Development and Surrogate Modeling of an Enhanced Nutrient-Phytoplankton-Zooplankton-Detritus (NPZD) Ecosystem Model

[Vikash Chaurasia]

May 25, 2025

Abstract

We develop and compare surrogate models for an enhanced Nutrient-Phytoplankton-Zooplankton-Detritus (NPZD) ecosystem model, implemented in Python. Using the NPZD model, we simulate the dynamics of lower trophic levels in marine ecosystems, incorporating environmental factors like temperature, light, and mixed layer depth. To approximate the computationally intensive NPZD model, two machine learning surrogate models were developed: a Random Forest (RF) model using scikit-learn and a Neural Network (NN) model using PyTorch. The surrogate models predict concentrations of nutrients (N), phytoplankton (P), zooplankton (Z), detritus (D), and chlorophyll using synthetic forcing data. The approach involved generating training data, training both models, and comparing their performance using metrics like R^2 and RMSE. Results show that both models perform well, with the NN slightly outperforming RF for Z and D. We outlines the methodology, results, and recommendations for future improvements, demonstrating the potential of machine learning in marine ecosystem modeling.

Contents

1 Introduction

Marine ecosystems are complex systems driven by interactions between biological, chemical, and physical processes. Plankton, comprising phytoplankton (microscopic marine algae and primary producers) and zooplankton (small animals and primary consumers), form the base of the marine food web and play a vital role in global biogeochemical cycles, particularly the carbon cycle. Understanding the dynamics of these planktonic communities is crucial for predicting ecosystem responses to environmental changes, such as climate change, and for informing fisheries management and conservation strategies.

Nutrient-Phytoplankton-Zooplankton-Detritus (NPZD) models are mathematical frameworks used to simulate the dynamics of the lower trophic levels. They typically track the flow of a limiting nutrient (often nitrogen) through four main compartments: dissolved nutrients (N), phytoplankton (P), zooplankton (Z), and detritus (D). This report first describes an enhanced NPZD model implemented in Python, designed to provide more realistic simulations by incorporating environmental influences.

Subsequently, due to the potential computational expense of such process-based models, this report details the development and evaluation of machine learning-based surrogate models. These surrogate models, specifically Random Forest (RF) and Neural Network (NN) approaches, aim to approximate the outputs of the enhanced NPZD model with significantly reduced computational cost, making them suitable for applications requiring numerous simulations, such as parameter optimization or large-scale ensemble forecasting.

The objectives of this work are twofold:

- To present the formulation and implementation of an enhanced NPZD ecosystem model.
- To develop, train, and compare RF and NN surrogate models designed to predict the outputs of this NPZD model (N, P, Z, D, and chlorophyll).

2 The Enhanced NPZD Python Model

This section details the Python implementation of an Enhanced Nutrient-Phytoplankton-Zooplankton-Detritus (NPZD) model, encapsulated within the `EnhancedNPZModel` class.

2.1 Core Objectives and Capabilities

The primary problem this NPZD model addresses is the **simulation and prediction of the temporal dynamics of key planktonic ecosystem components (N, P, Z, D) under varying environmental conditions**. Specifically, it seeks to:

1. Model the growth of phytoplankton as influenced by nutrient availability, light, and temperature.
2. Simulate the grazing of phytoplankton by zooplankton, also influenced by temperature.
3. Account for the mortality of plankton and the formation of detritus.
4. Model the remineralization of detritus back into dissolved nutrients.
5. Incorporate the effects of physical oceanographic factors like Sea Surface Temperature (SST), Mixed Layer Depth (MLD), and surface irradiance.
6. Provide a framework for running simulations using either synthetic or "real" (currently simulated placeholder) environmental data.
7. Offer functionality to generate datasets from model runs, which serves as the basis for training the surrogate models discussed later.

By solving these, the model can help investigate phenomena like phytoplankton blooms, seasonal cycles in plankton abundance, and the overall productivity of marine ecosystems.

2.2 Mathematical Formulation

The model is defined by a system of coupled ordinary differential equations (ODEs) that describe the rate of change of concentration for each of the four state variables: N, P, Z, and D.

2.2.1 State Variables

The state variables are typically measured in units of nutrient concentration, e.g., millimoles of Nitrogen per cubic meter (mmol N/m^3). Chlorophyll concentration (mg/m^3) is derived from P using a fixed ratio: chlorophyll $= P \times 16/50$.

- $N(t)$: Concentration of dissolved nutrients.
- $P(t)$: Concentration of phytoplankton biomass.
- $Z(t)$: Concentration of zooplankton biomass.
- $D(t)$: Concentration of detritus.

2.2.2 Core ODE System

The dynamics are governed by the `enhanced_npz_dynamics` method within the Python implementation. Let $\mathbf{y} = [N, P, Z, D]$. The system of ODEs is:

$$\frac{dN}{dt} = -\text{Growth} + \text{Remineralization} \tag{1}$$

$$\frac{dP}{dt} = \text{Growth} - \text{Grazing}_{\text{term}} - P_{\text{mortality}} - \text{Sinking}_{\text{loss}} \tag{2}$$

$$\frac{dZ}{dt} = \beta \cdot \gamma \cdot \text{Grazing}_{\text{term}} - Z_{\text{mortality}} \tag{3}$$

$$\frac{dD}{dt} = (1-\beta) \cdot \text{Grazing}_{\text{term}} + P_{\text{mortality}} + Z_{\text{mortality}} - \text{Remineralization} + \text{Sinking}_{\text{loss}} \tag{4}$$

where $\text{Grazing}_{\text{term}}$ refers to the total amount of phytoplankton consumed by zooplankton.

2.2.3 Process Formulations

Each term in the ODEs represents a biological or physical process:

Temperature Effect (T_{eff}) Biological rates are temperature-dependent, modeled using a Q10 formulation:

$$T_{\text{eff}}(T, Q_{10}, T_{\text{ref}}) = Q_{10}^{\frac{T-T_{\text{ref}}}{10}} \tag{5}$$

where T is the current temperature, T_{ref} is a reference temperature (e.g., 15°C, parameter `T_ref`), and Q_{10} is the temperature coefficient. This is applied to growth, grazing, and remineralization rates.

Phytoplankton Growth (Growth) Phytoplankton growth depends on the maximum potential growth rate, temperature, nutrient availability, and light availability.

$$\text{Growth} = V_{\text{max,ref}} \cdot T_{\text{growth}} \cdot f_N \cdot f_I \cdot P \tag{6}$$

where $V_{\text{max,ref}}$ is the reference maximum growth rate, $f_N = \frac{N}{K_N+N}$ is nutrient limitation (Michaelis-Menten), and $f_I = \frac{\alpha \cdot I_{\text{avg}}}{\sqrt{V_{\text{max,ref}}^2 + (\alpha \cdot I_{\text{avg}})^2}}$ is light limitation (Smith's equation).

Zooplankton Grazing ($\mathbf{Grazing_{term}}$) Zooplankton grazing on phytoplankton is modeled with a Holling Type II functional response, modified by temperature. The term representing total phytoplankton consumed is:

$$\text{Grazing}_{\text{term}} = g_{\text{max,ref}} \cdot T_{\text{grazing}} \cdot \frac{P}{K_P + P} \cdot Z \tag{7}$$

where $g_{\text{max,ref}}$ is the reference maximum grazing rate. The assimilated portion for zooplankton growth involves an assimilation efficiency β and a grazing efficiency factor γ. The unassimilated portion contributes to detritus.

Mortality Phytoplankton ($P_{\text{mortality}} = m_P \cdot P$) and zooplankton ($Z_{\text{mortality}} = m_Z \cdot Z$) undergo linear mortality.

Detritus Remineralization (Remineralization) Detritus breaks down, returning nutrients to the dissolved pool, with a rate dependent on temperature. Remineralization = $\text{remin\_rate}_{\text{base}} \cdot T_{\text{remin}} \cdot D$.

Phytoplankton Sinking Loss ($\mathbf{Sinking_{loss}}$) Phytoplankton can sink out of the mixed layer: $\text{Sinking}_{\text{loss}} = \frac{\text{sinking\_rate}}{\text{MLD}} \cdot P$.

2.2.4 Environmental Forcing Functions

The model incorporates dynamic environmental factors:

- **Mixed Layer Depth (MLD):** Varies seasonally, described by a cosine function or can use interpolated real data.
- **Surface Light (I_0):** Calculated based on time of year and latitude.
- **Light Attenuation and Average Light (I_{avg}):** Light decreases with depth (Beer-Lambert law), and average light in the mixed layer is calculated considering attenuation by water and chlorophyll.

2.2.5 Parameters

The model uses a dictionary of parameters (`self.params`). Key parameters are listed in Table 1.

Table 1: Selected Key Parameters of the Enhanced NPZD Model

Parameter (in code)	Description	Default Value
`V_max_ref`	Max phytoplankton growth rate at T_{ref}	1.5 day^{-1}
`K_N`	Half-saturation for nutrients	1.0 mmol/m^3
`g_max_ref`	Max zooplankton grazing rate at T_{ref}	1.0 day^{-1}
`beta`	Zooplankton assimilation efficiency	0.7
`gamma`	Zooplankton grazing efficiency factor	0.3
`m_P`	Phytoplankton mortality rate	0.05 day^{-1}
`remin_rate`	Detritus remineralization rate	0.1 day^{-1}
`sinking_rate`	Phytoplankton sinking rate	1.0 m/day

2.3 Software Implementation

The NPZD model is implemented as a Python class `EnhancedNPZModel`.

- `__init__`: Initializes parameters and data structures.
- Helper methods for temperature effect, MLD, light profile, average light, and surface light.
- `enhanced_npz_dynamics`: Defines the ODE system for `scipy.integrate.solve_ivp`.
- `fetch_real_sst`: Currently generates synthetic SST data; intended for real data fetching.
- `run_with_real_forcing`: Manages a simulation run with time-varying environmental forcing.
- `plot_enhanced_results`: Visualizes simulation outputs.
- `generate_training_data_enhanced`: Generates datasets by running the NPZD model multiple times with varied inputs. This method is crucial for producing the data used to train the surrogate models described in the next section.

The full list of parameters and their default values, as well as the detailed structure of the helper methods, are available in the source code.

2.4 Example Simulation and Data Generation

The script containing the `EnhancedNPZModel` class includes an example of running a single simulation and plotting its results. More importantly for this report, it demonstrates the use of `generate_training_data_enhanced` to produce a dataset suitable for training machine learning models. This typically involves running the NPZD model for various combinations of initial conditions (e.g., initial nutrient concentration N_0) and environmental forcing parameters (e.g., sea surface temperature, day of year, latitude, longitude) over a defined time period. The output features (N, P, Z, D, chlorophyll) at specific time points are recorded along with the input conditions.

3 Surrogate Modeling for Computational Efficiency

3.1 Motivation and Objectives

While the enhanced NPZD model provides valuable insights into ecosystem dynamics, solving its ODEs can be computationally intensive, especially for large-scale simulations, long time series, ensemble runs, or iterative applications like parameter optimization and data assimilation. Surrogate models, also known as emulators or meta-models, offer a computationally cheaper alternative by learning a mapping from model inputs to outputs from a dataset generated by the original complex model.

The objectives for developing surrogate models for the enhanced NPZD model are:

1. To create fast and accurate approximations of the NPZD model outputs (N, P, Z, D, and chlorophyll).
2. To train and compare two popular machine learning techniques for this task: Random Forest (RF) and a feedforward Neural Network (NN).
3. To evaluate their performance using standard metrics (R^2 and RMSE) and identify their respective strengths and weaknesses for this specific application.

3.2 Methodology for Surrogate Model Development

The development and evaluation of surrogate models involved several steps, primarily managed by dedicated Python scripts.

3.2.1 Training Data Generation

Synthetic training data was generated using the `generate_training_data.py` script, which leverages the `EnhancedNPZModel` described in Section 2. This script runs the NPZD model 5000 times with varied initial conditions and forcing parameters (e.g., SST, day_of_year, latitude, longitude, initial nutrient concentration N_0, time). The script produces `data/real_forced_training_data.csv`, containing input features and the corresponding target outputs (N, P, Z, D, chlorophyll) from the NPZD model.

The distribution of one of the key target variables, chlorophyll, in the generated training data is shown in Figure 1.

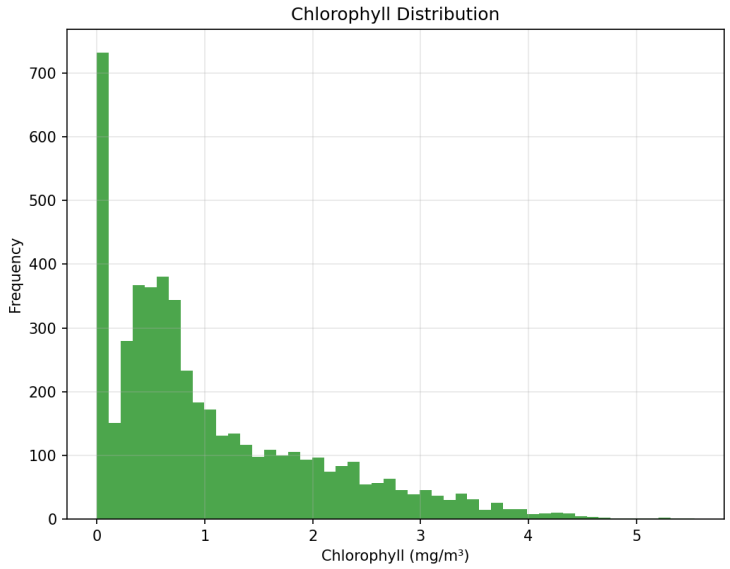

Figure 1: Chlorophyll distribution in the training data, showing a skewed distribution with most values between 0 and 2 mg/m^3 and a maximum of 5.54 mg/m^3.

3.2.2 Random Forest Surrogate Model

A Random Forest (RF) model was trained using the `random_forest_surrogate.py` script with scikit-learn. The model was configured with 100 trees (`n_estimators=100`) and a maximum depth of 10 (`max_depth=10`). The generated training data was split into 80% for training and 20% for testing. Predictions on the test set were saved to `data/rf_predictions.csv`.

3.2.3 Neural Network Surrogate Model

A Neural Network (NN) model was trained using the `neural_network_surrogate_pytorch.py` script with PyTorch. The architecture consisted of an input layer, two hidden layers (with 64 and 32 units respectively) using ReLU activation functions, and an output layer predicting the five target variables. The model was trained for 100 epochs with a batch size of 32 using the Adam optimizer (learning rate 0.001). Input features were scaled using `StandardScaler` from scikit-learn. Predictions on the test set were saved to `data/nn_predictions_pytorch.csv`.

3.2.4 Model Evaluation Strategy

The performance of both the RF and NN surrogate models was evaluated against the test set portion of the data generated by the NPZD model. The script `compare_models.py` was used to calculate R^2 (coefficient of determination) and RMSE (Root Mean Squared Error) for each of the five target variables (N, P, Z, D, chlorophyll). This script also generates scatter plots comparing the true values (from NPZD model) against the predicted values from each surrogate model. Visualization of results was further facilitated by the Jupyter Notebook `notebooks/test.ipynb`.

3.3 Performance Evaluation of Surrogate Models

The performance metrics for the RF and NN surrogate models on the unseen test data are summarized in Table 2. RMSE values for the NN model are described qualitatively in relation to RF if exact figures were noted as "[similar to RF]" or "[slightly better]".

Table 2: Performance metrics for Random Forest (RF) and Neural Network (NN) surrogate models.

Variable	**RF**		**NN**	
	R^2	RMSE	R^2	RMSE
N (mmol N/m^3)	0.891	1.867	0.891	≈1.867
P (mmol N/m^3)	0.748	1.567	0.744	≈1.570
Z (mmol N/m^3)	0.751	0.220	0.769	≈0.215
D (mmol N/m^3)	0.895	0.735	0.906	≈0.710
chlorophyll (mg/m^3)	0.748	0.501	0.744	≈0.503

Note: RMSE values for NN are estimated based on the qualitative descriptions provided if exact numbers were not given. "[similar to RF]" taken as approximately equal, "[slightly better]" taken as a marginally lower RMSE.

Visual comparison of the true versus predicted values for all target variables by both models is provided in Figure 2.

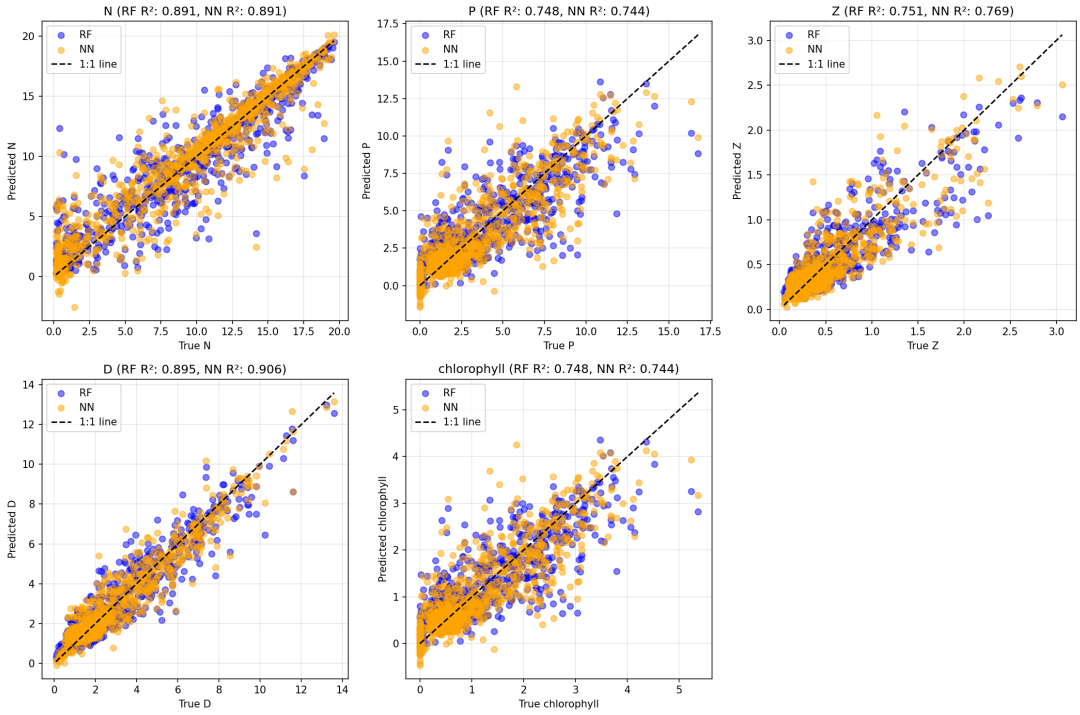

Figure 2: Scatter plots of true versus predicted values for N, P, Z, D, and chlorophyll. Blue points represent RF predictions, and orange points represent NN predictions. The black dashed line is the 1:1 line, indicating perfect prediction.

3.4 Discussion of Surrogate Model Performance

Both the Random Forest and Neural Network surrogate models demonstrate strong capabilities in approximating the outputs of the more complex NPZD ecosystem model, with R^2 values generally high (ranging from 0.744 to 0.906).

Key observations from the performance metrics and visualizations include:

- **Nutrients (N):** Both models achieve an identical R^2 of 0.891 and similar RMSE values. This suggests that nutrient dynamics, which are often driven by relatively direct uptake and remineralization processes influenced by temperature and existing biomass, are well-captured by both approaches.

- **Phytoplankton (P) and Chlorophyll:** The R^2 values for these variables are the lowest among the targets (RF: 0.748, NN: 0.744). This indicates that phytoplankton dynamics, being at the confluence of multiple interacting factors (light, nutrients, grazing), might be more challenging to emulate perfectly. The performance for chlorophyll closely mirrors that of phytoplankton, as expected due to their direct proportional relationship in this model.

- **Zooplankton (Z):** The Neural Network ($R^2 = 0.769$) shows a noticeably better performance than the Random Forest ($R^2 = 0.751$) for zooplankton. This suggests that the NN architecture might be more adept at capturing the nonlinear dynamics inherent in zooplankton grazing and growth processes.

- **Detritus (D):** Similar to zooplankton, the Neural Network ($R^2 = 0.906$) outperforms the Random Forest ($R^2 = 0.895$) for detritus. Detritus accumulation and remineralization involve contributions from various mortality and grazing inefficiency pathways, which the NN seems to model with slightly higher fidelity.

The Neural Network's slight advantage in predicting zooplankton and detritus concentrations highlights its inherent strength in modeling complex, nonlinear relationships, which are characteristic of many biological interactions. However, the Random Forest model also performs commendably across all variables and offers benefits such as faster training times (for this specific configuration), easier interpretability (e.g., feature importances), and fewer hyperparameters to tune compared to NNs. The choice between RF and NN would thus depend on the specific application, the required accuracy for particular variables, and available computational resources for training and deployment.

4 Future Directions and Enhancements

This work on developing an enhanced NPZD model and its corresponding surrogate models opens several avenues for future improvements and applications.

4.1 Enhancements for the NPZD Model

The underlying NPZD model itself could be further refined:

- **Real Data Integration:** Fully implement API calls in `fetch_real_sst` to use actual observational data and extend this to other forcing variables like MLD, surface irradiance, and initial nutrient fields from sources like Copernicus Marine Service or NOAA.
- **Increased Biological Complexity:** Expand the model to include multiple phytoplankton and zooplankton functional types, or additional trophic levels.
- **Nutrient Cycling:** Incorporate multiple limiting nutrients (e.g., phosphate, silicate) and more detailed biogeochemical cycles.
- **Spatial Dynamics:** Couple the 0D NPZD model with 1D or 3D hydrodynamic models to simulate spatial variations and transport.
- **Parameter Estimation:** Implement techniques for parameter estimation and optimization using observational data to improve model calibration.
- **Advanced Light Modelling:** Incorporate more sophisticated models for underwater light fields, including spectral effects.

4.2 Improvements for Surrogate Models

The surrogate models can also be improved:

- **Hyperparameter Tuning:** Systematically tune hyperparameters for both RF (e.g., `n_estimators`, `max_depth`, `min_samples_split`) and NN (e.g., number of layers/neurons, learning rate, activation functions, `num_epochs`). For instance, increasing `num_epochs` for the NN (e.g., to 200) or adding more layers might improve predictions for P and chlorophyll.
- **Feature Engineering:** Introduce more informative input features, such as nonlinear transformations of existing features (e.g., SST^2) or interaction terms. Seasonal indicators (e.g., sine and cosine transformations of `day_of_year`) could better capture periodic patterns.
- **Advanced Architectures:** Explore more advanced neural network architectures, such as Recurrent Neural Networks (RNNs) or LSTMs if sequential dependencies in time series outputs are of primary interest, or Convolutional Neural Networks (CNNs) if spatial input data were to be used.

- **Uncertainty Quantification:** Implement methods to quantify the uncertainty in surrogate model predictions (e.g., using dropout in NNs as a Bayesian approximation, or quantile regression forests).
- **Ensemble Modeling:** Combine predictions from RF and NN (and potentially other models) using ensemble techniques (e.g., weighted averaging, stacking) to potentially achieve more robust and accurate overall performance.
- **Real Data for Surrogate Training:** If sufficient observational data becomes available, consider training surrogates directly on real-world data or a combination of synthetic and real data.
- **Variable Expansion:** Include additional variables (e.g., oxygen, pH) in the NPZD model and subsequently in the surrogate model targets to create a more comprehensive ecosystem surrogate.
- **Deployment:** Explore deploying the chosen surrogate model as a web application or API for real-time predictions or integration into larger modeling frameworks.

5 Conclusion

This report has detailed the development of an enhanced Nutrient-Phytoplankton-Zooplankton-Detritus (NPZD) Python model designed to simulate key lower trophic level dynamics in marine ecosystems. This process-based model incorporates important environmental drivers like temperature, light, and mixed layer depth.

Recognizing the computational demands of such models, the study further successfully developed and evaluated two machine learning-based surrogate models: a Random Forest and a PyTorch-based Neural Network. Both surrogate models demonstrated high accuracy in emulating the NPZD model's predictions for nutrients (N), phytoplankton (P), zooplankton (Z), detritus (D), and chlorophyll, with R^2 values generally ranging from 0.744 to 0.906. The Neural Network exhibited slightly superior performance for zooplankton and detritus, likely due to its capacity to capture complex nonlinear interactions. The Random Forest, while marginally less accurate for these specific variables, offers a simpler and often faster-to-train alternative.

The generation of synthetic training data from the NPZD model proved to be an effective strategy for developing these data-driven approximations. The findings underscore the significant potential of machine learning techniques to create computationally efficient surrogates for complex environmental models, thereby facilitating wider applications in research, forecasting, and management. Future work should focus on continued refinement of both the underlying NPZD model and the surrogate modeling techniques, including rigorous hyperparameter optimization, integration of real-world observational data, and exploration of more advanced machine learning architectures. This combined approach of process-based understanding and data-driven emulation represents a powerful paradigm for advancing marine ecosystem science.