, which may alternately appeal to some users.
4. Although the ESPER_Mixed estimates perform better on average than LIRs or NNs do independently, there are cases where they have greater bias and RMSE than LIRs or NNs (e.g., when using equations 1-3 for phosphate or nitrate at all depths; Carter et al., 2021). Users may want to assess each scenario independently and choose which method is most appropriate according to their needs.
5. The NNs are more closely reproduced between the MATLAB and Python ESPER implementations.

2.1.1 Locally interpolated regressions

The most recent versions of ESPER_LIRs (version 1.01.01; version 3 of LIRs) use a standard set of equations of the format shown by Eq. (1) to estimate up to seven different biogeochemical water properties using up to 16 equations with different combinations of input parameters (*see* Sect. "*Appendix A*", Tables A1 and A2; Carter et al., 2021):

$$X = C_0 + \sum_{i}^{n} C_i P_i \quad (1)$$

where X is the estimated property (TA, DIC, pH_T, nitrate, phosphate, silicate, or oxygen), C_0 is the intercept, and C_i is the coefficient for each of the n predictors P_i. The intercepts (C_0) and coefficients (C_i) vary with location (latitude, longitude, and depth) and are different for each of the predictor variables (P_i; Tables A1 and A2; Carter et al., 2021). The most recent ESPERs were trained and assessed on the GLODAPv2.2020 (Olsen et al., 2020) data product, which includes data from 946 cruises

Formatted: Line spacing: 1.5 lines

Formatted: Font: 10 pt, Not Bold, Font color: Auto

Deleted: ¶

and spanning 1972–2019, and additional data sets from the Mediterranean Sea and Gulf of Mexico (Carter et al., 2021, Supplementary Information) taken from the Coastal Ocean Data Analysis Project (CODAP, Jiang et al., 2021) and the CARIMED data product (Álvarez et al., 2019).

When the ESPER_LIR function is called, the routines interpolate a pre-determined grid of *C*'s (intercepts and coefficients) to user-defined locations. Linear interpolation is used within the grid and for extrapolation, and this method utilizes an underlying Delaunay triangulation with MATLAB's scatteredInterpolant function (Carter et al., 2021). The three-dimensional interpolation and extrapolation algorithm is implemented differently in MATLAB and Python, and although both calculations are valid, this difference in implementation is the source of disagreements we find and later quantify between ESPER and PyESPER.

ESPER_LIR coefficients have been determined on a grid using a moving window regression strategy similar to the approach first outlined by Velo et al. (2013), resulting in a set of intercept and coefficient estimates for each of 16 equations for 7 possible properties at 44,957 total locations on a 5° latitude (-84.5°–85.5° N) x 5° longitude (-19.5°–375.5° E) x 33 depth (0–5500 m) ocean interior grid subsampled from the World Ocean Atlas gridded product (Carter et al., 2016, 2017, 2021). These coefficients were fit using regressions relating the property of interest (*X*) to different combinations of up to five predictor properties (*P*, Tables A1 and A2), relating to each possible equation as in Eq. (1). Depth (scaled to $\frac{1}{25}$) is included as a coordinate for coefficient interpolation, but depth is not used as a predictor for the current ESPER version (it was included in an earlier version, but only when predicting pH_T; Carter et al., 2017). Data for each regression fit are selected from "windows" of data that are within 15° latitude, 30°/cosine(latitude) in longitude, and within either (100 + *z*/10) m depth or 0.1 kg m^{-3} of the estimated density of seawater at that coordinate location, where *z* is depth in m (Carter et al., 2021). If either the depth-based or the density-based criterion applies, data are selected for that location, which allows water masses to impact window selection along with depth. If fewer than 100 measurements fall within a window, the dimensions are doubled. In LIRv2, windows were iteratively scaled by a factor of the iteration number until at least 100 measurements are selected to train each regression (Carter et al., 2017). For ESPER_LIRs (LIRv3), it is argued that increasing window size has the following benefits: (1) includes more data for regression fits, (2) introduces more modes of oceanographic variability into fitting data, and (3) reduces multicollinearity (Carter et al., 2021). However, the risk of increasing window size is that they will be less appropriate locally. A weighting term is applied to help account for this by reducing to cost of regression misfits to data that are distant or at significantly different depths from the location, with a cap to prevent overfitting to nearby coordinates (see Carter et al., 2021). Regression coefficients (C_0 and C_i) are then fit using Eq. (2), with separate regressions for northern hemisphere Atlantic, Mediterranean, and Arctic, and other global locations, to prevent interpolation across Central America or the Bering Strait.

Deleted: most

Formatted: Line spacing: 1.5 lines

Formatted: Font: 10 pt, Not Bold, Font color: Auto

Deleted: reduce

$$XW = \left(C_0 + \sum_{i}^{n} C_i P_i\right) W \quad (2)$$

PyESPER_LIR does not duplicate this portion of the effort but instead builds directly upon the grid of coefficients obtained for and utilized by the MATLAB implementation of ESPER_LIR.

When the function is called, ESPER_LIR uses MATLAB's scatteredInterpolant (linear interpolation and extrapolations) function to interpolate this previously-created grid of regression coefficients to the user-provided set of coordinates, resulting in coefficient estimates at the desired locations (Carter et al., 2021). This method uses a Delaunay triangulation of the scattered sample points to perform interpolations and extrapolations. Different valid mathematics can be used to obtain these Delaunay triangulations and to extrapolate and interpolate, and efforts to identify a Python method for these tasks that exactly replicated MATLAB results were unsuccessful. The most similar and least computationally intensive results to those of MATLAB's scatteredInterpolant were produced by combining Python's scipy package functions LinearNDInterpolator (interpolate subpackage) and Delaunay (spatial subpackage; Virtanen et al., 2020). However, since LinearNDInterpolator does not extrapolate, and other Python functions did not produce similar results to those of MATLAB when using similar methods, the gridded set of three-dimensional coordinates (44,957 locations based on the World Ocean Atlas) and corresponding coefficient estimates provided by ESPER_LIRs were expanded in MATLAB to 106,400 locations on a grid with estimates every 5° latitude (-94.5°–90.5° N) and longitude (-19.5°–375.5° E) and up to 9000 m depth and applied to scatteredInterpolant within ESPER_LIR to provide coefficient estimates for the external locations through extrapolation. This grid, with equivalent coefficients within the original parts of the grid and extrapolations outside of the grid, was read in Python when LIRs were called. The expanded grid allowed Python functions to avoid extrapolations and rely solely on interpolation and triangulation methods when estimating coefficients at user-defined locations. While some of these locations are unphysical (e.g., ± >90º N or on land), the coefficients nevertheless provide valid extrapolations from MATLAB for the full possible domain that can then be interpolated in PyESPER_LIR. PyESPER_LIR otherwise replicated ESPER_LIR's separation of data from the Atlantic Ocean, Mediterranean Sea, and Arctic Ocean and data from the Indo-Pacific and Southern Ocean regions.

Deleted: extrapolating

During the creation of this expanded grid, a grouping error was observed in current versions of MATLAB ESPER_LIRs. Specifically, the mirrored portion of the grid found at < 0º E and > 360º E and north of 40º S are not correctly flagged as belonging to the Atlantic grid. The practical effect of this bug was that estimates near the Prime Meridian and near the cutoff between the Southern Ocean and the Atlantic Ocean had extrapolated coefficients instead of interpolated coefficients. This bug was fixed for both MATLAB ESPER_LIR and PyESPER_LIR comparisons for this paper, and a fixed grouping routine is now provided at the original MATLAB ESPER repository with corresponding documentation and will be included in future updates to ESPER_LIRs.

2.1.2 Neural networks

ESPER_NNs use feed-forward neural networks with latitude, depth, cosine(longitude-20°E), cosine(longitude-110°E) and the parameters from Table A2 as predictors. Four neural networks are used in each of the two ocean regions, which are the same as those used for LIRs (Atlantic-Mediterranean-Arctic and Indo-Pacific-Southern), resulting in 896 total neural networks (8 for each of 16 combinations of predictors for 7 property estimates; Carter et al., 2021). An ensemble of four previously-created neural networks with different combinations of neurons and hidden layers, including a single one-hidden-layer network with 40 neurons and three two-hidden-layer networks with 30/10, 25/15, and 20/20 neurons in the first/second hidden layers is used to minimize the impact of errors from any one neural network (Carter et al., 2021).

In ESPER_NN the neural networks are encoded as functions to avoid requiring access to the Machine Learning toolbox within MATLAB. Here we further translate these functions to Python. The resultant Python functions replicate the functions in ESPER_NN to within machine precision. ESPER_NNs linearly interpolate between the two regions of neural networks by latitude across the Southern Atlantic Ocean and Bering Sea and between the North Pacific and Arctic Oceans. Zonal transitions in the Southern Atlantic and Indo-Pacific-Southern Ocean network are also implemented. This interpolation uses custom-written 1 or 2D interpolations that are handled identically in both programming environments.

2.1.3 Mixed estimates

The mixed estimate for each input location is the mean of the LIR and NN estimates and therefore is trivially reproduced by a simple single function call and module within Python.

2.1.4 Anthropogenic carbon

The impacts of anthropogenic carbon (C_{ant}) are approximated in ESPER and PyESPER using a 1° x 1° gridded transit time distribution (Waugh et al., 2006) based C_{ant} product referenced to the year 2002 (Lauvset et al., 2016). ESPERs assume that oceanic C_{ant} increases proportionally to atmospheric anthropogenic CO_2 (transient steady state assumptions; Gammon et al., 1982; Gruber et al., 2019; Tanhua et al., 2007). This implies that the "shape" of the C_{ant} vertical profile (gradient) remains constant with continuous exponential increases of atmospheric CO_2 and ocean C_{ant} according to Eq. (3; Carter et al., 2021).

$$C_{ant\_year\_location} = C_{ant\_year\_location} e^{0.018989(year-2002)} \quad (3)$$

The coefficient in Eq. (3) is derived from Gruber et al.'s (2019) assumption of a 28% increase in C_{ant} from 1994–2007, and enables estimating C_{ant} for a location in a desired year when C_{ant} is known for that same location in a reference year (2002;

Deleted: committee (
Deleted:)
Deleted: v1.0
Deleted: , which rely on
Moved down [1]: (Gammon et al., 1982; Gruber et al., 2019; Tanhua et al., 2007)
Deleted:
Deleted: , and
Deleted: estimate C_{ant} using a transient steady state assumption (Gammon et al., 1982; Gruber et al., 2019; Tanhua et al., 2007). The ESPER estimate assumes
Moved (insertion) [1]
Deleted: ,
Deleted: and furthermore that
Deleted: structure

Carter et al., 2021). This approach does not allow for non-steady-state variations, which is accounted for in overall uncertainty estimates, and is noted as a significant source of uncertainty for projections beyond ~2030.

ESPERs were trained on data for pH_T and DIC which were transformed to the year 2002, then modified back to the original measurement dates using Eq. (3). ESPERs and PyESPERs estimate the C_{ant} component of DIC and pH_T in output variables for 2002 by interpolating the 2002 C_{ant} grid to user-provided coordinates and then applying Eq. (3) to estimate C_{ant} for the user-requested estimate year. As with original ESPERs, this method is not meant to be used when C_{ant} is of primary interest, but rather provides a means of quickly adjusting DIC or pH_T to a reference year (Carter et al., 2021). Likewise, these methods are not adequate for making reliable projections beyond the year 2030, or perhaps sooner, in coastal or other areas where the underlying global open-ocean anthropogenic carbon estimations have greater uncertainties (Carter et al., 2021).

Deleted: Likewise, these methods are not adequate for making reliable predictions beyond the year 2030

Deleted: in

Deleted: with abnormal atmospheric CO_2 absorption profiles

Deleted: i.e., upwelling, coastal areas, high freshwater outflow;

Deleted:

Deleted:

Deleted: (

Deleted: ¶

2.2 Uncertainty estimation

ESPERs and PyESPERs return depth- and salinity-dependent uncertainties for each property at the 1**σ** (one standard uncertainty) level, meaning approximately 95% of new open-ocean measurements from GLODAPv2.2022 should fall within ± twice the ESPER uncertainties (Carter et al., 2021). As in Carter et al. (2021), baseline error estimates in depth and salinity space (E_{X_Est}) are interpolated based on root mean square errors (RMSEs) of all predictions from validation versions of the routines within bins of salinity and depth. ESPER_LIRs and PyESPER_LIRs scale these uncertainties using user-provided predictor uncertainty estimates ($E_{Pi\_Provided}$). Eq. (4) is used when user-provided uncertainties exceed default assumed input uncertainties ($E_{Pi\_Default}$; Table A3):

$$E_{X\_Output} = \sqrt{E_{X\_Est}^2 - \sum_{i=1}^{n}\left(\frac{\partial X}{\partial P_i}E_{Pi\_\mathrm{Default}}\right)^2 + \sum_{i=1}^{n}\left(\frac{\partial X}{\partial P_i}E_{Pi\_\mathrm{Provided}}\right)^2} \quad (4)$$

where $\frac{\partial X}{\partial P_i}$ is the sensitivity of the property estimate X to the i^{th} predictor P_i. ESPER_NNs and PyESPER_NNs estimate sensitivities by iteratively perturbing the input predictors if the user specifies uncertainties that are larger than default. Mixed uncertainties are the minimum uncertainties assessed for LIR and NN estimates.

2.3 Assessment

For many applications, the most critical validation is a test of the reconstruction of withheld data. However, such an exercise requires training alternative versions of the method after withholding data, and, as of now, PyESPER is not separately trained, but is instead reliant on the ESPER training that was performed and validated previously with MATLAB (Carter et al. 2021). For this publication, we aim to instead show that PyESPER and ESPER provide quantitatively similar results and assert that the validation presented earlier for ESPER in MATLAB can be considered to also be appropriate for PyESPER in all but a

Deleted: v1.0

Deleted: v1.0

limited number of specific exceptional cases. To support this claim, PyESPER and ESPER were used to estimate values for the GLODAPv2.2022 data product (1,381,248 sets of measurements; Fig. 1) with each equation and output variable combination. This dataset included a wide range of input data, and comparison of PyESPER and ESPER was primarily considered from application to the high-quality "open ocean" ($_o$) portion of the GLODAP dataset as in Carter et al. (2021), defined as GLODAP data with only World Ocean Circulation Experiment (WOCE) data quality control flag categories of 2 (Acceptable) and secondary quality control flag categories of 1 (subjected to full secondary quality control) for all possible input and measurement data, and for salinities between 30–37 (n=306,227 for TA, 343,580 for DIC, 199,304 for pH_T, and 764,301 for phosphate, nitrate, silicate, and oxygen). Additional comparison with the entire GLODAPv2.2022 dataset ("whole ocean" or $_w$), including NaNs and anomalous data with salinities <30 and temperatures <0 °C, which are not recommended for use with ESPERs, is presented in Sect. "*Appendix B*". These comparisons are used as a rigorous test of the fidelity of the PyESPER estimates to the ESPER estimates. Resulting estimates were compared graphically and with normalized root mean square error ($RMSE_n$; equivalent to RMSE divided by the mean of the MATLAB estimate for each variable) for each equation case globally and regionally, and across depths. $RMSE_n$ was used because it allows for comparison between variables of different scales. Additionally, where measured values were present in the dataset, both ESPER and PyESPER were validated against the measured data, though, again, this is not a validation of the method as much as a check that both variants provide similar values.

Deleted: v1.0

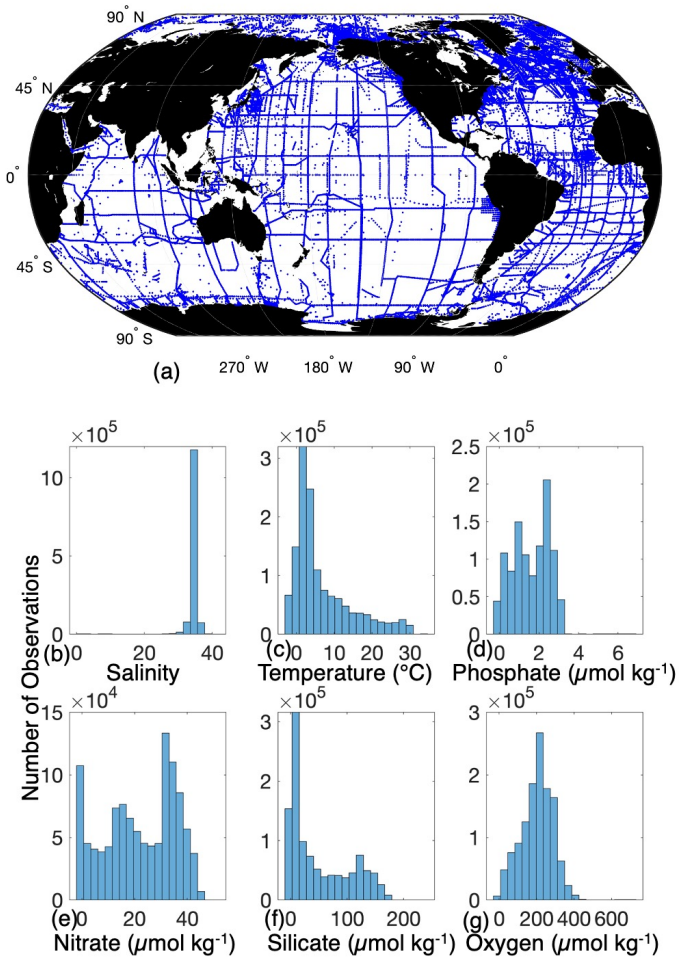
Deleted: v1.0

Deleted: v1.0

Deleted: v1.0

2.3.1 DIC application

As an additional comparison of the LIR method differences, DIC estimates from both PyESPER_LIR and ESPER_LIR were applied to the Roemmich and Gilson Argo-derived climatology (Roemmich and Gilson, 2009) to create mapped annual surface estimates of DIC.



280

Figure 1: Location of GLODAPv2.2022 data used to compare PyESPER to MATLAB ESPER estimates (a), and histograms of the distributions of measured GLODAPv2.2022 variables used as inputs for PyESPERv1.0.0 and ESPER algorithms (b-g).

Deleted: (*n*=1,381,248)

Deleted: 1

Deleted: 1

3 Results and Discussion

PyESPER and ESPER produced open ocean estimates with mean differences (Python estimate − MATLAB estimate) of <±0.04 for all parameters, and NNs had smaller mean differences of <±0.004 for all parameters (units are µmol kg^{-1} except for pH_T) estimated from open ocean GLODAPv2.2022 data, although the standard deviations of these differences and uncertainties associated with estimates were at times larger than the mean differences (Tables 1 and 2). The greatest $RMSE_n$ was $2.08x10^{-2}$ for silicate estimates using LIRs. PyESPER_NN functioned as an equivalent data product to ESPER_NN for all data. For open ocean data, PyESPER_LIRs functioned similarly to ESPER_LIRs, with a large majority of identical estimates produced between the two data products.

3.1 Data product validation

Results of comparisons between MATLAB ESPERs and PyESPERs are described below.

3.1.1 Locally interpolated regressions

When compared to the ESPER_LIR results for the open ocean ($_o$) GLODAPv2.2022 dataset, all equation-case and desired outcome variable combinations from PyESPER (PyESPER_LIR − ESPER_LIR estimates) resulted in mean differences of <±0.04 (Table 1). Mean (±standard deviation; $RMSE_n$) PyESPER − ESPER_LIR differences for each property are shown in Table 1. The very wide range of input data resulted in a wide range of estimates from both ESPER_LIRs and PyESPER_LIRs for all variables (Table 1; Fig. 2; for $_w$ *see* Sect. "*Appendix B*", Fig. B1), representing the large range of biogeochemical property values that can be found in the oceans. PyESPER_LIR and ESPER_LIR results worked similarly well in predicting measured values at locations, even with the outlier and unusual input data used (see Table B1), suggesting that Python estimates, although not identical to MATLAB estimates for these interpolations, were equivalently valid reconstructions.

Deleted: v1.0

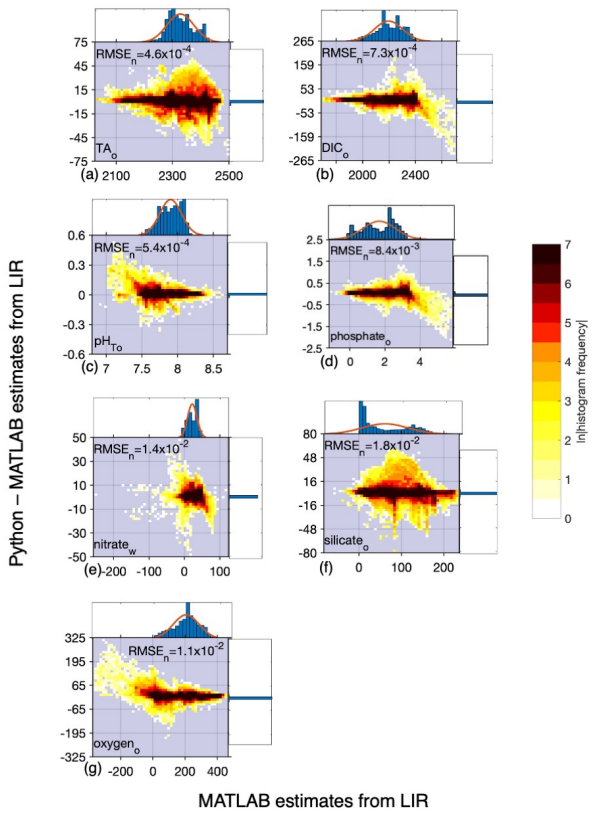
Deleted: that is within the World Ocean Atlas Grid of <5500 m depth…

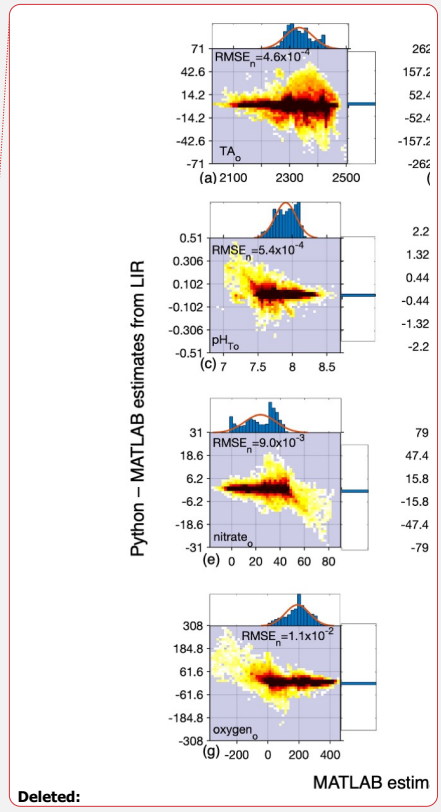
Deleted: for TA was $-4.8x10^{-4}$ µmol kg^{-1} (±1.1 µmol kg^{-1}; $RMSE_n=4.6x10^{-4}$), DIC was $3.4x10^{-2}$ µmol kg^{-1} (±1.6 µmol kg^{-1}; $RMSE_n=7.3x10^{-4}$), pH_T was $-5.6x10^{-5}$ (±$4.2x10^{-3}$; $RMSE_n=5.4x10^{-4}$), phosphate was $3.1x10^{-4}$ µmol kg^{-1} (±$1.4x10^{-2}$ µmol kg^{-1}; $RMSE_n=8.4x10^{-3}$), nitrate was $2.2x10^{-3}$ µmol kg^{-1} (±$3.1x10^{-1}$ µmol kg^{-1}; $RMSE_n=1.3x10^{-2}$), silicate was $2.3x10^{-2}$ (±1.2 µmol kg^{-1}; $RMSE_n=2.1x10^{-2}$), and oxygen was $4.0x10^{-3}$ µmol kg^{-1} (±2.1 µmol kg^{-1}; $RMSE_n=1.1x10^{-2}$; Table 1).

ble 1: Mean (standard deviation), maximum, minimum, and normalized RMSE (RMSE$_n$), for differences between MATLAB and Python Rs, ESPER_LIR and measured values, and PyESPER_LIR and measured values for TA, DIC, pH$_T$, phosphate, nitrate, silicate, and ygen estimates (all units except pH$_T$ are µmol kg^{-1}) for open ocean ($_o$) data and all equations combined (n=13,384,096 for TA, 13,384,096 · DIC, 13,384,096 for pH$_T$, 13,384,096 for phosphate, 12,718,592 for nitrate, 12,640,896 for silicate, and 12,757,792 for oxygen).

	Python – MATLAB				MATLAB - Measured				Python - Measured			
	Mean	**Max**	**Min**	**RMSE$_n$**	**Mean**	**Max**	**Min**	**RMSE$_n$**	**Mean**	**Max**	**Min**	**RMSE$_n$**
TA	-4.75×10^{-4} (1.08)	6.44×10^{1}	-7.03×10^{1}	4.64×10^{-4}	2.71×10^{-1} (6.34)	8.13×10^{2}	-1.69×10^{2}	2.72×10^{-3}	2.70×10^{-1} (6.32)	8.13×10^{2}	-1.73×10^{2}	2.71×10^{-3}
DIC	3.39×10^{-2} (1.60)	2.01×10^{2}	-2.61×10^{2}	7.29×10^{-4}	-4.40×10^{-1} (8.55)	6.20×10^{2}	-3.20×10^{2}	3.90×10^{-3}	-4.02×10^{-1} (8.47)	6.20×10^{2}	-3.16×10^{2}	3.90×10^{-3}
pH	-5.65×10^{-5} (4.24×10^{-3})	5.05×10^{-1}	-3.77×10^{-1}	5.36×10^{-4}	-2.51×10^{-3} (2.24×10^{-2})	1.14×10^{0}	-6.80×10^{-1}	2.86×10^{-3}	-2.56×10^{-3} (2.23×10^{-2})	1.14×10^{0}	-5.46×10^{-1}	2.84×10^{-3}
Phosp-hate	3.08×10^{-4} (1.41×10^{-2})	1.65	-2.17	8.44×10^{-3}	-1.54×10^{-4} (6.21×10^{-2})	2.90	-3.12×10^{0}	3.90×10^{-2}	-1.61×10^{-4} (6.09×10^{-2})	2.57×10^{0}	-3.50×10^{0}	3.61×10^{-2}
Nitrate	2.20×10^{-3} (3.08×10^{-1})	1.89×10^{1}	-4.13×10^{1}	1.30×10^{-2}	-5.43×10^{-3} (8.58×10^{-1})	4.23×10^{1}	-3.45×10^{1}	3.62×10^{-2}	-7.67×10^{-3} (8.14×10^{-1})	3.04×10^{1}	-4.24×10^{-1}	3.43×10^{-2}
Silicate	2.27×10^{-2} (1.18)	5.92×10^{1}	-5.85×10^{1}	2.08×10^{-2}	-6.60×10^{-2} (3.19)	8.42×10^{1}	-2.08×10^{2}	5.57×10^{-2}	-4.28×10^{-2} (3.01)	8.20×10^{1}	-2.08×10^{2}	5.25×10^{-2}
Oxygen	3.98×10^{-3} (2.11)	3.08×10^{2}	-2.31×10^{2}	1.10×10^{-2}	6.06×10^{-2} (8.90)	3.28×10^{2}	-4.23×10^{2}	4.66×10^{-2}	6.32×10^{-2} (8.70)	3.21×10^{2}	-3.75×10^{2}	4.55×10^{-2}

Deleted: , calculated using GLODAPv2.2022 with only World Ocean Circulation Experiment data quality control flag categories of 2 (Acceptable) and secondary quality control flag categories of 1 (subjected to full secondary quality control) for all possible input and measurement data, and for salinities between 30–37…





Deleted:

330 **Figure 2: Difference between Python and MATLAB locally interpolated regression estimates (*y*-axis) compared to MATLAB estimates (*x*-axis) for open ocean ($_o$) data and all equations combined for TA (a, 13,384,096 total estimates from all equations), DIC (b, 13,384,096 estimates), pH$_T$ (c, 13,384,096 estimates), phosphate (d, 13,384,096 estimates), nitrate (e, 12,718,592 estimates), silicate (f, 12,640,896), and oxygen (g, 12,757,792 estimates; *n*=306,227 for TA, 343,580 for DIC, 199,304 for pH$_T$, and 764,301 for phosphate, nitrate, silicate, and oxygen). Units for all except pH$_T$ are in µmol kg^{-1}. Top and bottom side histograms represent the distribution of the x and y axes, respectively. Note the differences in *x*- and *y*-axes scales. RMSE*n* is the normalized root mean square error, or the RMSE of all divided by the mean of all estimates.**

Deleted:) calculated from GLODAPv2.2022 with only World Ocean Circulation Experiment data quality control flag categories of 2 (Acceptable) and secondary quality control flag categories of 1 (subjected to full secondary quality control) for all possible input and measurement data, and for salinities between 30–37 (

PyESPER_LIRs were within 2σ (~95% of measurements should fall within this uncertainty level) for most ocean regions, with a few exceptions which occurred predominantly in coastal areas or deep waters near the edges of the original MATLAB grid (Figs. 3 and 4). Spatial patterns in distribution of outliers shown in Fig. 4 appear to reflect locations where more edge-of-grid biogeochemical measurements were collected (e.g., near coasts and in deep waters). Hence, these exceptionally different locations aligned well with places where coefficients were extrapolated in MATLAB for use in PyESPER_LIRs, compared to interpolations with far away "dummy points" within MATLAB ESPER_LIRs (see Sect. 2.1.1, "*Locally interpolated regressions*"; Figs. 3, 4, and 5; for $_{w}$ Fig. B2 and B3). Within regions where MATLAB and Python were interpolating similarly, far outliers were uncommon (Figs. 3, 4, 5, B2, and B3). When ESPER_LIR and PyESPER_LIR were applied to temperature and salinity from the Roemmich and Gilson climatology for the year 2023 (Roemmich and Gilson, 2009), patterns of surface DIC distribution were similar with a few minor nuances (Fig. C1). Notably, low DIC estimates covered a broader spatial extent in the western equatorial Pacific and Indian Oceans for PyESPER_LIR estimates, and PyESPER_LIR appeared to have a slightly low bias in some places relative to ESPER_LIR. Beyond these minor differences, the mapped DIC demonstrates the similarity of the data products' functionality in an applied setting. While ESPER_LIR and PyESPER_LIR do not produce quantitatively identical estimates, it should be noted that both routines perform similarly well at reconstructing the GLODAPv2.2022 data product (Table 1; for $_{w}$ Table B1). These routines should not be considered identical but are comparable.

Deleted: T

Deleted: locations

Deleted: in the MATLAB implementation

Deleted: was

Deleted: 6

3.1.2 Neural networks

When compared to the ESPER_NN results for the open ocean ($_{o}$) GLODAPv2.2022 dataset, all equation-case and desired outcome variable combinations from PyESPER_NN (PyESPER − ESPER_NN estimates) resulted in mean differences of <±0.004 (Table 2), a much smaller difference than for LIR comparisons. Mean (±standard deviation; RMSE$_n$) offset for each property is shown in Table 2. Since a very wide range of input data were used, a wide range of estimates were produced from both ESPER_NNs and PyESPER_NNs for all variables (Fig. 6), representing the high variability that can be found in the oceans (especially coastal regions, some of which were included in the "open ocean" dataset due to having salinities between 30–37 and quality-controlled data). Both PyESPER_NN and ESPER_NN results were nearly identical, even when outlier results were obtained from unusual input data from environments where ESPERs are not recommended for use (for example, resulting in negative DIC estimates in Fig. B4; see also Table B2). The largest relative disagreements were found for DIC and pH_T, though these disagreements remained small relative to measurement uncertainties. These minor offsets are attributed to the programming language differences in the interpolation of the C_{ant} adjustment, which is only applied to these two properties.

Deleted: TA for PyESPER_NN – ESPER_NN was -4.5x10^{-12} µmol kg^{-1} (±5.9x10^{-9} µmol kg^{-1}; RMSE$_n$=2.5x10^{-12}), DIC was -3.0x10^{-3} µmol kg^{-1} (±9.3x10^{-2} µmol kg^{-1}; RMSE$_n$=4.2x10^{-5}), pH_T was 1.1x10^{-5} (±2.3x10^{-4}; RMSE$_n$=3.0x10^{-5}), phosphate was -6.2x10^{-14} µmol kg^{-1} (±6.6x10^{-11} µmol kg^{-1}; RMSE$_n$=1.3x10^{-10}), nitrate was -7.8x10^{-13} µmol kg^{-1} (±8.9x10^{-10} µmol kg^{-1}; RMSE$_n$=3.8x10^{-11}), silicate was -1.2x10^{-12} µmol kg^{-1} (±2.0x10^{-9} µmol kg^{-1}; RMSE$_n$=3.4x10^{-11}), and oxygen was -4.4x10^{-13} µmol kg^{-1} (±2.1x10^{-10} µmol kg^{-1}; RMSE$_n$=1.1x10^{-12};

Deleted:)

Deleted: 7

Deleted: larger

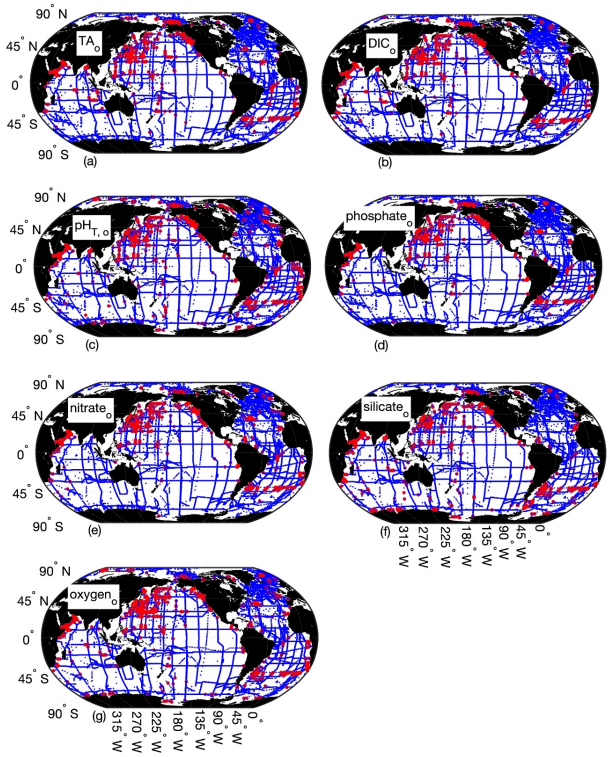


Figure 3: Map of differences between Python and MATLAB ESPER locally interpolated regression estimates (total estimates *n*=13,384,096 for TA (a), DIC (b), pH_T (c), and phosphate, 12,718,592 for nitrate (d), 12,640,896 for silicate (e), and 12,757,792 for oxygen (f)) for the open ocean ($_o$), where small blue circles represent differences <2 x uncertainties of the MATLAB estimates (*n*=13,344,924 for TA, 13,354,980 for DIC, 13,349,438 for pH_T, 13,357,843 for phosphate, 12,688,861 for nitrate, 12,597,608 for silicate, and 12,721,483 for oxygen), and red circles represent differences >2 x uncertainties of the MATLAB estimates (*n*=39,172 for TA, 29,116 for DIC, 34,658 for pH_T, 26,253 for phosphate, 29,731 for nitrate, 43,288 for silicate, and 36,309 for oxygen).

Deleted:)

Deleted: Open ocean estimates are calculated from GLODAPv2.2022 with only World Ocean Circulation Experiment data quality control flag categories of 2 (Acceptable) and secondary quality control flag categories of 1 (subjected to full secondary quality control) for all possible input and measurement data, and for salinities between 30–37.

405

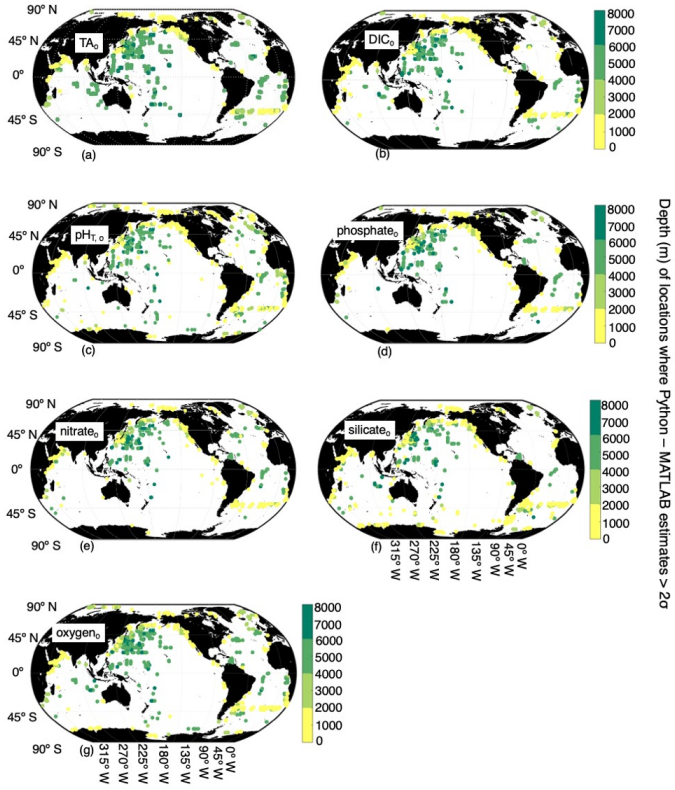


Figure 4: Map of locations and depths (colorbar) where differences between Python and MATLAB ESPER locally interpolated regression estimates are greater than 2 x the estimate uncertainties for the open ocean ($_o$, n=13,344,924 for TA (a), 13,354,980 for DIC (b), 13,349,438 for pH_T (c), 13,357,843 for phosphate (d), 12,688,861 for nitrate (e), 12,597,608 for silicate (f), and 12,721,483 for oxygen (g)).

410

Deleted: Open ocean estimates are calculated from GLODAPv2.2022 with only World Ocean Circulation Experiment data quality control flag categories of 2 (Acceptable) and secondary quality control flag categories of 1 (subjected to full secondary quality control) for all possible input and measurement data, and for salinities between 30–37.

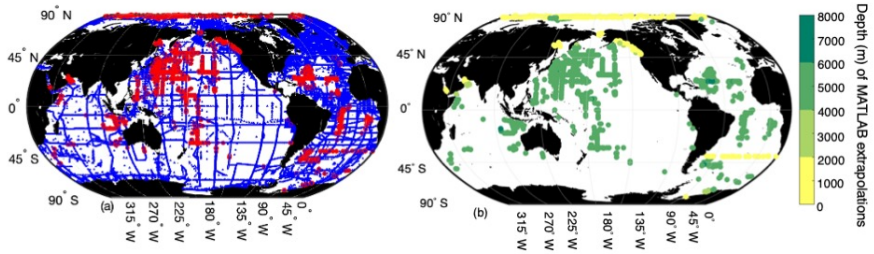


Figure 5: Map of locations where MATLAB was interpolating (*n*=1,365,170, blue) and extrapolating (*n*=16,078, red) from the grid to GLODAPv2.2022 data (a) and depth of extrapolations (b).

3.1.3 Anthropogenic carbon estimates

Although inconsistencies in results occur between Python and MATLAB when interpolating (same issue noted in Sect. 2.1.4, "*Anthropogenic carbon*"), anthropogenic carbon (C_{ant}) estimates were similar between the two versions of ESPER. This was demonstrated by differences in DIC and pH_T estimates for NNs, which only interpolate when estimating the contribution of C_{ant} to estimates (Fig. 6). The next generation of ESPER updates will include a new method for estimating C_{ant} (Tracer-Based Rapid Anthropogenic Carbon Estimation, or TRACEv1; Carter et al., *submitted*), which uses neural networks and should eliminate the need for interpolation. Currently, when C_{ant} estimates are required, the results from PyESPER_NNs remain functionally identical to those from ESPER_NNs, despite minor offsets from the interpolation methods.

3.2 Speed of calculation

PyESPERs take considerably longer than ESPERs to produce estimates. On a MacBook Air using Python in the terminal with standard internet connection, PyESPER_NN produced results 4–1070 x slower than ESPER_NN, while PyESPER_LIR produced results about 16–310 x slower than ESPER_LIRs, with magnitude of the slowdown dependent upon the number of variable inputs and equation cases requested and number of estimates required (Table 3). ESPER_NNs were the fastest to execute, and took <2 s for all time tests, even when large datasets and all variable-equation case scenarios were requested. ESPER_LIRs were the next-fastest, requiring <33 s for all time tests, followed by PyESPER_NNs, which typically required 0–11 s to execute, but required 841 s (14 min) for running large datasets and all variable-equation case scenarios. PyESPER_LIRs were the slowest, and typically required 13–400 s to execute, but the longest scenario required 5382 s (90 min; Table 3). It is possible that this code can be further optimized for speed in future updates.

Deleted: 7
Deleted: are
Deleted: **Timing**
Deleted: Jupyter Notebook
Deleted: 0
Deleted: 1500
Formatted: Not Highlight
Formatted: Not Highlight
Deleted: 7–500
Formatted: Not Highlight
Deleted: 5
Deleted: 15
Deleted: >1400
Deleted: 23
Deleted: 22
Deleted: 500
Deleted: 7530
Deleted: 125
Formatted: Not Highlight
Deleted: ¶ ... [1]

ble 2: Mean (standard deviation), maximum, minimum, and normalized RMSE ($RMSE_n$) are shown for three scenarios: (1) between thon – MATLAB NNs, (2) MATLAB ESPER_NN – measured values, and (3) PyESPER_NN – measured values. Separate rows exist for ı, DIC, pH_T, phosphate, nitrate, silicate, and oxygen estimates. All units except pH_T are µmol kg^{-1} and data are for open oceans ($_o$) and equations combined.

	Python – MATLAB				MATLAB - Measured				Python - Measured			
	Mean	**Max**	**Min**	**$RMSE_n$**	**Mean**	**Max**	**Min**	**$RMSE_n$**	**Mean**	**Max**	**Min**	**$RMSE_n$**
TA	-4.49×10^{-12} (5.89×10^{-9})	4.00×10^{-6}	-2.00×10^{-6}	2.53×10^{-12}	3.40×10^{-1} (5.21)	8.15×10^{2}	-1.78×10^{2}	2.24×10^{-3}	3.40×10^{-1} (5.21)	8.15×10^{2}	-1.78×10^{2}	2.24×10^{-3}
DIC	-3.01×10^{-3} (9.29×10^{-2})	2.31	-3.69	4.22×10^{-5}	-2.94×10^{-1} (7.67)	6.17×10^{2}	-3.37×10^{2}	3.49×10^{-3}	-2.97×10^{-1} (7.67)	6.18×10^{2}	-3.37×10^{2}	3.49×10^{-3}
pH_T	1.07×10^{-5} (2.34×10^{-4})	5.60×10^{-3}	-7.65×10^{-3}	2.98×10^{-5}	-4.59×10^{-3} (1.59×10^{-2})	4.71×10^{-1}	-6.58×10^{-1}	2.10×10^{-3}	-4.58×10^{-3} (1.59×10^{-2})	4.71×10^{-1}	-6.58×10^{-1}	2.10×10^{-3}
Phosphate	-6.19×10^{-14} (6.60×10^{-11})	2.50×10^{-8}	-1.25×10^{-7}	1.31×10^{-10}	1.15×10^{-3} (5.19×10^{-2})	2.12	-2.81	3.06×10^{-2}	1.15×10^{-3} (5.19×10^{-2})	2.12	-2.81	3.06×10^{-2}
Nitrate	-7.80×10^{-13} (8.91×10^{-10})	1.35×10^{-7}	-2.28×10^{-6}	3.76x10-11	-2.24×10^{-3} (7.17×10^{-1})	4.06×10^{1}	-3.40×10^{1}	2.93×10^{-2}	-2.24×10^{-3} (7.17×10^{-1})	4.06×10^{1}	-3.40×10^{1}	2.93×10^{-2}
Silicate	-1.24×10^{-12} (1.98×10^{-9})	2.11×10^{-7}	-2.97×10^{-6}	3.42×10^{-11}	4.96×10^{-3} (2.55)	1.23×10^{2}	-8.29×10^{1}	4.42×10^{-2}	4.96×10^{-3} (2.55)	1.23×10^{2}	-8.29×10^{1}	4.42×10^{-2}
Oxygen	-4.42×10^{-13} (2.09×10^{-10})	1.00×10^{-8}	-1.00×10^{-7}	1.06×10^{-12}	5.33×10^{-2} (7.29)	3.54×10^{2}	-2.06×10^{2}	3.82×10^{-2}	5.33×10^{-2} (7.29)	3.54×10^{2}	-2.06×10^{2}	3.82×10^{-2}

Deleted: , for
Deleted: differences
Deleted: MATLAB and
Deleted: and
Deleted: and
Deleted: for
Deleted: (all
Deleted:)
Deleted: data
Deleted: , calculated using GLODAPv2.2022 with only World Ocean Circulation Experiment data quality control flag categories of 2 (Acceptable) and secondary quality control flag categories of 1 (subjected to full secondary quality control) for all possible input and measurement data, and for salinities between 30–37 (n=306,227 for TA, 343,580 for DIC, 199,304 for pH_T, 764,301 for phosphate, 764,301 for nitrate, 764,301 for silicate, and 764,301 for oxygen)

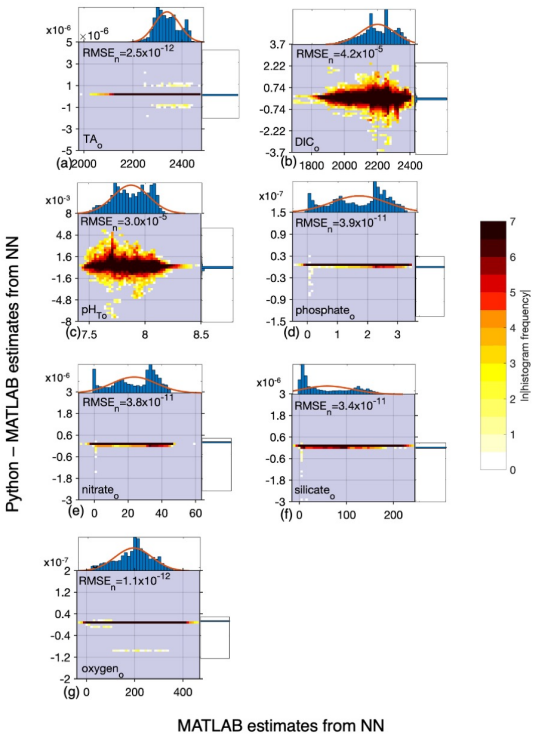


Figure 6: Difference between Python and MATLAB neural network estimates (*y*-axis) compared to MATLAB estimates (*x*-axis) for open ocean ($_o$) data and all equations combined for TA (a, 4,899,512 total estimates from all equations), DIC (b, 5,497,004 estimates), pH_T (c, 3,188,864 estimates), phosphate (d, 12,228,432 estimates), nitrate (e, 12,228,432 estimates), silicate (f, 12,228,432 estimates), and oxygen (g, 12,228,560 estimates; *n*=306,227 for TA, 343,580 for DIC, 199,304 for pH_T, and 764,301 for phosphate, nitrate, silicate, and oxygen). Units for all except pH_T are in µmol kg^{-1}. Top and bottom side histograms represent the distribution of the x and y axes, respectively. Note the differences in *x*- and *y*-axes scales. $RMSE_n$ is the normalized root mean square error, or the RMSE divided by the mean of all estimates from MATLAB_NN.

Deleted: 7

Deleted:)

Deleted: calculated from GLODAPv2.2022 with only World Ocean Circulation Experiment data quality control flag categories of 2 (Acceptable) and secondary quality control flag categories of 1 (subjected to full secondary quality control) for all possible input and measurement data, and for salinities between 30–37 (

Table 3: Time required to produce estimates for PyESPERv1.0.0s and ESPERs (LIRs and NNs) for different desired variable, equation-case, and number of estimates scenarios.

Variable	Equation(s)	Number of Estimates	PyESPER_NN time (s)	ESPER_NN time (s)	PyESPER_LIR time (s)	ESPER_LIR time (s)
TA	1	10	0.06	0.01	13.26	0.77
TA	1	100	0.12	0.01	13.10	0.60
TA	2	100	0.13	0.01	12.94	0.79
TA	3	100	0.14	0.01	13.05	0.81
TA	4	100	0.14	0.01	12.99	0.78
TA	5	100	0.12	0.00	13.01	0.75
TA	6	100	0.12	0.01	12.97	0.79
TA	7	100	0.13	0.00	12.98	0.78
TA	8	100	0.12	0.00	13.04	0.80
TA	9	100	0.14	0.00	13.01	0.81
TA	10	100	0.11	0.01	13.04	0.74
TA	11	100	0.13	0.00	12.99	0.74
TA	12	100	0.13	0.00	13.02	0.75
TA	13	100	0.15	0.00	13.00	0.79
TA	14	100	0.12	0.01	13.87	0.77
TA	15	100	0.12	0.00	12.94	0.84
TA	16	100	0.11	0.01	13.02	0.74
TA	1-16	100	1.49	0.04	192.23	0.62
TA	1	1000	0.68	0.03	16.11	0.76
TA	1	10,000	7.70	0.12	41.76	0.83
TA	1	100,000	69.47	0.62	312.72	1.55
DIC	1	100	6.44	1.55	19.55	2.69
DIC	1-16	100	8.09	1.53	198.43	1.54
pH	1	100	6.50	0.06	31.83	0.81
pH	1-16	100	11.62	0.46	299.08	3.41
Phosphate	1	100	0.11	0.01	13.17	3.39
Phosphate	1-16	100	1.74	0.06	192.27	0.80
Nitrate	1	100	0.11	0.01	13.10	0.74
Nitrate	1-16	100	1.74	0.05	192.30	3.56
Silicate	1	100	0.11	0.02	13.18	3.64
Silicate	1-16	100	1.75	0.04	193.37	0.82
Oxygen	1	100	0.11	0.01	13.46	0.78
Oxygen	1-16	100	1.75	0.04	192.74	2.15
All Variables	1	100	7.16	0.01	108.12	13.86
All Variables	1	10,000	106.98	0.10	387.01	15.17
All Variables	1-16	100	49.53	0.09	3182.56	15.26
All Variables	1-16	10,000	841.01	1.67	5382.32	32.13

Deleted: 1…01 ... [2]
Deleted: 6.55
Deleted: 22.35
Deleted: 193.16
Deleted: 5.87
Deleted: 19.98
Deleted: 207.68
Deleted: 5.82
Deleted: 25.90
Deleted: 204.92
Deleted: 5.79
Deleted: 22.82
Deleted: 216.22
Deleted: 5.90
Deleted: 24.01
Deleted: 209.98
Deleted: 5.80
Deleted: 23.60
Deleted: 194.39
Deleted: 5.88
Deleted: 22.42
Deleted: 194.30
Deleted: 5.88
Deleted: 23.03
Deleted: 201.98
Deleted: 5.84
Deleted: 22.51
Deleted: 195.30
Deleted: 5.87
Deleted: 22.42
Deleted: 195.42
Deleted: 5.82
Deleted: 22.60
Deleted: 194.58
Deleted: 5.84
Deleted: 22.28
Deleted: 193.33
Deleted: 5.90
Deleted: 22.43
Deleted: 201.39
Deleted: 5.88
Deleted: 22.37
Deleted: 5.82
Deleted: 22.46
Deleted: 5.81
Deleted: 22.35
Deleted: 5.81
Deleted: 22.57
Deleted: 11.06
Deleted: 312.13
Deleted: 11.50
Deleted: 29.69
Deleted: 61.54
Deleted: 57.59

3.3 Future improvements

Updated ESPERs will be trained and assessed using GLODAPv2.2023 (or later versions), which includes 1108 cruises (compared to 946 cruises from GLODAPv2.2020, the current data product used. Additionally, future ESPERs will incorporate depth (z) as an optional predictor variable for consistency with LIPHR, a prior version for estimating pH_T (Carter et al., 2017). The implementation of updated C_{ant} estimation methods should additionally improve the accuracy and efficiency of both ESPERs and PyESPERs when C_{ant} estimates are required. Future versions of ESPER written in MATLAB may be modified to improve interoperability with the Python implementation (i.e., to ensure the interpolation routines are identical in all instances between languages).

4 Data Availability

Data used for reconstruction and estimate comparisons is available through GLODAP (https://glodap.info; see Lauvset et al., 2022, doi:10.5194/essd-14-5543-2022 and Olsen et al., 2020, doi:10.5194/essd-12-3653-2020). The temperature and salinity gridded climatology created by Roemmich & Gilson (2009), doi:10.1016/j.pocean.2009.03.004 was created with data from the Argo Program.

5 Code Availability

PyESPERv1.0.0, affiliated files, and analyses files are available through LMD's GitHub page (https://github.com/LarissaMDias) and archived through Zenodo (doi: 10.5281/zenodo.15133085). Updates to PyESPERv1.0.0 will also be published through LMD's GitHub page and archived through Zenodo. ESPERs (Carter, 2021) and original associated files used in creation of PyESPERv1.0.0 are available at BRC's GitHub page at https://github.com/BRCScienceProducts. Input data used for comparisons are available through the GLODAP website (https://glodap.info).

6 Conclusions

A near-replicate of ESPERs has been produced in the freely available Python programming language. This algorithm data product will allow Python users or researchers with limited funds an alternate, free method for using ESPERS (other than the proprietary MATLAB), increasing the accessibility of the original ESPER algorithms. The same logic applied to the original MATLAB ESPERs was applied within the Python coding language (PyESPERs, version 1.0.0), and results have demonstrated comparability to ESPER estimates. Estimates from PyESPER_NNs precisely align with those from ESPER_NNs for all equations and desired outcome variable combinations (Fig. 6) and estimates from these two routines align very closely for all estimates, and to within machine precision for all but pH_T and DIC, which exhibit slight differences due to impacts of

Deleted: directions

Formatted: Font: (Default) +Body (Times New Roman), 10 pt

Deleted: 1

Deleted: 1 preliminary Jupyter Notebook

Deleted:

Deleted: 1

Deleted: 1

Deleted: 1

Deleted: 1

Deleted: 1

Deleted: 7

Deleted: are essentially identical.

Formatted: Subscript

Formatted: Font: 10 pt, Not Bold, Font color: Auto

interpolating for C_{ant}. PyESPER_LIR estimates differ from ESPER_LIR estimates for some coastal and deep-water regions between the two coding languages due to triangulation, extrapolation, and interpolation differences, but were more similar throughout all portions of the open ocean (Figs. 2, 3, and 4). Notably, PyESPER_LIR performs equivalently to ESPER_LIR when reconstructing the training data from GLODAPv2.v2022, so estimates produced from these two routines should be considered comparable rather than identical. Nevertheless, we do not recommend using PyESPER_LIR in coastal or deep (>5500 m) waters when primarily interested in comparing results with those of the MATLAB implementation of ESPER_LIR. Future updates to ESPERs will include updates to PyESPERs, with adjustments to allow for greater consistency and speed.

7 Appendices

Appendix A: ESPER specifications

Sets of equations, predictor variables, and measurement uncertainties used in ESPER and PyESPER (adapted from Carter et al., 2021) are shown below.

Table A1: Input predictor variable combinations used for each ESPER equation (adapted from Carter et al., 2021), where *S* is salinity, *T* is temperature, and *A*, *B*, and *C* are defined in Table A2 (below).

Equation Number	Predictor Variables
1	*S, T, A, B, C*
2	*S, T, A, C*
3	*S, T, B, C*
4	*S, T, C*
5	*S, T, A, B*
6	*S, T, A*
7	*S, T, B*
8	*S, T*
9	*S, A, B, C*
10	*S, A, C*
11	*S, B, C*
12	*S, C*
13	*S, A, B*
14	*S, A*
15	*S, B*
16	*S*

Deleted: S2

735

740

Table A2: Input predictor variables (*A*, *B*, and *C*) for each estimated property (adapted from Carter et al., 2021).

Estimated Property	*A*	*B*	*C*
TA	Nitrate	Oxygen	Silicate
DIC	Nitrate	Oxygen	Silicate
pH_T	Nitrate	Oxygen	Silicate
Phosphate	Nitrate	Oxygen	Silicate
Nitrate	Phosphate	Oxygen	Silicate
Silicate	Phosphate	Oxygen	Nitrate
Oxygen	Phosphate	Nitrate	Silicate

Table A3: Default measurement uncertainties ($E_{Pi\_Default}$) for ESPERs and PyESPERs (adapted from Carter et al., 2021), where θ is potential temperature.

Property	Units	Uncertainty
S	–	0.003, absolute
θ	°C	0.003, absolute
Phosphate	µmol kg^{-1}	2%, relative
Nitrate	µmol kg^{-1}	2%, relative
Silicate	µmol kg^{-1}	2%, relative
Oxygen	µmol kg^{-1}	1%, relative

745 **Appendix B: Comparison using entire GLODAPv2.2022**

Results of comparisons of PyESPER with ESPER for the entire GLODAPv2.2022 dataset, including the entire oceanic and coastal salinity range and data of all quality control flag categories are shown below.

Deleted: ¶ ... [3]

ble B1: Mean (standard deviation), maximum, minimum, and normalized RMSE ($RMSE_n$), for differences between MATLAB and thon LIRs, ESPER_LIR and measured values, and PyESPER_LIR and measured values for TA, DIC, pH_T, phosphate, nitrate, silicate, d oxygen estimates (all units except pH_T are µmol kg^{-1}) for all equations combined, from the entire GLODAPv2.2022 (w/ n=1,381,248).

Deleted:), including NaN's, coastal data, and all data quality control flag categories (where necessary input data were available,

	Python – MATLAB				MATLAB - Measured				Python - Measured			
	Mean	**Max**	**Min**	**$RMSE_n$**	**Mean**	**Max**	**Min**	**$RMSE_n$**	**Mean**	**Max**	**Min**	**$RMSE_n$**
TA	-2.76×10^{-2} (2.61)	6.46×10^{2}	-6.98×10^{2}	1.12×10^{-3}	3.36×10^{-1} (1.19×10^{1})	1.19×10^{3}	-7.34×10^{2}	5.12×10^{-3}	3.22×10^{-1} (1.21×10^{1})	1.19×10^{3}	-7.34×10^{2}	5.22×10^{-3}
DIC	-5.35×10^{-3} (2.82)	5.17×10^{2}	-7.46×10^{2}	1.29×10^{-3}	-6.85×10^{-1} (1.63×10^{1})	9.79×10^{2}	-1.75×10^{3}	7.46×10^{-3}	-6.69×10^{-1} (1.63×10^{1})	9.79×10^{2}	-1.75×10^{3}	7.45×10^{-3}
pH	-6.05×10^{-5} (6.74×10^{-3})	7.65×10^{-1}	-1.59×10^{0}	8.52×10^{-4}	2.03×10^{-3} (3.39×10^{-2})	3.24×10^{0}	-1.39×10^{0}	4.29×10^{-3}	2.14×10^{-3} (3.37×10^{-2})	2.85×10^{0}	-1.40×10^{0}	4.26×10^{-3}
Phosp-hate	8.88×10^{-5} (2.02×10^{-2})	3.96	-3.31	1.27×10^{-2}	-1.04×10^{-3} (9.26×10^{-2})	4.77	-1.37×10^{1}	5.74×10^{-2}	-9.66×10^{-4} (9.27×10^{-2})	4.66	-1.39×10^{1}	5.74×10^{-2}
Nitrate	-7.07×10^{-4} (4.37×10^{-1})	6.18×10^{1}	-6.35×10^{1}	1.99×10^{-2}	-7.11×10^{-3} (1.27)	8.17×10^{1}	-1.82×10^{2}	5.70×10^{-2}	-7.79×10^{-3} (1.29)	7.55×10^{1}	-1.76×10^{2}	5.76×10^{-2}
Silicate	3.78×10^{-3} (2.24)	5.49×10^{2}	-5.09×10^{2}	4.39×10^{-2}	-1.22×10^{-1} (4.06)	3.11×10^{2}	-1.11×10^{3}	7.76×10^{-2}	-1.16×10^{-1} (4.03)	1.16×10^{2}	-5.61×10^{2}	7.77×10^{-2}
Oxygen	-4.47×10^{-2} (3.20)	4.28×10^{2}	-5.78×10^{2}	1.58×10^{-2}	2.33×10^{-1} (1.27×10^{1})	1.33×10^{3}	-9.02×10^{2}	6.37×10^{-2}	1.87×10^{-1} (1.23×10^{1})	9.69×10^{2}	-9.02×10^{2}	6.19×10^{-2}

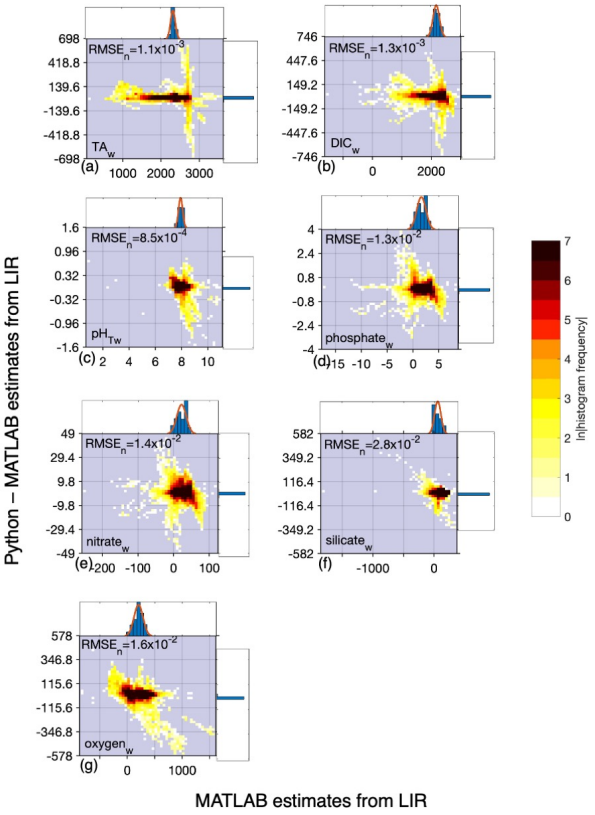


Figure B1: Difference between Python and MATLAB locally interpolated regression estimates (*y*-axis) compared to MATLAB estimates (*x*-axis) for whole ocean ($_w$) data and all equations combined (22,099,968 total estimates from all equations for each variable), for TA (a), DIC (b), pH_T (c), phosphate (d), nitrate (e), silicate (f), and oxygen (g) derived using all equations and calculated from entire GLODAPv2.2022 data (n=1,381,248). Units for all except pH_T are in µmol kg^{-1}. Top and bottom side histograms represent the distribution of the x and y axes, respectively. Note the differences in *x*- and *y*-axes scales. RMSE*n* is the normalized root mean square error, or the RMSE of all divided by the mean of all MATLAB estimates. The large range of sometimes unrealistic estimates along the x-axis can be attributed to anomalous and sometimes erroneous input data used for predictions.

Deleted: , including NaN's, coastal data, and all data quality control flag categories

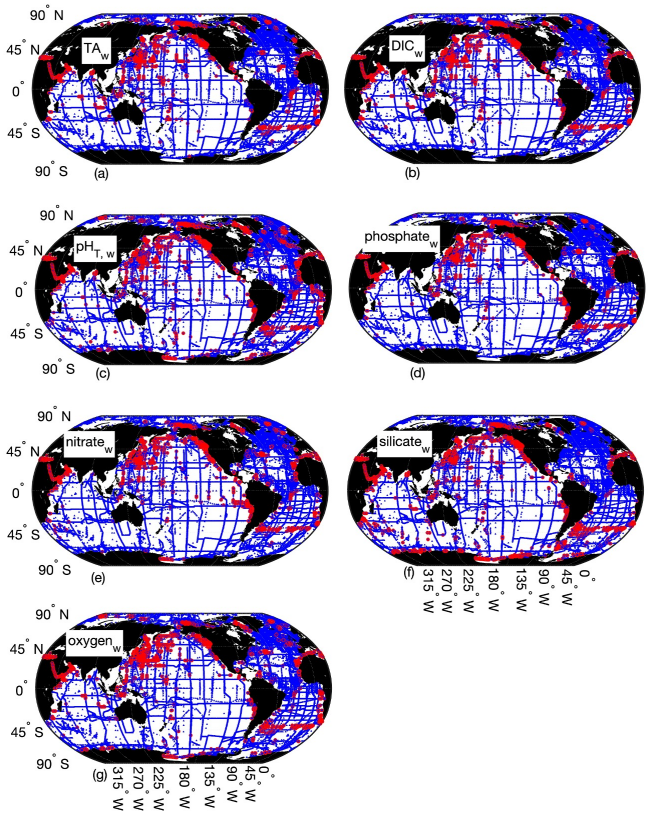
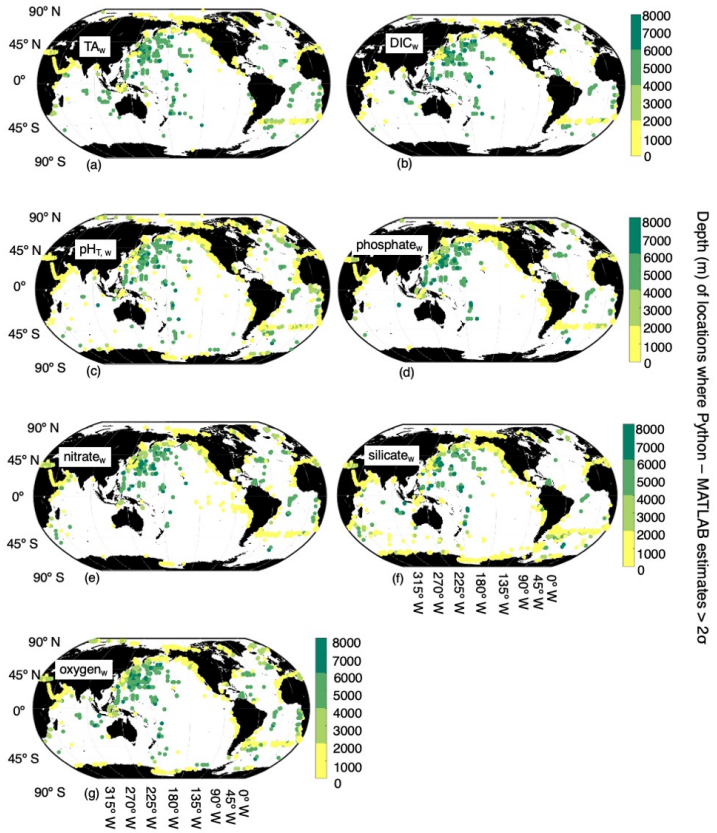
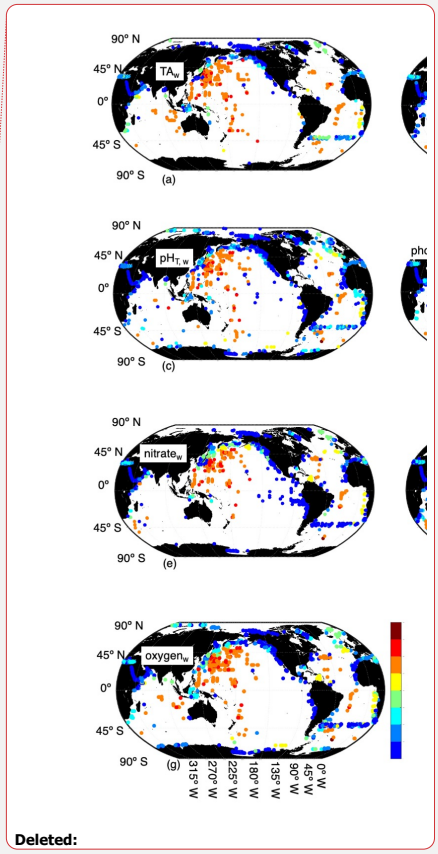


Figure B2: Map of differences between Python and MATLAB ESPER locally interpolated regression estimates (total estimates *n*=22,099,968 for all variables) for the whole ocean ($_w$), where small blue circles represent differences <2 x uncertainties of MATLAB estimates (*n*=22,034,967 for TA (a), 22,054,048 for DIC (b), 22,045,316 for pH_T (c), 22,057,220 for phosphate (d), 22,045,770 for nitrate (e), 22,024,674 for silicate (f), and 22,045,827 for oxygen (g)), and red circles represent differences >2 x uncertainties of MATLAB estimates (*n*=65,001 for TA, 45,920 for DIC, 54,642 for pH, 42,748 for phosphate, 54,198 for nitrate, 75,294 for silicate, and 54,141 for oxygen; *n*=1,381,248).

Deleted:). Whole ocean estimates are calculated from the entire GLODAPv2.2022 dataset, including NaN's, coastal data, and all data quality control flag categories (





Deleted:

Figure B3: Map of locations and depths (colorbar) where differences between Python and MATLAB ESPER locally interpolated regression estimates are greater than 2 x the estimate uncertainties for the whole ocean ($_w$, n=22,034,967 for TA (a), 22,054,048 for DIC (b), 22,045,316 for pH_T (c), 22,057,220 for phosphate (d), 22,045,770 for nitrate (e), 22,024,674 for silicate (f), and 22,045,827 for oxygen (g); n=1,381,248).

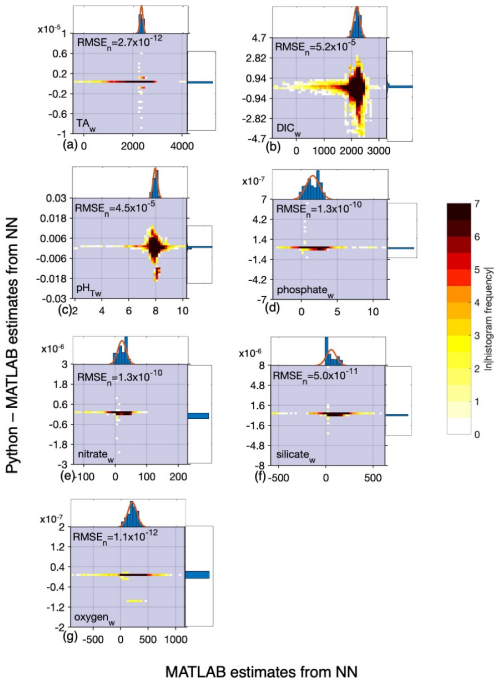
Deleted:). Whole ocean estimates are calculated from the entire GLODAPv2.2022 dataset, including NaN's, coastal data, and all data quality control flag categories (

ble B2: Mean (standard deviation), maximum, minimum, and normalized RMSE (RMSE$_n$), for differences between MATLAB and thon NNs, ESPER_NN and measured values, and PyESPER_NN and measured values for TA, DIC, pH$_T$, phosphate, nitrate, silicate, d oxygen estimates (all units except pH$_T$ are µmol kg^{-1}) for all equations combined, from the entire GLODAPv2.2022 dataset (where cessary input data were available, *n*=1,381,248).

Deleted:), including NaN's, coastal data, and all data quality control flag categories

Deleted: (

	Python – MATLAB				MATLAB - Measured				Python - Measured			
	Mean	**Max**	**Min**	**RMSE$_n$**	**Mean**	**Max**	**Min**	**RMSE$_n$**	**Mean**	**Max**	**Min**	**RMSE$_n$**
TA	-6.35×10^{-12} (6.24×10^{-9})	6.00×10^{-6}	-9.00×10^{-6}	2.69×10^{-12}	4.99×10^{-1} (1.46×10^{1})	2.12×10^{3}	-2.24×10^{3}	6.30×10^{-3}	4.99×10^{-1} (1.46×10^{1})	2.12×10^{3}	-2.24×10^{3}	6.30×10^{-3}
DIC	-3.24×10^{-3} (1.13×10^{-1})	2.88	-4.68	5.19×10^{-5}	-4.82×10^{-1} (1.75×10^{1})	1.97×10^{3}	-2.22×10^{3}	8.01×10^{-3}	-4.82×10^{-1} (1.75×10^{1})	1.97×10^{3}	-2.22×10^{3}	8.01×10^{-3}
pH$_T$	6.08×10^{-6} (3.58×10^{-4})	1.21×10^{-2}	-2.03×10^{-2}	4.52×10^{-5}	-3.01×10^{-3} (3.54×10^{-3})	2.53	-5.74	4.49×10^{-3}	-3.00×10^{-3} (3.54×10^{-2})	2.53	-5.74	4.49×10^{-3}
Phosp-hate	6.32×10^{-14} (2.08×10^{-10})	6.39×10^{-7}	-1.25×10^{-7}	1.31×10^{-10}	-5.84×10^{-4} (8.25×10^{-2})	1.14×10^{1}	-6.02	5.06×10^{-2}	-5.84×10^{-4} (8.25×10^{-2})	1.14×10^{1}	-6.02	5.06×10^{-2}
Nitrate	5.12×10^{-13} (3.06×10^{-9})	1.17×10^{-5}	-2.28×10^{-6}	1.39×10^{-10}	-1.07×10^{-2} (1.17)	1.97×10^{2}	-1.45×10^{2}	5.06×10^{-2}	-1.07×10^{-2} (1.17)	1.97×10^{2}	-1.45×10^{2}	5.06×10^{-2}
Silicate	-2.35×10^{-13} (2.55×10^{-9})	7.46×10^{-6}	-2.97×10^{-6}	4.96×10^{-11}	-2.37×10^{-2} (3.71)	6.25×10^{2}	-7.32×10^{2}	7.06×10^{-2}	-2.37×10^{-2} (3.71)	6.25×10^{2}	-7.32×10^{2}	7.06×10^{-2}
Oxygen	-4.65×10^{-13} (2.15×10^{-10})	1.00×10^{-9}	-1.00×10^{-7}	1.06×10^{-12}	-3.46×10^{-3} (1.13×10^{1})	7.12×10^{2}	-1.22×10^{3}	5.65×10^{-2}	-3.46×10^{-2} (1.13×10^{1})	7.12×10^{2}	-1.22×10^{3}	5.65×10^{-2}



805

Figure B4: Difference between Python and MATLAB neural network estimates (*y*-axis) compared to MATLAB estimates (*x*-axis) for whole ocean ($_w$) data and all equations combined for TA (a, 17,802,134 total estimates from all equations), DIC (b, 17,802,134 estimates), pH_T (c, 17,799,566 estimates), phosphate (d, 17,802,134 estimates), nitrate (e, 17,395,954 estimates), silicate (f, 17,445,310 estimates), and oxygen (g, 17,220,360 estimates) derived using all equations
810 **and calculated from entire GLODAPv2.2022 dataset (*n*=1,381,248). Units for all except pH_T are in µmol kg^{-1}. Top and bottom side histograms represent the distribution of the x and y axes, respectively. Note the differences in *x*- and *y*-axes scales. RMSE$_n$ is the normalized root mean square error, or the RMSE of all divided by the mean of all estimates. The large range of sometimes unrealistic estimates along the x-axis can be attributed to anomalous and sometimes erroneous input data used for predictions.**

Deleted: , including NaN's, coastal data, and all data quality control flag categories

Appendix C: Example of mapped DIC estimates from PyESPER and ESPER

Surface ocean DIC estimates form PyESPER_LIR and ESPER_LIR applied to the Roemmich and Gilson climatology (Roemmich and Gilson, 2009). Differences in surface ocean DIC between the two coding languages (c) illustrate the need to avoid using PyESPER_LIR for DIC in the surface ocean when comparing to MATLAB ESPER_LIR.

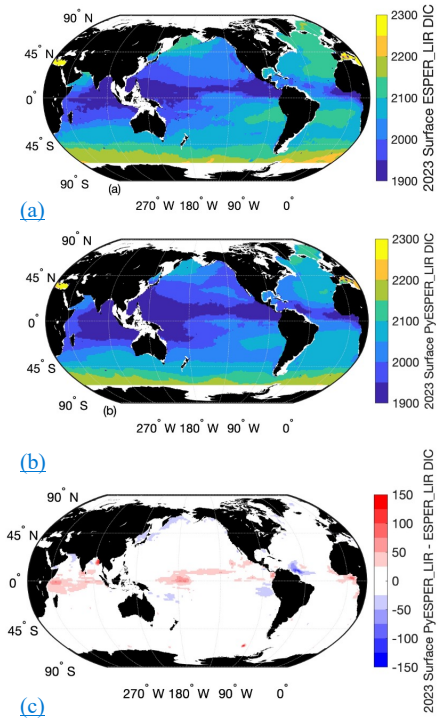


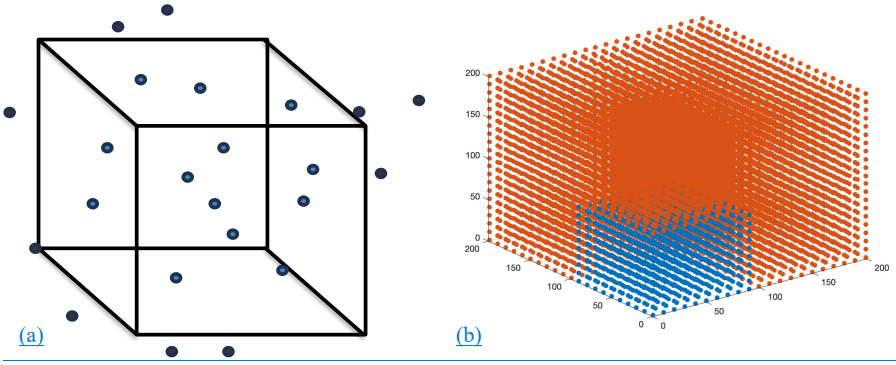
Figure C1: Maps of 2023 mean annual surface estimates of MATLAB ESPER_LIR DIC (a), Python PyESPER_LIR DIC (b), and PyESPER_LIR – ESPER_LIR_DIC (c; units are µmol kg^{-1}) from application of ESPERs to the Roemmich and Gilson Argo-based (Argo, 2000) climatology (Roemmich and Gilson, 2009).

Appendix D: Comparison of interpolation and extrapolation values between MATLAB and Python

Formatted: Font: Not Bold

MATLAB ESPER_LIRs avoid extrapolation by addition of a false set of data points at very far distances from the grid. However, when this method was implemented in Python, significant errors were introduced due to the differences in triangulation (which were both valid) between coding languages. Therefore, it was necessary to find another means of calculating extrapolations in PyESPER_LIRs which was more similar to those of ESPER_LIRs. We did this by producing a larger grid in MATLAB and reading that into Python. A simple demonstration of the errors introduced by this method is described below.

For this comparison we imagine a hypothetical cube, with x, y, and z coordinates, upon which we wish to provide estimates for a fourth variable (p) via both interpolation and extrapolation (Fig. D1a). We have created a random dataset of points and values within this cube for these demonstration purposes. We then followed the same procedure as in the PyESPER data product creation, whereby we extended this grid in three-dimensional space and used MATLAB scatteredInterpolant extrapolations to estimate values on the expanded grid (Fig. D1b). This method conducts a Delaunay triangulation, then uses both linear interpolation and extrapolation to estimate values. These extrapolated values were then used for interpolation only within Python using scipy's Delaunay and LinearNDInterpolator functions, which produced more consistent results than interpolation and extrapolation within Python.



(a) (b)

Figure D1: Hypothetical "grid" whereby estimates (p) interpolated within the grid are shown in blue and extrapolations are shown in red (a). Grid created for demonstration purposes, with interpolated values in blue and areas where we extrapolated values in red (b).

When interpolations within Python were compared to locations on the hypothetical grid where interpolations occurred in MATLAB also, results were more similar than those where the grid was extrapolated within MATLAB. This is because

Formatted: Font: Bold

Formatted: Font: Bold

Deleted: ¶

Formatted: Font: Not Bold

different, but equally valid, mathematics are used to interpolate and extrapolate. Namely, a triangulation is used as the basis for interpolations, whereas extrapolations are based on boundary gradients. Despite these differences, results were still more similar with this method between the two coding languages than when extrapolations were done in both Python and MATLAB.

Table D1: Comparison of differences between MATLAB interpolations and extrapolations and Python results (all interpolations).

	MATLAB Interpolation - Python Interpolation	MATLAB Extrapolation - Python Interpolation
Mean	0.0004	-0.6693
Standard Deviation	0.9559	5.2088
Max	2.2582	13.3083
Min	-2.4593	-15.6633

Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Centered
Formatted Table
Formatted: Centered
Formatted: Centered
Formatted: Centered
Formatted: Centered
Formatted: Font: Not Bold

8 Author Contributions

LMD was primarily responsible for Python data product development, validation, formal analysis, investigation, data curation, writing, and visualization. BRC primarily responsible for project conceptualization, MATLAB data product development, supervision, project administration, providing resources, funding acquisition, and editing. Methods were devised by both LMD and BRC.

9 Competing Interests

The authors declare that they have no conflict of interest.

10 Acknowledgments

The University of Washington Cooperative Institute for Climate, Ocean, and Ecosystem Studies (CICOES) has assigned CICOES Publication Contribution Number 2024-1382. The National Oceanic and Atmospheric Administration (NOAA) Pacific Marine Environmental Laboratory has assigned PMEL Contribution Number 5707. BRC and LMD thank the OAR Climate Program Office and NOAA's Global Ocean Monitoring and Observation program for support under award number NA21OAR4310251. The data used for DIC data products were collected and made freely available by the International Argo Program and the national programs that contribute to it (http://www.argo.ucsd.edu, http://argo.jcommops.org). The Argo Program is part of the Global Ocean Observing System. BRC and LMD also would like to sincerely thank Matthew

Humphreys, who served not only as a reviewer but also provided careful editing and help with packaging the PyESPER code.
BRC and LMD also thank Daniel Sandborn, who provided useful Python coding tips.

Deleted:

11 References

Álvarez, M., Velo, A., Tanhua, T., Key, R., and Van Heuven, S.: Carbon, tracer and ancillary data in the MEDsea, CARIMED: An internally consistent data product for the Mediterranean Sea, N43, 2019.

Argo: Argo float data and metadata from GLOBAL Data Assembly Centre (Argo GDAC), https://doi.org/10.17882/42182, 2000.

Bittig, H. C., Steinhoff, T., Claustre, H., Fiedler, B., Williams, N. L., Sauzède, R., Körtzinger, A., and Gattuso, J.-P.: An alternative to static climatologies: Robust estimation of open ocean CO<sub>2 variables and nutrient concentrations from T, S, and O<sub>2 data using Bayesian neural networks, Frontiers in Marine Science, 5, 1–29, https://doi.org/10.3389/fmars.2018.00328, 2018.

Bittig, H. C., Maurer, T. L., Plant, J. N., Wong, A. P., Schmechtig, C., Claustre, H., Trull, T. W., Bhaskar, T. V. S. U., Boss, E., Dall'Olmo, G., Organelli, E., Poteau, A., Johnson, K. S., Hanstein, C., Leymarie, E., Le Reste, S. L., Riser, S. C., Rupan, A. R., Taillandier, V., Thierry, V., and Xing, X.: A BGC-Argo guide: Planning, deployment, data handling and usage, Frontiers in Marine Science, 6:502, https://doi.org/10.3389/fmars.2019.00502, 2019.

Carter, B. R.: Empirical seawater property estimation routines, 2021.

Carter, B. R., Williams, N. L., Gray, A. R., and Feely, R. A.: Locally interpolated alkalinity regression for global alkalinity estimation, Limnology and Oceanography: Methods, 14, 268–277, https://doi.org/10.1002/lom3.10087, 2016.

Carter, B. R., Feely, R. A., Williams, N. L., Dickson, A. G., Fong, M. B., and Takeshita, Y.: Updated methods for global locally interpolated estimation of alkalinity, pH, and nitrate, Limnology and Oceanography: Methods, 16, 119–131, https://doi.org/10.1002/lom3.10232, 2017.

Carter, B. R., Bittig, H. C., Fassbender, A. J., Sharp, J. D., Takeshita, Y., Xu, Y., Álvarez, M., Wanninkhof, R., Feely, R. A., and Barbero, L.: New and updated global empirical seawater property estimation routines, Limnology & Ocean Methods, 19, 785–809, https://doi.org/10.1002/lom3.10461, 2021.

Dias, L. M. and Carter, B.: PyESPER: A Python version of Empirical Seawater Property Estimation Routines (ESPERs) (version 0), 2025.

Gammon, R. H., Cline, J., and Wisegarver, D.: Chlorofluoromethanes in the northeast Pacific Ocean: Measured vertical distributions and application as transient tracers of upper ocean mixing, Journal of Geophysical Research: Oceans, 87, 9441–9454, https://doi.org/10.1029/JC087iC12p09441, 1982.

Gruber, N., Clement, D., Carter, B. R., Feely, R. A., van Heuven, S., Hoppema, M., Ishii, M., Key, R. M., Kozyr, A., Lauvset, S. K., Monaco, C. L., Mathis, J. T., Murata, A., Olsen, A., Perez, F. F., Sabine, C. L., Tanhua, T., and Wanninkhof, R.: The oceanic sink for anthropogenic CO<sub>2 from 1994 to 2007, Science, 363, 1193–1199, https://doi.org/10.1126/science.aau5153, 2019.

Hauck, J., Hauck, J., Nissen, C., Landschützer, P., Rödenbeck, C., and Bushinsky, S.: Sparse observations induce large biases in estimates of the global ocean CO<sub>2 sink: An ocean model subsampling experiment, Philosophical Transactions of the Royal Society A, 381, 20220063, https://doi.org/10.1098/rsta.2022.0063, 2023.

Jiang, L.-Q., Dunne, J., Carter, B. R., Tjiputra, J. F., Terhaar, J., Sharp, J. D., Olsen, A., and Alin, S.: Global surface ocean acidification indicators from 1750 to 2100, Journal of Advances in Modeling Earth Systems, 15, 1–23, https://doi.org/10.1029/2022MS003563, 2020.

Jiang, L.-Q., Feely, R. A., Wanninkhof, R., Greeley, D., Barbero, L., Alin, S., Carter, B. R., Pierrot, D., Featherstone, C., Hooper, J., Melrose, C., Monacci, N., Sharp, J. D., Shellito, S., Xu, Y.-Y., Kozyr, A., Byrne, R. H., Cai, W.-J., Cross, J., Johnson, G. C., Hales, B., Langdon, C., Mathis, J., Salisbury, J., and Townsend, D. W.: Coastal Ocean Data Analysis Product in North America (CODAP-NA) – an internally consistent data product for discrete inorganic carbon, oxygen, and nutrients on the North American ocean margins, Earth Syst. Sci. Data, 13, 2777–2799, https://doi.org/10.5194/essd-13-2777-2021, 2021.

Keppler, L., Landschützer, P., Gruber, N., Lauvset, S. K., and Stemmler, I.: Seasonal carbon dynamics in the near-global ocean, Global Biogeochemical Cycles, 34, e2020GB006571, https://doi.org/10.1029/2020GB006571, 2020.

Lauvset, S. K., Key, R. M., Olsen, A., van Heuven, S., Velo, A., Lin, X., Schirnick, C., Kozyr, A., Tanhua, T., Hoppema, M., Jutterström, S., Steinfeldt, R., Jeansson, E., Ishii, M., Perez, F. F., Suzuki, T., and Watelet, S.: A new global interior ocean mapped climatology: the 1◦ × 1◦ GLODAP version 2, Earth System Science Data, 8, 325–340, https://doi.org/doi:10.5194/essd-8-325-2016, 2016.

Lauvset, S. K., Lange, N., Tanhua, T., Bittig, H. C., Olsen, A., Kozyr, A., Alin, S., Álvarez, M., Azetsu-Scott, K., Barbero, L., Becker, S., Brown, P. J., Carter, B. R., Da Cunha, L. C., Feely, R. A., Hoppema, M., Humphreys, M. P., Ishii, M., Jeansson, E., Jiang, L.-Q., Jones, S. D., Lo Monaco, C., Murata, A., Müller, J. D., Pérez, F. F., Pfeil, B., Schirnick, C., Steinfeldt, R., Suzuki, T., Tilbrook, B., Ulfsbo, A., Velo, A., Woosley, R. J., and Key, R. M.: GLODAPv2.2022: The latest version of the global interior ocean biogeochemical data product, Earth Syst. Sci. Data, 14, 5543–5572, https://doi.org/10.5194/essd-14-5543-2022, 2022.

Maurer, T. L., Plant, J. N., and Johnson, K. S.: Delayed-mode quality control of oxygen, nitrate and pH data on SOCCOM biogeochemical profiling floats, Frontiers in Marine Science, 8, https://doi.org/10.3389/fmars.2021.683207, 2021.

Olsen, A., Lange, N., Key, R. M., Tanhua, T., Bittig, H. C., Kozyr, A., Álvarez, M., Azetsu-Scott, K., Becker, S., Brown, P. J., Carter, B. R., Cotrim Da Cunha, L., Feely, R. A., Van Heuven, S., Hoppema, M., Ishii, M., Jeansson, E., Jutterström, S., Landa, C. S., Lauvset, S. K., Michaelis, P., Murata, A., Pérez, F. F., Pfeil, B., Schirnick, C., Steinfeldt, R., Suzuki, T., Tilbrook, B., Velo, A., Wanninkhof, R., and Woosley, R. J.: An updated version of the global interior ocean biogeochemical data product, GLODAPv2.2020, Earth Syst. Sci. Data, 12, 3653–3678, https://doi.org/10.5194/essd-12-3653-2020, 2020.

Roemmich, D. and Gilson, J.: The 2004-2008 mean and annual cycle of temperature, salinity, and steric height in the global ocean from the Argo Program, Progress in Oceanography, 82, 81–100, https://doi.org/10.1016/j.pocean.2009.03.004, 2009.

Sharp, J. D., Fassbender, A. J., Carter, B. R., Johnson, G. C., Schultz, C., and Dunne, J. P.: GOBAI-O2 : Temporally and spatially resolved fields of ocean interior dissolved oxygen over nearly 2 decades, Earth Syst. Sci. Data, 15, 4481–4518, https://doi.org/10.5194/essd-15-4481-2023, 2023.

Sharp, J. D., Jiang, L.-Q., Carter, B. R., Lavin, P. D., Yoo, H., and Cross, S. L.: A mapped dataset of surface ocean acidification indicators in large marine ecosystems of the United States, Sci Data, 11, 715, https://doi.org/10.1038/s41597-024-03530-7, 2024.

Tanhua, T., Körtzinger, A., Friis, K., Waugh, D. W., and Wallace, D. W. R.: An estimate of anthropogenic CO<sub>2 inventory from decadal changes in oceanic carbon content, Proceedings of the National Academy of Sciences of the United States of America, 104, 3037–3042, https://doi.org/10.1073/pnas.0606574104, 2007.

Tanhua, T., Lauvset, S. K., Lange, N., Olsen, A., Álvarez, M., Diggs, S., Bittig, H. C., Brown, P. J., Carter, B. R., da Cunha, L. C., Feely, R. A., Hoppema, M., Ishii, M., Jeansson, E., Kozyr, A., Murata, A., Pérez, F. F., Pfeil, B., Schirnick, C., Steinfeldt, R., Telszewski, M., Tilbrook, B., Velo, A., Wanninkhof, R., Burger, E., O'Brien, K., and Key, R. M.: A vision for FAIR ocean data products, Communications Earth and Environment, 2, 19–22, https://doi.org/10.1038/s43247-021-00209-4, 2021.

Velo, A., Pérez, F. F., Tanhua, T., Gilcoto, M., Ríos, A. F., and Key, R. M.: Total alkalinity estimation using MLR and neural network techniques, Journal of Marine Systems, 111–112, 11–18, https://doi.org/10.1016/j.jmarsys.2012.09.002, 2013.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., Van Der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., Van Mulbregt, P., SciPy 1.0 Contributors, Vijaykumar, A., Bardelli, A. P., Rothberg, A., Hilboll, A., Kloeckner, A., Scopatz, A., Lee, A., Rokem, A., Woods, C. N., Fulton, C., Masson, C., Häggström, C., Fitzgerald, C., Nicholson, D. A., Hagen, D. R., Pasechnik, D. V., Olivetti, E., Martin, E., Wieser, E., Silva, F., Lenders, F., Wilhelm, F., Young, G., Price, G. A., Ingold, G.-L., Allen, G. E., Lee, G. R., Audren, H., Probst, I., Dietrich, J. P., Silterra, J., Webber, J. T., Slavič, J., Nothman, J., Buchner, J., Kulick, J., Schönberger, J. L., De Miranda Cardoso, J. V., Reimer, J., Harrington, J., Rodríguez, J. L. C., Nunez-Iglesias, J., Kuczynski, J., Tritz, K., Thoma, M., Newville, M., Kümmerer, M., Bolingbroke, M., Tartre, M., Pak, M., Smith, N. J., Nowaczyk, N., Shebanov, N., Pavlyk, O., Brodtkorb, P. A., Lee, P., McGibbon, R. T., Feldbauer, R., Lewis, S., Tygier, S., Sievert, S., Vigna, S., Peterson, S., More, S., et al.: SciPy 1.0: fundamental algorithms for scientific computing in Python, Nat Methods, 17, 261–272, https://doi.org/10.1038/s41592-019-0686-2, 2020.

Waugh, D. W., Hall, T. M., Mcneil, B. I., Key, R., and Matear, R. J.: Anthropogenic CO<sub>2 in the oceans estimated using transit time distributions, Tellus, Series B: Chemical and Physical Meteorology, 58, 376–389, https://doi.org/10.1111/j.1600-0889.2006.00222.x, 2006.

Zeebe, R. E. and Wolf-Gladrow, D. A.: CO<sub>2 in Seawater: Equilibrium, Kinetics, Isotopes, Elsevier Science B.V., Amsterdam, The Netherlands, 2001.

Page 17: [1] Deleted	**Larissa Dias**	**5/28/25 10:23:00 AM**
Page 20: [2] Deleted	**Larissa Dias**	**6/13/25 12:31:00 PM**
Page 20: [2] Deleted	**Larissa Dias**	**6/13/25 12:31:00 PM**
Page 23: [3] Deleted	**Larissa Dias**	**5/28/25 10:30:00 AM**