A hierarchical ensemble manifold methodology for new knowledge on spatial data: An application to ocean physics.

Maike Sonnewald[1,2,3*]

[1]**University of California Davis, Davis, California, USA**

[2]**University of Washington, Seattle, Washington, USA**

[3]**NOAA/Geophysical Fluid Dynamics Laboratory, Princeton, New Jersey, USA**

[*]**Corresponding author. Contact email: sonnewald@ucdavis.edu**

ABSTRACT

Extracting meaningful patterns from large, complex, and nonlinear earth science data remains a major challenge. Many traditional methods, such as Principal Component Analysis and k-means clustering, impose strong statistical assumptions that often fail in these settings, leading to misleading results. I introduce the Native Emergent Manifold Interrogation (NEMI) method, a novel workflow that integrates manifold learning, dynamical systems, and ensemble clustering to reveal meaningful structures in noisy, high-dimensional data. NEMI constructs a manifold to enhance underlying associations and applies unsupervised clustering to identify coherent regions of interest.

A key strength of NEMI is its intuitive validation framework, which enables practitioners to assess model reliability through visual inspection, ensemble agreement, and domain-specific expectations. By leveraging stochastic regularization, conceptualized as a smoothing of the space explored by the machine learning optimization, and uncertainty quantification, NEMI ensures that detected structures are robust and not artifacts of methodological choices. Furthermore, the method is flexible and scalable, allowing adaptation to different spatial scales, whether for identifying global dynamical regimes or resolving localized patterns within a specific region.

Demonstrated on oceanographic data, NEMI provides a generalizable, interpretable, and computationally efficient approach for data-driven discovery in the earth sciences. By balancing mathematical rigor with practical usability, NEMI offers a powerful tool for exploring complex geophysical datasets while ensuring results are transparent, reproducible, and tailored to the problem at hand.

Keywords: Data mining, Unsupervised Learning, Model validation

Significance Statement

Within the earth sciences, data is growing unmanageably large and challenging to extract insight from. Most commonly used methods employ highly restrictive assumptions regarding the underlying statistics of the data and may even offer misleading results unless sufficiently validated. To enable and accelerate scientific discovery, I drew on tools from computer science, statistics, and dynamical systems theory to develop the Native Emergent Manifold Interrogation (NEMI) method. NEMI is intended for wide use within the earth sciences and applied to an oceanographic example here. Using domain-specific theory, manifold representation of the data, clustering, and an ensemble strategy, NEMI is able to highlight particularly interesting areas within the data. I stress the underlying philosophy and appreciation of methods to facilitate using unsupervised ML as a tool to gain new knowledge.

1 INTRODUCTION AND PROBLEM STATEMENT

In this paper, I introduce a generic methodology for identifying patterns, or clusters, within a dataset that may have arbitrarily complex and non-linear covariance structures. The method is called Native Emergent Manifold Interrogation (NEMI). Described throughout this paper, and outlined in section 1.1, NEMI is a machine learning (ML) 'clustering workflow', that addresses key challenges in analyzing complex geophysical data by integrating manifold learning, clustering, stochastic regularization, uncertainty quantification, and intuitive validation, which enables robust and interpretable cluster identification. NEMI is agnostic to the application domain, with current applications including ocean physics, sea-ice, ocean biogeochemistry, and atmospheric physics. Tailored to non-expert users, NEMI is designed to 1) allow easy validation through accessible visual and domain-specific assessment of the ML model, 2) not make strong statistical assumptions, such as an underlying normal distribution necessary to the success of k-means, Gaussian Mixture Models and Principal Component Analysis, 3) have minimal parameters that require statistical insight to tune, and 4) an intuitively hierarchical usage that allows both 'global' and 'local' applications.

There is intense interest in ML, and many reviews posit ML methods could lead to paradigm-

shifting insights. In some important ways, discussed below, ML application is different in philosophy and base assumptions to the methods that are conventionally applied in the Earth Sciences. Challenges posed by modern data volumes and inherent non-normal and non-linear behaviors, mean that some traditional as well as novel ML methods of analysis can be inadequate. The rate of adoption of ML should be matched by a growth in the intuition needed to validate and verify results. Indeed, many types of data available do not yet have off-the-shelf ML methods that are appropriate for them. A core motivation behind NEMI is thorough and intuitive validation, where the NEMI workflow is designed to make how well the model performs visually apparent.

To illustrate NEMI, I use both synthetic data and a more complex application to ocean physics. In the NEMI workflow, and this manuscript, **I address the reader who is not deeply familiar with unsupervised ML and statistics**, and various sections are designed to allow readers to focus on the aspect that most appeals to them. I intend to give a thorough explanation of the reasoning behind NEMI, and aim to empower practitioners with the rationale behind different method choices so it can be applied to different data. Aspects of NEMI span a wide array of fields, and only brief explanations are within the present scope to keep the paper informative and within a reasonable page limit. Beyond introducing NEMI, the paper is intended as starting points for further reading. This paper is a companion to the code-base which, for ease of use, has undergone extensive development (github.com/CompClimate/NEMI).

1.1 The NEMI algorithm: Method overview

NEMI is based on an inner manifold learning loop that is repeated as needed and determines one set of clusters per iteration, or ensemble member (Fig. 1 and Fig. 2.1 for the ocean physics example), and an outer ensemble assessment loop (Fig. 1 and Fig. 2.2) that takes the individual clusters and determines overlap and uncertainty used to tune parameters to arrive at the final clusters that are determined across the ensemble. The outer loop is used to fine-tune parameters in the inner loop, as this is where the uncertainty is quantified. Data should be preprocessed as appropriate. Table 1 shows a short summary and a longer-form pseudocode, following the example in Fig. 2, is as follows:

Inner manifold learning loop (Fig. 2.1, section 3.1):

I. Create graph and embedding, default is UMAP:

- Validate embedding, initially from visual or numeric assessment.

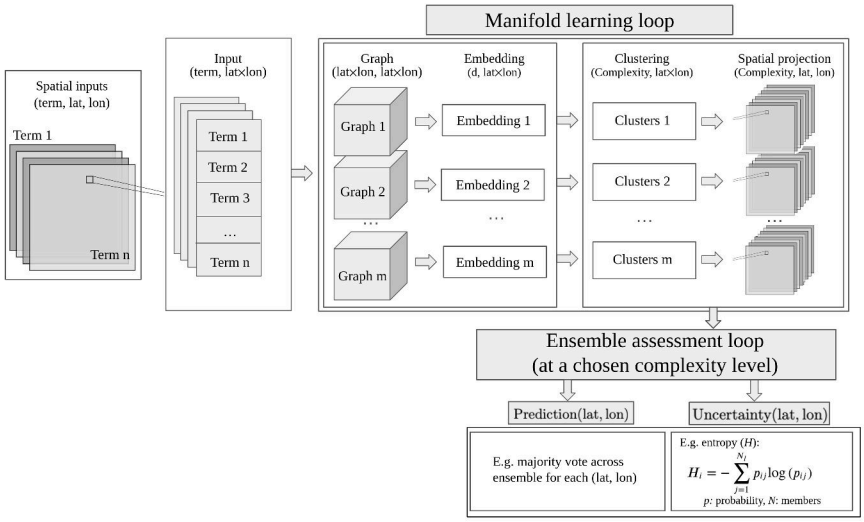

Figure 1. Sketch of abstract workflow in NEMI. Spatial input fields are first transformed into input vectors. An inner manifold-learning loop constructs an m-member ensemble by building graphs, learning embeddings, clustering, and projecting clusters back into physical space. These projections enable visual and quantitative assessment of clustering choices and parameter tuning for ensemble members. In an outer ensemble-assessment loop, predictions and associated uncertainty are computed across the ensemble at a selected complexity level. Validation occurs at multiple stages of the workflow (see Section 1.1), and the resulting uncertainty informs the adequacy of parameter choices for the target application. Figure in collaboration with Makayla McDevitt and Dr. Laique Djeutchouang.

II. Cluster on the embedding:

- Choose clustering method, with the default being agglomerative clustering.
- Validate clustering, visually assessing if clustering has partitioned embedded space well.

Outer ensemble assessment loop (Fig. 2.2, section 3.2):

I. Across ensemble from inner loop validation:

- Set the desired number of clusters if using agglomerative algorithm.
- Determine overlap between the different clusters across the ensemble.

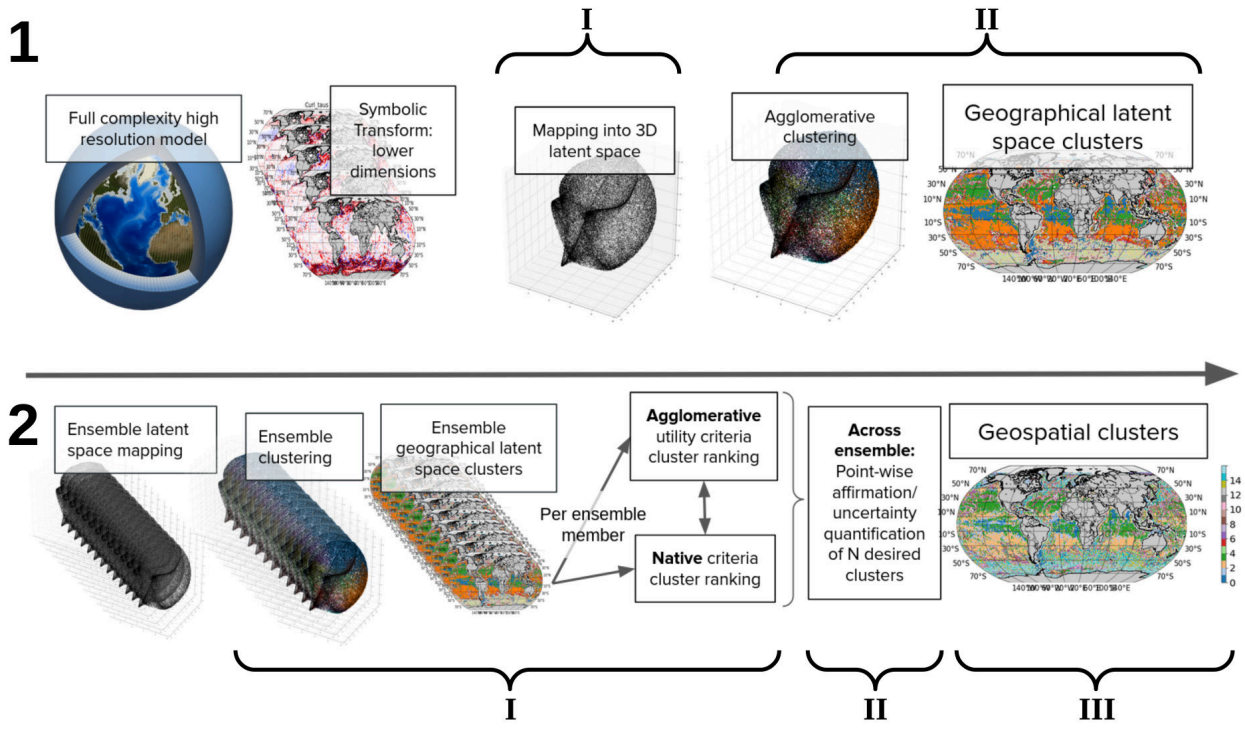

Figure 2. Sketch NEMI application to ocean physics example. Structure follows description and roman numerals in section 1.1. Part 1 (top row) in illustrates moving from the data in its raw form, through initial symbolic rendition, manifold transformation displaying the latent space, or the hidden structures and relationships within the data, and clustering. Part 2 (bottom row) shows the ensembling, agglomerative utility ranking, and native (field-specific) utility ranking within each ensemble member. Finally, the cluster for each location is determined by looking across the ensemble. (Top left image of model adapted from encyclopedie-environnement.org).

- Assign cluster number via a majority vote, or similar statistics.

II. Assess the uncertainty of cluster assignment via entropy, or similar statistics.

III. External validation: Assess against domain-specific expectation.

Repeat until an acceptable uncertainty is reached across clusters of interest.

1.2 The role of methodological progress for gaining insight

It is worth understanding why ML methods can have such profound impact. Scientific progress can be seen as paced (Kaiser et al., 2022) by our ability to discern and understand patterns in the system of interest. The scientific method is rooted in formulating testable hypotheses, effectively

finding patterns in the system of interest, and constructing frameworks to explain and ultimately predict features of interest (Kaiser et al., 2022). From geology to cloud microphysics we make progress by collecting theories, described qualitatively or quantitatively. ML, and unsupervised ML in particular, is being harnessed to determine patterns in data.

ML offers an avenue towards more objective identification of patterns and subsequent formulation of theories. To illustrate why this is beneficial, note that the history of science is populated by theories that today seem outlandish. For example, Leibniz, who invented the differential and integral calculus, also stated in *Protogaea* the existence of a four-footed unicorn the size of a horse as fact complete with an illustration of a reconstructed skeleton. The missing components were blamed on the 'ignorant' workers who had excavated the bones. Today, scientific consensus classifies unicorns as fictional and 'Leibniz's Unicorn' as an early example of the paleontology of a woolly rhinoceros. Leibniz, unquestionably a genius, operated within the limitations of available data and prevailing assumptions, drawing conclusions that were reasonable at the time. Preconceived notions can impact entire fields of study. For example, in fluid dynamics, the 'hydrodynamic' paradox is a contradiction determined by d'Alembert in 1752: for inviscid and incompressible potential flow the drag force is zero on a body moving with constant velocity relative to the fluid. Fluid mechanics as a field was discredited by engineers thereafter, as such zero drag contradicts the observation of substantial drag. Theoretical fluid mechanics *were not used by engineers* until a discovery by Prandtl in 1904 of a thin boundary layer that remained as a result of viscous forces. The thin boundary layer had been present in every relevant experiment, and preconceived notions distracted an entire field for 152 years from recognizing its importance. The results of this reconciliation include the development of air foils that allow modern air travel. The examples of Leibniz's unicorn and d'Alembert's paradox serve as allegories to more contemporary exploration of data, and beg the question: What breakthroughs can we discover if we explore data without preconceived notions? In unsupervised ML applications, outdated or overly rigid methods risk misinterpreting or oversimplifying complex datasets we have available today. NEMI enables scientists to navigate beyond the 'myths' of linearity and normality in traditional methods, as discussed below, to uncover nuanced aspects hidden in modern data.

Approaches that are more 'objective', meaning less rooted in preconceived notions, hold great promise. The main strength and promise of ML is increased objectivity. Precursors to modern ML methods, such as regression and principal component analysis were popularized by Lorenz (1956). While highlighting their limitations, and offering NEMI as a natural progression with

numerous benefits is the main message of this manuscript, it is worth appreciating the astounding insight even methods from the 1920s and earlier, such as PCA and linear regression have achieved. PCA, for example, was popularized in meteorology as a method of dimensionality reduction of large geospatial datasets. In 1928, during the British occupation of India, Walker Walker (1928) was tasked with discovering the cause of the interannual fluctuation of the Indian monsoon. A failure in the monsoon meant widespread drought in India, and in colonial times also famine Davis (2001). To find possible correlations, Walker used a large number of Indian clerks, or "computers", to carry out calculations by hand across all available data. This labor, enabled by colonial administrative structures, yielded the discovery of the Southern Oscillation, the seesaw in the West-East temperature gradient in the Pacific, which we know now by its modern name, El Niño Southern Oscillation (ENSO). Beyond observed correlations, theories of ENSO and its emergence from coupled atmosphere-ocean dynamics appeared decades later Zebiak and Cane (1987).

Since Walker's seminal work, progress has been made in the development of methods and the amount and quality of data available. As a scientific community, we now need methods that do justice to the data we have. To illustrate why, consider the still widely popular PCA, linear regression, and even methods such as 'k-means' (from Hugo Steinhaus in 1956). These methods are statistical tools that we use to make a simplified version of the data. However, by choosing a tool, we are also including its underlying assumptions, which, for example, for PCA as it is widely used includes that we have an underlying normal distribution. Considering the underlying assumptions, and choosing appropriate ones, is what modern ML tools allow us to do. A hammer, while designed for a nail, would if wielded with determination be able to get a screw into a wall, but using a screwdriver would do the job more efficiently as it can account for the curved rills of the screw.

How do we know if our dataset is a nail or a screw? A 'strict' assumption includes assuming a system is linear or has a normal distribution. These are fair starting points, and a more complicated method should only be used if proven necessary. For example, even if changes due to a warming climate are suspected, a linear model should initially be tested, and a normal distribution should still be a first guess. However, intuition makes it clear that, for example, a dataset tracking the global mean surface temperature, such as the "hockey stick" figure from Mann et al. (1999) stretching back over a thousand years, is not captured well using a linear model, that is, a straight line. This is because there is a sharp increase towards modern times, collectively known as global warming. In many areas of the earth system, global warming is changing the underlying statistical properties of

data. Simple statistical tests, as described below, can readily demonstrate if a statistical assumptions are appropriate.

1.3 NEMI in the context of modern data analysis

NEMI was developed to address the need to identify meaningful patterns in the increasingly large, highly complex, and complicated data that are becoming common within the earth sciences and beyond. Here, the term complex is used to mean having emergent properties based on underlying rules. However, with more complex methods and data come novel challenges. In terms of assisting practitioners navigate these novel challenges, NEMI is designed to address the issues of 1) validation, and 2) choice of clustering algorithm for high-dimensional and nonlinear data. Validation, which is discussed further below, is important because it helps determine if a useful statistical model of the dataset has been found. An ML model is only useful if it determines a similar statistical model every time it is run, which is not a given using ML. By leveraging stochastic regularization, which can be conceptualized as a smoothing of the space explored by the ML optimization, paired with uncertainty quantification, NEMI ensures that detected structures are robust and not artifacts of methodological choices. The choice of clustering method, also discussed further below, targets the issue of needing expertise and a comprehensive understanding of underlying assumptions within available methods. Note that the present manuscript focuses on NEMI, and I do not include a general overview of ML within the earth sciences of which there are many, including Fleming et al. (2021); Sonnewald et al. (2021); Beucler et al. (2021); Sun et al. (2024); Dramsch (2020); Bracco et al. (2024); Lai et al. (2025).

NEMI blends dynamical systems theory with clustering, through utilizing manifolds, but importantly invites room at key areas for domain specific input native, or field specific, to the research problem the workflow is applied to. With NEMI, I address the issue of mismatching data science methods and data, where the practitioners of earth science or more pure computational science, including ML specialists, often face the difficulty of interdisciplinary communication. NEMI is a generalization of the methodology in Sonnewald et al. (2020) that targeted plankton ecosystems, in that it is designed to be agnostic to the earth science sub-domain and scale to larger datasets.

Scaling is one of the key bottlenecks in unsupervised ML for scientific applications. NEMI is generalized to work with any data, where the particular example application used here is geospatial

data. In the presented example, I use an explicitly hierarchical approach, resulting in a less parametric methodology (fewer parameters to tune and less danger of noise interference), which, as demonstrated in the example application below, is intuitively useful both for global (such as, the whole Earth) and more local applications (such as, a basin or more regional assessment). Another novelty in NEMI is the lack of a fixed field-specific benchmark criteria (used in Sonnewald et al. (2020)), where I have generalized so a field agnostic option is available. Lastly, NEMI invites the use of a range of uncertainty quantification options in the final cluster evaluation. The intended readership of this manuscript is the interested practitioner from the earth sciences, meaning scientists wishing to apply NEMI, together with an interest in understanding the underlying philosophy and rationale beneath the architecture of the pipeline. I have attempted to describe concepts in detail and refer the interested reader to further materials, especially to do with mathematical derivations. Oceanographic concepts and data from an ocean numerical model is used as an example, and not explained in detail.

The rest of the paper is structured as follows. To give NEMI context, I initially move through explaining the problems related to exploring data using ML methodologies (section 2), first using a synthetic example to illustrate overall principles (section 2.3), and then using data from a realistic ocean model (section 2.4) highlighting the principles motivating NEMI. In section 3, I move through the different components of NEMI using the ocean model data introduced in section 2.4. The sections follow the structure given in the outline within section 1.1 below. Section 3.2 describes the outer, and section 3.1 the inner workflow, with roman numerals consistent between sections and the overview in section 1.1 and Fig. 2. Finally, section 4 provides an outlook on potential applications and implications. I refer to NEMI as the full workflow, but separate parts can be used and adapted as appropriate for the practitioner. The following provided code and examples use the python programming language and key parameters are highlighted. Note that parameters not discussed could be significant depending on the application, and the documentation should be consulted. NEMI is a framework and codebase that is as per writing being actively developed, and I welcome contributions from the community. The code for NEMI is available on GitHub and also as a PyPi package: https://github.com/compclimate/NEMI.

2 KEY CONCEPTS FROM MACHINE LEARNING APPLICATIONS

This section is intended to give a brief overview of the key underlying concepts that are used within unsupervised ML, with illustrations first based on synthetic data, and then on data from a realistic ocean model (section 2.4) from MOM6 (Griffies et al., 2024a,b)). The structure is intended to allow unfamiliar practitioners to have enough background to follow the context and challenges addressed with NEMI. More experienced practitioners can move directly to the section that deals with NEMI applied to the ocean model data (section 3). Introducing the NEMI framework is the main purpose of this manuscript, and throughout section 2 I refer back to how the concepts are folded into NEMI. Assuming that readers may not read the manuscript sequentially, some repetition is present.

2.1 Methods from unsupervised learning

Novel methods from ML are increasingly being used to great advantage. In Sonnewald et al. (2021) a review of current progress and a brief introduction of methods can be found focused on physical oceanography. However, matching methods to data and robustly verifying their results requires knowledge both of the algorithm and the application. A computer scientist may believe she has arrived at a significant and interesting answer, but this may not be useful to an earth scientist if, for example, the uncertainty related to the spatial position of an identified region is too great, or something trivial is revealed, such as the seasons, due to poor preprocessing.

Along with an appreciation of a method's benefits should come an appropriate level of skepticism and emphasis on validation, statistical or otherwise. However, often methods available for validation are also not appropriate for the needs of a practitioner. NEMI addresses the issue of not having satisfying metrics to use to determine the statistical significance of clustering results. Clustering is the task of dividing data into sub-groups so that data points within each group are similar to each other, and dissimilar to the data points in other groups. Clustering is largely regarded to be an 'unsupervised' ML methodology, meaning that the data are given to the method without explicit 'labels'. As such, clustering can be seen as the act of determining labels that can then be interpreted and offer insight to the practitioner through subsequent analysis.

A large and growing number of clustering algorithms are available. It is beyond the scope of this article to give an overview of these. The act of seeking to determine sub-groups within the data, which is common to all clustering methods, requires that one define how differences between the overall data and any potential sub-groups should be quantified. Note that such partitioning of

the covariance space of the data are also the backbone of, for example, neural networks. When applying a clustering algorithm, the resultant 'model' is the algorithm and the chosen values for any parameters or similar. As such, it is critical to quantify how well the clustering model is able to represent the data. This is also seen in simple regression, which offers an intuitive allegory. Within clustering and regression, we search for an underlying and general model, or formula, to describe the data. To illustrate, Fig. 3 has an example where the underlying model in the top row is a sinusoid curve with a linear trend (Fig. 3a-c), and in the bottom row there is a dataset of red and green dots (Fig. 3d-f). The left column (Fig. 3a and d) shows an underfitted model, attempting to partition the data with a straight line. This is not a good fit for large numbers of data points. The rightmost column (Fig. 3c and f) illustrates a model fit where every point is accounted for, which also fails to reasonably approximate the underlying model. The middle column illustrates a general fit that more closely represents a model that closely approximates the underlying model from which the data was drawn (Fig. 3b and e). In using the term model we assume that the underlying system has some non-random relationships that can be statistically approximated.

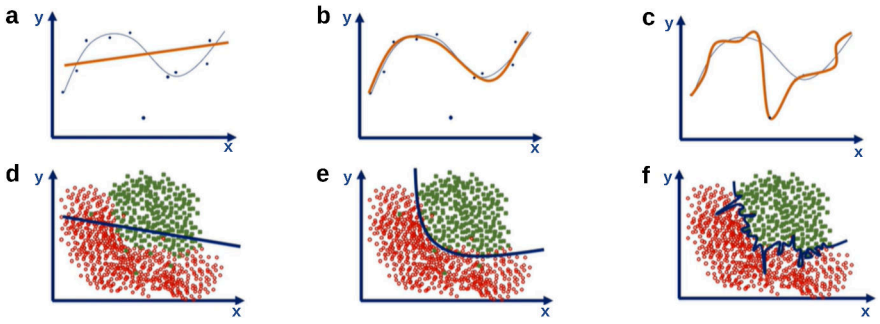

Figure 3. Illustration of a model fitting exercise. Top row (a-c) uses a regression example where the true model (gray line) is a sine curve with a linear trend and the data are the scattered points. The bottom row (d-f) shows a clustering example. An underfitted, good and overfit model is seen in the left, middle and right column, respectively. Adapted from 365datascience.com.

2.2 Validation

To validate a clustering application, that is, showing that we have successfully discovered a reasonable representation of the underlying model, there are two main techniques: 1) external, and 2) internal validation. External validation requires a subset of the data to have "known labels" to compare to which can mean many different things, as discussed below. Internal validation revolves

around cohesion within a cluster and the degree of separation between different clusters, which is highly influenced by statistical assumptions both in the metric and clustering method.

For internal validation methods, a clustering application is successful if it can determine that the cohesion within a cluster is bigger than the degree of separation between it and other clusters. Many internal methods for verification of model skill exist, which broadly use information content, distance, density and neighborhood structure approaches. In brief, a metric is likely to favor a clustering application where the clustering method and the metric are based on the same statistical assumptions. I do not attempt a full review of internal validation methods here, but rather to give the reader a general sense. See Jenniges et al. (2025) for a comprehensive comparison of metrics within the NEMI framework, Arbelaitz et al. (2013) for a systematic review of over 30 metrics, and Schlake and Beecks (2024) for a general review. To illustrate the connection between clustering model and validation metric, consider k-means (MacQueen, 1965). The k-means algorithm is based on looking for round structures, see note on this below, and the Calisnki-Harabasz metric shares this underlying assumption in how is measures how well the clusters delineate groupings of the data. Briefly, distance based metrics assess the internal cohesion of determined clusters by assessing if all points in a cluster are closer to each other than to other points belonging to a different cluster, and examples include the Silhouette score, the Calisnki-Harabasz coefficient, and the Dunn index. Density and neighborhood-based methods include k-Density-Based Cluster Validation (k-DBV), Clustering Validation Index based on Nearest Neighbors (CVNNH), and Contiguous Density Region (CDR). The proliferation of validation techniques would suggest that validation would be straightforward. Similarly to the choice of clustering method, the choice of validation tools should be carefully considered. A key issue across the board for internal validation methods is that overlapping clusters and noise impact skill, which are features of most geospatial datasets (Jenniges et al., 2025; Arbelaitz et al., 2013). It is beyond the scope of this article to go through all of the above, but the example below will introduce information criteria. I use “round” and “Gaussian” loosely to mean unimodal and smoothly decaying from a center. Other distributions with similar properties, such as, Cauchy, or covariance structures, such as Matérn and exponential, would work equally well. This point further highlights the importance of preprocessing data carefully.

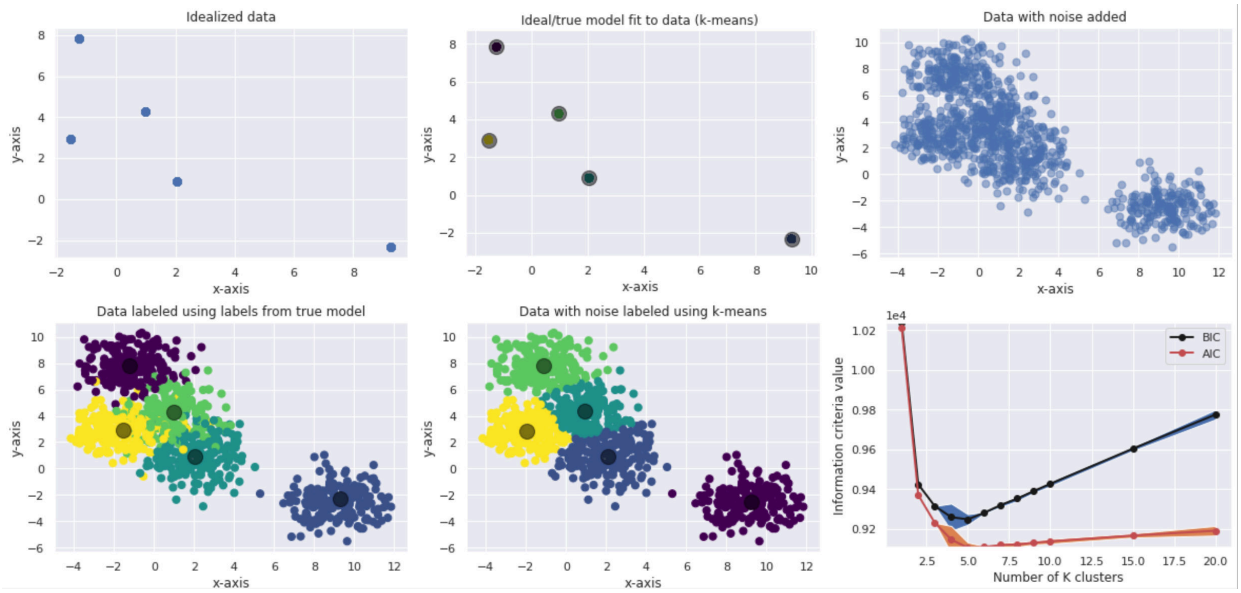

Figure 4. An illustration of concepts on idealized data. The top left figure shows randomly generated data around five points, top middle shows k-means applied to the data, top right shows the same data with added noise. Bottom left shows the true cluster assignments as determined by the initial clustering before noise was added, bottom middle shows the results of a k-means application, note the misclassifications. Bottom right shows the AIC and BIC values. Note that the colors are randomly assigned.

2.3 Practical example: k-means on idealized 2D data

A very simple and popular method for clustering is called k-means (MacQueen, 1965). I choose k-means to illustrate as many readers may be familiar with this method. In the example in this section, the 2D data are synthetic (Fig. 4), meaning automatically generated following prescribed conditions, described below. K-means is fast and conceptually simple, making it an excellent first choice for initial data exploration. The k-means algorithm involves an iterative minimization of the sum of squares of the Euclidean distance partitioning of the space given by data. Initially, the k-means algorithm makes a stochastic guess. This means that points are initially scattered across the data and the algorithm iterates until a "minimum" is found, which indicates an optimal partitioning of the data. This minimum is determined by minimizing the objective function J:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} ||\mathbf{x}_i^j - \mathbf{c}_j||^2, \tag{1}$$

where k is the number of clusters, n is the number of data points, the vector $\mathbf{x}_i$ corresponds to the dimensions of the data, here two, and $\mathbf{c}_j$ is the estimated location of cluster j. The number of k clusters is a free parameter that is chosen before the algorithm is applied. Initially, the cluster centers have random values scattered throughout the parameter space. Each cluster $j = 1, \ldots, k$ is represented by the characterizing vector $\mathbf{c}_j$, and the k-means classification attributes each vector $\mathbf{x}_i$ to a unique cluster c_j, so $\mathbf{x}_i = \mathbf{x}_i^j$. The distance between a data point is given by $\mathbf{x}_i^j$ and the cluster center $\mathbf{c}_j$ is determined as: $||\mathbf{x}_i^j - \mathbf{c}_j||^2$. In this way, each data point in $\mathbf{x}$ is associated with the closest k-cluster. Then, the position of $\mathbf{c}_j$ is calculated again, and the association is reassessed until the solution converges.

A key concept to note is that statistical assumptions can be incorporated through the objective function, here J. The k-means algorithm uses only one parameter, the number of clusters, and an initial stochastic guess for the location of the cluster centers. Effectively, k-means clustering minimizes within-cluster variances (squared Euclidean distances), which also entails that k-means would work *perfectly* if the data were separated into tidy clumps with Gaussian distributions (round). Unfortunately, very few data have this type of covariance space and suffer from interconnected and decidedly non-Gaussian (and nonlinear) statistics. Put differently, the strength but also the weakness of this clustering method is that it works by partitioning the data into Voronoi cells. Effectively, the algorithm can only draw straight lines to partition the data, working well on structures that are clearly separated, but not so well on more complex covariance structures.

Fig. 4 shows an idealized scenario that illustrates how k-means can be successfully used. Note that the colors are arbitrary in Fig. 4. The top left panel shows a dataset of tightly clustered points that are well-separated from neighboring clusters and each has a Gaussian distribution (round). The top middle panel illustrates how k-means is successfully applied to discover this correct underlying structure in the data (labeled 'true'). The top right panel shows the *same* data but with noise added. Using the labels discovered from the 'true' underlying model in the top middle panel, the bottom left panel shows where the data *should* be classified. The middle lower panel contain the classification results. The colors are arbitrary, but note that while there are some misclassifications, the performance of the model determined using k-means is reasonable.

As discussed above, validation is key to determining a successful ML application, and without validation an ML model can amount to statistical fiction. Each run of the k-means algorithm on the data results in one statistical 'model'. As such, we want to assess how well different models fit

the data, where we can vary the number of clusters (k) and different model initializations, which give different results due to the initial stochastic guess (note that some implementations use a fixed seed which masks this potential source of model error). To assess the fit of the model, we can apply a range of validation methods. I refer the interested reader to Jenniges et al. (2025) for an extensive and critical examination of a range of validation methods on ocean data. A key takeaway from Jenniges et al. (2025) is that internal validation methods can disagree and are subject to their own assumptions, and it is critical to apply several. For illustration principles, I here assess the 'fit' of the model using external validation and two information criteria (IC) which serves as an internal method example. The IC can tell us how 'complex' our k-means model should be, where increasing the number of k is equivalent to increasing the complexity. The IC illustrates whether we are capturing **more information** if we add another k, or if the maximum is reached and the model is complex enough. Recalling the example in Fig. 3 where accounting for every data point is not desirable, we see that this statistical model of the data is too complex, meaning it will not generalize well.

Different methods exist for estimating the IC, and here I will briefly discuss the Akaike IC and Bayesian IC (AIC and BIC respectively). The AIC and BIC have been very well studied, and are therefore often preferred. The likelihood function is the basis of both and is the primary tool for estimating the parameters of an assumed probability distribution given data (here the data we cluster on). The likelihood ($\mathscr{L}$) is defined as:

$$\mathscr{L}(\theta \mid x) = p_\theta(x) = P_\theta(X = x), \tag{2}$$

Here, X is a discrete random variable with probability mass function p depending on a parameter θ. If thought of as a function of θ, P is the likelihood function, given the outcome x of the random variable X. Suppose that we have a statistical model of some data. Let k be the number of estimated parameters in the model (for example the number of cluster guesses from k-means). Then, $\hat{L}$ is the maximum value of the likelihood function for the model. Then the AIC value is estimated as follows:

$$\mathrm{AIC} = 2k - 2\ln(\hat{L}). \tag{3}$$

With a set of different candidate models (for example, comparing models determined using different numbers of k clusters), the AIC with the lowest number will be the one that fits the data best. The goodness of fit is assessed by the likelihood function. To discourage overfitting, the penalty term ($2k$) increases as the complexity (number of k clusters) increases. As such, the AIC will in general asymptote, and a good model is determined when this happens.

The BIC also uses the likelihood function to determine the goodness of fit, but uses a different penalization to determine if the model is overfitted:

$$\text{BIC} = \ln(n)k - 2\ln(\hat{L}), \tag{4}$$

where n is the sample size. As discussed in Yang (2005); Harvey (1982), the AIC can overestimate the order, making k too high, where the BIC penalization term discourages this more strongly. See figure 3 for an example of how a model can fail to find an underlying model. In short, the AIC should asymptote, while the BIC should start increasing. A number of k somewhere between these two (if they both occur) could offer a good fit.

The AIC and BIC are inappropriate if the number of k is unmanageably large or is close to the number of data points, without having a reason to suspect it should be. The relative simplicity of the AIC and BIC compared to many other internal model validation methods demonstrates the difficult nature of assessing if a 'good' approximation of the underlying model has been found, and stresses the importance of applying appropriate judgment and additional checks including external ones. Note that the AIC and BIC are useful in many applications of model selection, for example, auto-regressive model estimation (Kaur et al., 2023; Sonnewald et al., 2018) as is commonly used without validation of the chosen order. The use of a statistical model without assessing how well the model approximates the data can be very unfortunate, including the fact that a model that is unnecessary complex is chosen or vise versa as discussed in Kaur et al. (2023); Sonnewald et al. (2018).

Returning to the idealized example in Fig. 4, the bottom right panel illustrates the use of the AIC and BIC, where the 'correct' number of k is 5. In Fig. 4 the AIC appears to asymptote, and the BIC to reach its lowest point before turning upwards. As such, 5 clusters are correctly identified as the optimal number. We can also see, by visual inspection, that five is a good partitioning of the data, which serves as simple external validation.

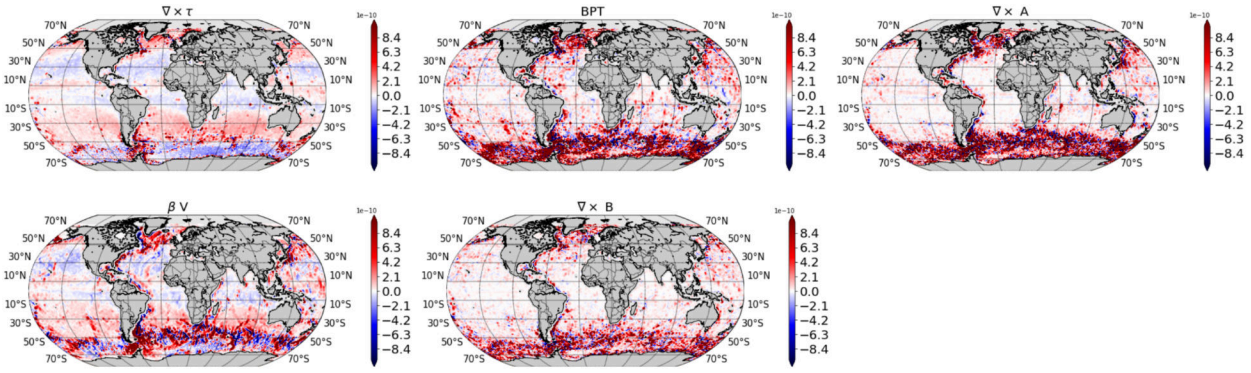

Figure 5. The terms of the barotropic vorticity equation. Each term is in $m\ s^{-1}$. Note how certain areas have clear large spatial patterns, while others can be highly variable. Top from left: $\nabla \times \tau$, $\nabla \times (p_b \nabla H)$ and $\nabla \times \mathbf{A}$. Bottom from left: $-\beta V$ and $\nabla \times \mathbf{B}$. See Fig. 6 for close-ups illustrating the complexity of the data further.

2.4 Practical example: k-means on realistic ocean model data

When using real data, recognizing where a dataset has come from and its basic statistics is a vital first step, known as exploratory data analysis, where the information from this initial analysis informs how the data are preprocessed. In this example, I use complex ocean physics data from a realistic ocean model to illustrate key introductory points. The preprocessing includes both a domain specific equation transform for initial dimensionality reduction, and a standard ML approach. The ocean model data will be used throughout the rest of the manuscript to illustrate the NEMI methodology. A more general introduction to preprocessing can be found in Bishop (2006), and see Furtado et al. (2025) for an article focusing on geophysical data.

2.4.1 The ocean model data: Exploratory data analysis and preprocessing

For the NEMI demonstration, I use a dataset from an ocean model (MOM6, Griffies et al. (2024a,b)). The ocean model is discretized in latitude and longitude, as well as in depth, meaning that the model equations are solved on a grid that subdivides the ocean area and depth. The area covered within each grid point varies widely. The data, approached naively, would consist of one point in depth, latitude, and longitude, where the model has 75 depth levels. We are interested in how the ocean is moving (as is described by the model equations in terms of momentum), and for each location in space this amounts to 39 different fields, where each field is one term in the equations that the model is solving, along with three additional ones at the sea floor. The equation terms can be thought of as

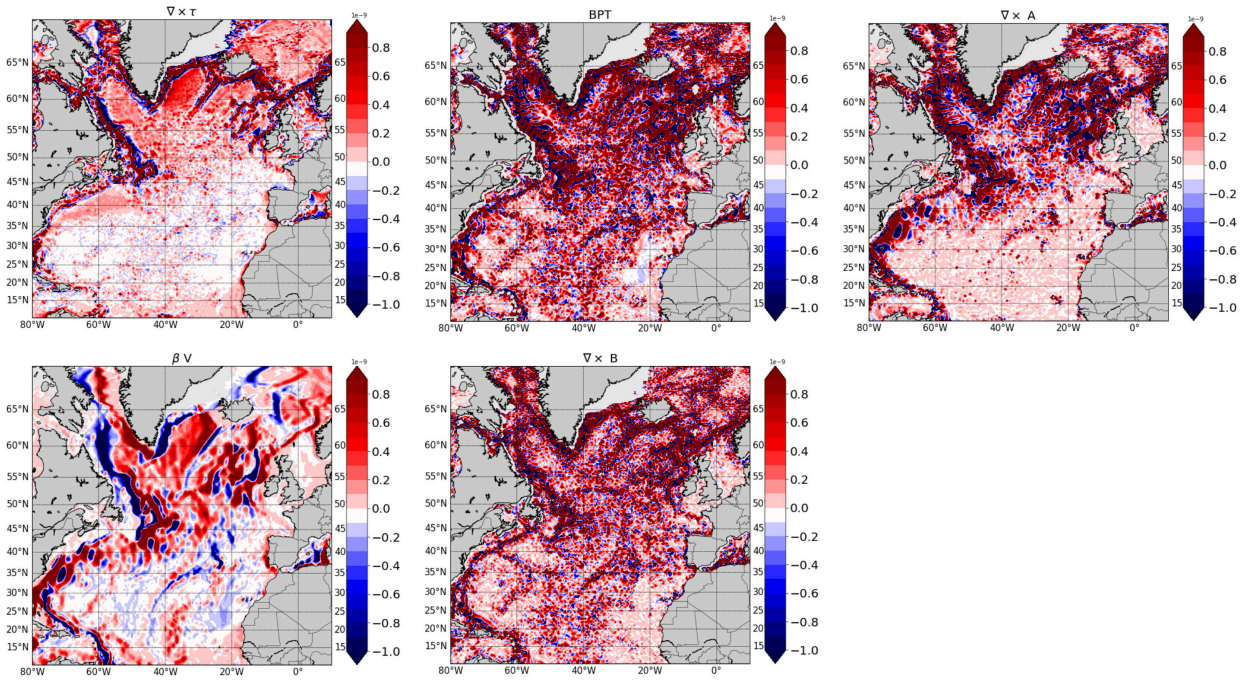

Figure 6. The terms of the barotropic vorticity equation, North Atlantic section. Each term is in $m\ s^{-1}$. Terms labeled as in Fig. 5.

our 'features' or 'dimensions', expressed using ML terms. As such, each dimension has a dataset made up of the various locations in latitude, longitude, and depth. See Khatri et al. (2024) for further details on the specific dataset. Using ML terminology, the data I use for my example, which is a closed set of equations described below, corresponds to a 'dimension' or 'feature' of the underlying dataset. Thus, a 'dimension' does not refer to a geographic dimension such as latitude, longitude, or depth, but an abstract dimension in feature space. Although each point in this feature space corresponds to a location in longitude, latitude, and depth, two points that are near each other in this feature space are not necessarily near each other in geographic space, which facilitates grouping of similar points. NEMI carries out clustering in the abstract feature space, however it is defined.

Working in a space consisting of 39+3 dimensions is challenging, so we initially make the data more manageable using oceanographic theory. This can be thought of as simplifying the latent space within the data and is highly field-specific. However, most numerical model output has undergone some field specific processing before being given to users. The latent space refers to the space given by the variables which contains the hidden patterns and relationships we wish to determine using ML methodologies. For initial domain-specific dimensionality reduction, I use the barotropic

vorticity (BV) equation terms as the data for the example in this section and throughout the NEMI demonstration, effectively applying an equation transform to the data from the 39+3 dimensions into 5. This was described in detail in Sonnewald et al. (2019), and briefly below. This equation transform, moving the data into the BV framework, also facilitates external validation, as we will see later. Despite being simplified, the data are still highly complex. The global terms (Fig. 5) illustrate that large areas have small magnitudes, while others are comparatively intense, illustrated further in a closeup of the North Atlantic (Fig. 6). Having such large differences in terms and complex interactions between terms is at the core of why the data are challenging to analyze with conventional methods. The BV data are output from a fully realistic numerical ocean model.

Using the framework of the BV equation has a long history in oceanography. Early works (Sverdrup, 1947; Munk, 1950; Stommel, 1948) recast the intractably complicated full equations to describe how meridional ocean flows develop by taking the curl of the depth-integrated (barotropic) momentum equations, thereby arriving at the BV equation. The steady BV balance under incompressibility is expressed as:

$$\beta V = \nabla \times (p_b \nabla H) - \nabla \times \tau + \nabla \times \mathbf{A} + \nabla \times \mathbf{B}, \tag{5}$$

where $\beta = \partial f / \partial y$ is the northward derivative of the Coriolis parameter (f), $V = \int \rho v \mathrm{d}z$ is the depth-integrated northward mass transport from density ρ and meridional velocity v, ∇ is the horizontal gradient operator, p_b is the pressure at the bottom, and $H = h + \eta$ is the water column thickness, where h is the distance from the resting ocean surface to the bottom topography, and η the sea surface height anomaly. The stress produced by wind and bottom friction (external) is denoted by τ, and $\mathbf{A}$ and $\mathbf{B}$ are the depth integrals of the nonlinear and the horizontal viscous terms, respectively (Hughes and de Cuevas, 2001).

Returning to the dimensions of the data and how these relate to the grid, recall that the data used here is a parameter, say x, that is a vector field defined at every grid cell (lon, lat) on the discretized MOM6 ocean sphere. Each element x_i represents a five-dimensional vector, given by the terms in the BV equation, on the horizontal grid of the model. The index i uniquely identifies a grid point on the sphere, with (lon, lat) = (ϕ_i, θ_i).

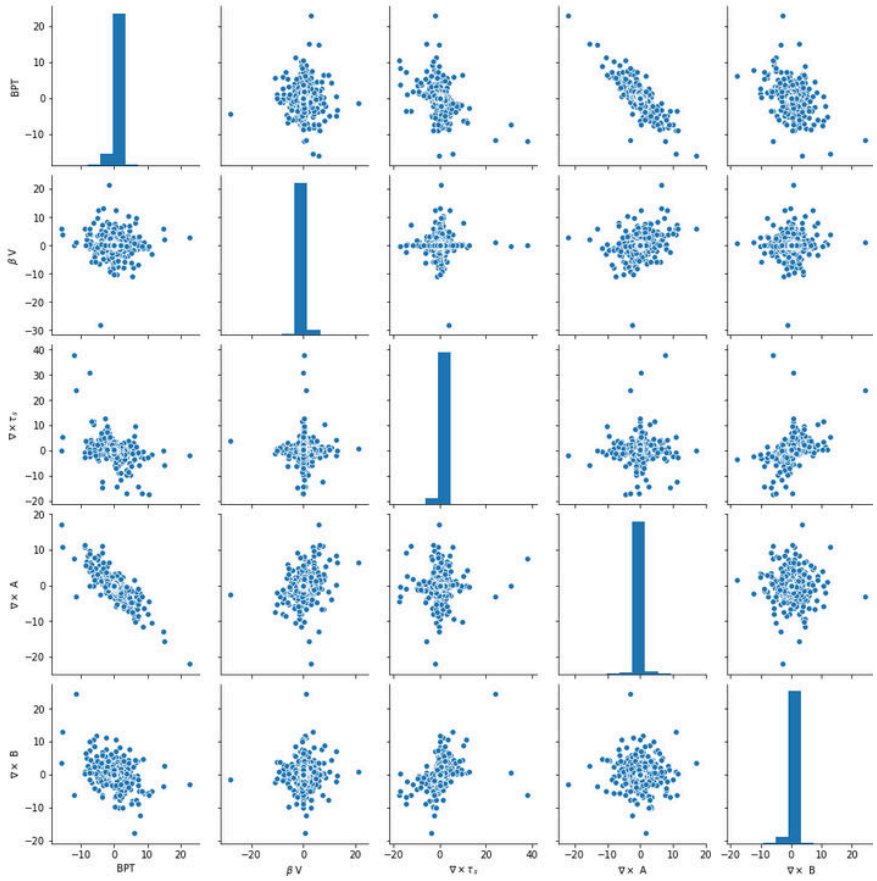

Figure 7. The scaled BV data. Each variable in the BV data are plotted against the other.

2.4.2 Common problems with realistic data

In the example featuring the BV data from the numerical ocean model MOM6, a very complex problem is chosen. In general terms, data can suffer from issues such as: 1) noise, which is meaningless information in the data that masks components of interest, and sources can include instrument error or numerical artifacts, and 2) sparseness, where only part of the desired data are available. Examples include the wealth of data available at the ocean surface, but difficulty in acquiring subsurface data; 3) lack of balance, which refers to data that has a wide range and only a small proportion of information of interest, for example, a global dataset where one wishes to detect episodic ocean convection happening in small areas.

The extent to which data are afflicted by various issues is often unknown, and checking the nature of the data extensively before starting analysis is always advisable when using unsupervised learning. Note that best practices can differ, for example, for supervised learning, and task-specific considerations are advised. I will briefly illustrate common principles and best practices using

the BV data, and refer the reader to Furtado et al. (2025) and Bishop (2006) for more. For the BV data, issues 1 (numerical noise) and 3 (small areas that are interesting) described above are intuitively recognized as problematic. To illustrate the data, see Fig. 5 and 6, where a smoother with a Gaussian kernel with a standard deviation of 1 has been applied to the BV data in geographical space. The data must be appropriately cleaned and preprocessed to identify useful clusters successfully. Standardizing and normalizing is a simple first approach; for example, one can scale as $z = (x-u)/s$, where z is the scaled data, x is the original data, u is the mean and s is the standard deviation. This is done separately for each dimension or equation term. To illustrate the effect applied to the BV data, we arrive at the pair plot shown in Fig. 7. The pairplot in Fig. 7 shows each dimension (here each term in the BV equation) as a scatter plot in respect to each other term, with the associated probability distribution function of the dimension as a barplot across the diagonal.

What we are looking for are meaningful relationships between the dimensions, where if we only saw vertical or horizontal lines there would be no interesting relationships between the variables. Fig. 7 shows more varied relationships. Note that the individual distributions only give a vague representation of the data density. Many other methods for data-scaling exist that are suited for e.g., log distributed data. Experimenting with the initial scaling can be highly beneficial. The rationale behind scaling and normalizing is that the covariance between variables has more utility than their individual magnitudes. For example, consider the global data of ocean temperature and fish stock abundance, where the magnitude of variability in temperature is small compared to the magnitude of variability in fish stock abundance. Without scaling, the temperature variable would appear meaningless for fish stock abundance, even though we expect a difference between Arctic and tropical regions. After getting to know the data through initial inspection and scaling, we are ready to consider methodologies for further exploration. Note that the underlying structures within the data remain the same, and determining a successful statistical model should find the same regions. However, what the model finds is made subjective based on the choice of scaler. To illustrate, if a method is chosen that highlights outliers, one may be more likely to find deep convection events in the ocean, as more common patterns would be condensed and more challenging to partition. In this manner, the choice of scaler can depend both on what the practitioner is focused on as well as the tolerance for uncertainty, as I discuss later in the manuscript.

2.4.3 Evaluation of the k-means algorithm on realistic ocean model data

With the BV data from the MOM6 model, I can illustrate the performance of the k-means algorithm. Using k-means on the BV data and assessing the model fit with the AIC and BIC we can see an immediate difference in Fig. 8 for the spatial distributions in panel a and b, and the AIC and BIC in panel c. To illustrate spatially what k-means does, Fig. 8 shows the geographical patterns associated with k set to 50 and 200. With k=50, most of the ocean is put in one cluster, and the rest appears like noise, with k=200 most of the ocean is still in one cluster, the regions of noise have expanded, and only one more distinct cluster has appeared. Panel c demonstrates that k-means is not an appropriate model for the data. This is evident in that the AIC has not become stable after adding even 350 k, and while the BIC has started turning upwards the standard deviation (shown in the blue shading) is fairly large. A visual illustration of how k-means partitions the BV data can be found in Section 3.1.2.

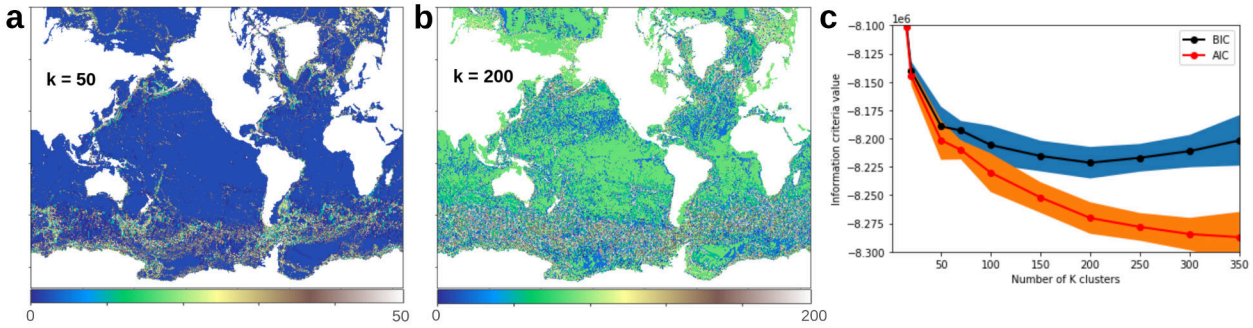

Figure 8. An illustration of running k-means on the BV data. To the left in panel a) a k of 50 is chosen. In the middle in panel b) a k of 200 is used. Panel c) shows an illustration of the AIC and BIC run on the BV data. Note that the AIC fails to converge and the BIC stays fairly flat. The AIC and BIC indicate that the k-means algorithm is not converging as the AIC keeps decreasing. From a and b, we can see that the added complexity has not resulted in much physical insight compared to more useful large scale rationalizations as in Sonnewald et al. (2019).

3 THE NATIVE EMERGENT MANIFOLD INTERROGATION (NEMI) WORKFLOW

Below I describe the components of the NEMI algorithm as outlined in 1.1 and in Fig. 2. As discussed in section 2.4, preprocessing the data before applying NEMI is beneficial. The roman

numerals within the titles of subsections 3.1 and 3.2 refer to those in the NEMI workflow description from section 1.1. Throughout for a practical example, we use the data from the global realistic ocean model MOM6, preprocessed as described in section 2.4. The data are referred to as the barotropic vorticity (BV) equation terms.

3.1 Inner workflow: Manifold learning

Starting with the inner workflow, it is important to note that several iterations are almost always necessary to deliver robust and meaningful results. This is because NEMI utilizes an ensemble methodology to quantify uncertainty, which also facilitates parameter tuning, where the inner workflow should be run numerous times until sufficient uncertainty characterization is established, as described in the outer workflow below.

3.1.1 I: Manifold approximation of the underlying covariance structure in data

The data embedding used in NEMI has been found to unequivocally improve clustering results (Jenniges et al., 2025). Because of the importance of validation, the NEMI methodology enables this in multiple forms. The first is an assessment of the latent space of the data. The latent space is a term for the covariance structures in the data that are hidden from our human perception by being too complicated, high-dimensional, or both to be easily perceived.

To characterize the latent space, approximating it in terms of a manifold is very useful. Intuitively, a manifold assumes that there are relationships within the data, these relationships can be leveraged for simplification. For example, looking at the BV data, there are often interactions between the wind stress curl and planetary vorticity advection terms while other terms are less important. A mathematical manifold is a construct from topology: any local point resembles the Euclidean space near each point. Effectively, the 'distances' between different data points are used to determine the relationships between different points. For example, a parcel of water near the equator is likely warmer than one near the Arctic, suggesting that they are not closely related in temperature, but they could both have low biomass compared to other locations, making them closely related along the biomass dimension. A characterization of distances between data points has the convenient property, which makes it useful in NEMI, that the space is homeomorphic. This homeomorphism means that one shape can be transformed into another, without violating the relationships between the data points. One common example is that a doughnut (torus) can be transformed into a coffee mug, as both have one hole. It is beyond the scope of this article to give a thorough introduction to

topology, but the key concept is that a confusing ball of covariances, or a very complicated latent space, can efficiently be simplified **without losing the nonlinear structures**. For a visual example, imagine a scarf crumpled on a table. The scarf has various patterns that may look oddly disjointed when tangled together, for example, if sections of polka-dots are inter-spaced with stripes. However, if the scarf is spread out, the complicated 3D structure becomes a smooth two-dimensional object with clearly separated polka-dots and stripes fully visible.

NEMI employs the Uniform Manifold Approximation and Projection (UMAP, McInnes et al. (2018)) methodology with three key benefits: First, it can be used to reduce the dimensionality with relevance to visualization and validation. The data are projected onto three (or two) dimensions, allowing visualization. Visualization allows an additional external validation step, discussed later. Second, it can 'strengthen' associations between different areas of the data, allowing relevant patterns to emerge more clearly, and with the specific method used here it can be tuned to the practitioners' interest. Third, the projection into a lower-dimensional space has a stochastic element and utilizes optimization. This element allows for quantification of uncertainty as discussed below. Here, I describe NEMI using UMAP, but other methods can readily be applied including the t-Distributed Stochastic Neighbor Embedding (t-SNE, van der Maaten and Hinton (2008)) and Self Organizing, or Kohonen Maps (SOM, (Kohonen, 2004)). UMAP is the method of choice in NEMI due to computational efficiency and the formulation of the optimization process, discussed in detail below. Jenniges et al. (2025) discussed choice of projection algorithm in NEMI further, and Nanga et al. (2021) provides a review of methods.

Characterizing the underlying manifold in the data

To illustrate the benefits of using UMAP, it is useful to visit the underlying theory in descriptive terms. See McInnes et al. (2018) for a thorough description of the underlying mathematics. Constructing the manifold, we start with simple combinatorial building blocks (called simplices) of the distances between the data points. One data point is a 0 simplex, two connected points are a 1 simplex (line), three connected points are a 2 simplex (triangle), a 3 simplex has four connected points (pyramid), and we can continue upwards adding dimensions. We can construct different simplexes and combine these, and in practice, the simplexes do not need to have very high order to cover their local space. This is different from a k nearest neighbor graph because the choice of the radii can have a detrimental impact on the k nearest neighbor graph's ability to approximate the underlying

space, which is amplified if a dimensionality reduction is attempted. If the space were uniformly sampled within its dimensions, the choice of radii would not be problematic, but this is highly improbable using realistic data. As such, we can only assume that our data are not uniformly distributed. Using Riemannian geometry the non-uniformness can be leveraged to our advantage, as described in the following section.

Using UMAP, we **assume** that we have data that is sufficiently uniformly distributed such that the actual distances between the data points can be used to create a map of the underlying manifold. Most data is not perfectly uniformly distributed, and this attribute, different for each dataset, is a key source of uncertainty. Effectively, to map out the manifold we choose a unit 'ball' about a point that stretches to the k-th nearest neighbor of the point, where k is the sample size we are using to approximate the local sense of distance. I use 'k' to conform with the overall ML literature, but note that this is distinct from the k in k-means. In UMAP, each point is given its *own* unique distance function to its neighbors as determined by radii needed to span its k-neighbors. To illustrate this, we now add to the concept of the manifold that it is locally connected, meaning that it describes one space, rather than a set of disconnected spaces. However, in a simplified sense, because we looked at the neighboring points to assess the distances, two neighboring points may individually have different values describing the same distance, because they used a different reference point.

The impact of using different reference points is that two points that are next to each other may have different distance functions to each other. As such, a useful mental construct to envision this set in UMAP is to think of it as a weighted graph, where the weights describe the distances. If there are conflicting weights associated with the simplices, we interpret the weights as the probability of the simplex existing. Thus, we want to merge the weights to just have one combined distance. This is referred to as taking the union. In simplified terms, if we have disagreeing weights a and b, then we should have a combined distance $\mathbf{a} + \mathbf{b} - \mathbf{a} \cdot \mathbf{b}$. A way to conceptualize this is that the distances are the probabilities that an edge (1-simplex) exists between the two. The combined weight is the probability that at least one of the edges exists.

Embedding a manifold onto a lower-dimensional space

Having discussed the implications of characterizing the 'neighborhood' of a dataset, let us review how these qualities can be used to project the dataset onto a lower dimensional space. Embedding the manifold into a lower-dimensional space can now happen based on the notion that we have

the information about the manifold approximated by the data points, and we wish to conserve the associated probabilities between the data points in the lower-dimensional space. In essence, we can now compare the original topological structure of the manifold with a lower-dimensional candidate. Both would share the same 0 simplices, and we can imagine that we are comparing the two vectors of probabilities indexed by the 1-simplices.

For this assessment, we use the cross-entropy as our metric, which is a term used with slightly different connotations between fields. The term cross-entropy is originally from information theory, but in ML it is often used interchangeably to refer to the loss function in supervised ML. In information theory, the cross-entropy is a concept measuring the difference between two probability distributions, specifically the inefficiency when coding one distribution using optimal code from another. The cross-entropy measures how many bits (on average) you need to encode messages from one distribution when using a code optimized for a different distribution. This code can be thought of, loosely as the weights being trained, and the target distribution can be thought of as the training labels. Because the labels are fixed, the formula used can strictly be referred to as the Kullback-Leibler (KL) divergence. For more information on the terminology and concepts see Mackay (2003). Following UMAP's terminology, we will use the term cross entropy. Other methods, such as the t-SNE (t-Distributed Stochastic Neighbor Embedding, van der Maaten and Hinton (2008)), discussed below, use the KL-divergence more explicitly.

To estimate the cross-entropy, say the set of all 1-simplices is E, and we have arrived at weight functions so the weight of the 1-simplex e is $w_h(e)$ in the *high* dimensional case. Now $w_l(e)$ is the weight of e in the *low* dimensional case, and the cross entropy will be:

$$\sum_{e \in E} w_h(e) \log\left(\frac{w_h(e)}{w_l(e)}\right) + (1 - w_h(e)) \log\left(\frac{1 - w_h(e)}{1 - w_l(e)}\right). \tag{6}$$

Now, we minimize the cross-entropy to arrive at our low dimensional embedding of the high dimensional manifold, where the weights are the internal parameters being optimized that allow us to do this. Here, the first term $w_h(e) \log\left(\frac{w_h(e)}{w_l(e)}\right)$ can be thought of as a force that attracts the points whenever there is a large weight associated in the high dimensional case. If $w_l(e)$ is as large as possible, the term will be minimized. This occurs when the distance between the points is as small as possible, and effectively when the UMAP algorithm is focusing on the very local structure. In contrast, a repulsive force is found in the $(1 - w_h(e)) \log\left(\frac{1-w_h(e)}{1-w_l(e)}\right)$ term between the ends of e

when $w_h(e)$ is small. The minimization is an iterative process, where the optimal weights are the internal parameters arrived at are sensitive to the initialization, as described below. NEMI utilizes this by running the optimization multiple times, once for each ensemble member. This results in an ensemble of embeddings where each member has a separate set of optimal weights as the product of a separate optimization with a different initialization. If an embedding is stable, the optimization should have arrived at very similar weights, and the difference between different runs should be small, that is, the uncertainty will be low.

Using embeddings on realistic data

The notion of uncertainty is key, but together with the core assumption in UMAP, is something we use to our advantage in NEMI. A central limitation of such 'manifold' based methods, and why different initializations arrive at different weights, is that we are assuming that our data points populate the manifold of the underlying model sufficiently to be able to characterize the latent space. For example, if we think about a landscape with mountains, we may have fewer data points among the mountains than in the surrounding areas that are also 'smoother'. A manifold representation of the landscape based on incomplete data would likely miss interesting features of the mountainous regions. Most datasets have limitations, and it follows that estimating the uncertainty associated with, for example, the location of cliffs in the mountain example above would be advantageous. NEMI utilizes ensemble methodologies and UMAP to do exactly this, as described further in section 3.2.2. Key to estimating the uncertainty is that UMAP has a stochastic component associated with initializing its *internal* parameters (weights) before optimization, as described above. This stochastic component comes from that, following setting the hyperparameters of the algorithm that are fixed throughout each ensemble i.e. `min_dist` and `n_neigbhours`, UMAP performs an optimization to determine its *internal* parameters. Note that each ensemble uses the same `min_dist` and `n_neigbhours`. If a good combination is found, this results in similar embeddings. For clarity, we will refer to the hyperparameters simply as parameters throughout the text when discussing the embedding.

UMAP is similar to other methods such as t-SNE. NEMI as presented here uses UMAP, but note that the t-SNE method was used in Sonnewald et al. (2020). Both UMAP and t-SNE have drawbacks, and one should weigh carefully if these are appropriate for the data. These include that t-SNE, like UMAP, does not completely preserve density. UMAP, like t-SNE, can also create tears

in clusters that should not be there, resulting in a finer clustering than is necessarily present in the data. Overall, such issues are exactly why NEMI was developed with additional validation steps. As such, NEMI uses both external and internal validation.

Practical example: The BV ocean data

What does a UMAP rendition of the highly complex and complicated BV data look like? Fig. 9 shows a three-dimensional rendition from different angles. The shape can vary depending on the parameters chosen, as stressed above. In Fig. 9 we can see that there are clear areas that, from all angles, are more dense and some that are more sparsely populated. We will use the visualization to choose a clustering algorithm below. The sensitivity to parameters (or how 'brittle' the method is) is highly dependent on the data's complexity and how well the latent space is sampled, as described in the above sections. In this example, a large ensemble sweeping through the UMAP parameters was needed to arrive at a reproducible manifold representation. Effectively, this could be seen as in the outer loop, see section 3.2, but I discuss it here as its success, or stability, is more easily explained in this section. The concept of a reproducible manifold means that one should be able to run the algorithm on the data and recover the same (or sufficiently similar, see section 3.2) structure. Here, small differences can have a large impact and they can be difficult to determine by eye. The importance of small differences is part of the reason why NEMI employs additional checks and leverages the associated uncertainty. Fig. 10 illustrates three renditions of running the UMAP algorithm on the processed BV data. Each time, UMAP is initialized with a stochastic component, and thus it is possible that very different embeddings result. The plots in panels a-c in Fig. 10 may look very similar to the human eye, but printing the associated arrays illustrates the slight differences. The array here refers to the three dimensional dataset of the embedding. For example, the first number in the array goes from 7.895877 in the manifold in Fig. 10a, to 7.892489 and 7.875971 in b and c respectively. These differences may appear small, but they are present and can skew the results. Determining the acceptable and appropriate level of difference is critical to the success of NEMI, and is explained further in section 3.2.2.

3.1.2 II: Clustering to leverage the manifold approximation of data

The use of manifold and dimensionality reduction methodologies in NEMI leads to a three-dimensional rendition of the data, which can be visualized (Figs. 9 and 10). When choosing

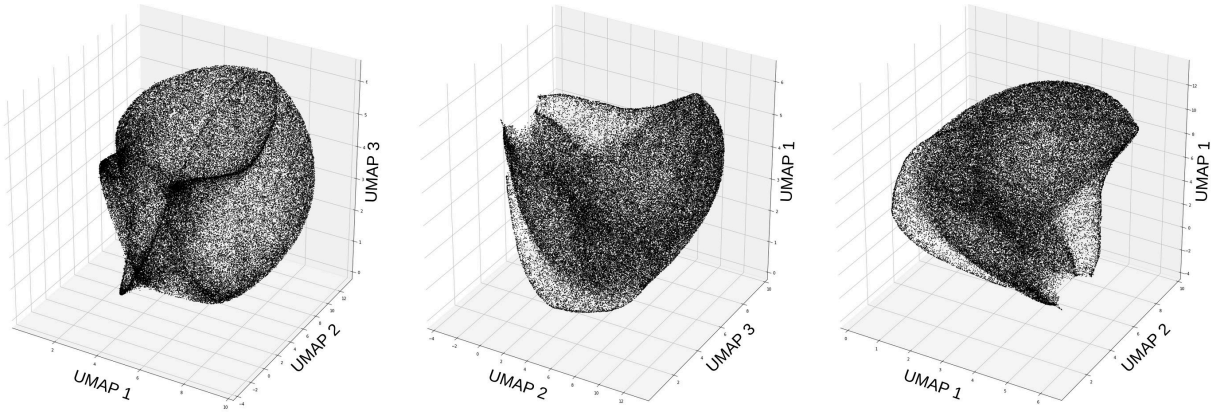

Figure 9. One UMAP manifold from different angles. UMAP 3D embedding of the five-dimensional BV ocean data.

a clustering algorithm, this visualization greatly assists in the choice of algorithm, making it visually apparent if an algorithm needs to be able to handle data that is: 1) not well-separated (e.g., one continuous-seeming structure), 2) highly nonlinear, or 3) of varying densities meaning that the points are more likely to be found in certain areas. Which clustering method is chosen, and a demonstration of the fidelity improvement of working on embeddings can be found in Jenniges et al. (2025).

There is a growing number of different clustering algorithms available to the practitioner, and NEMI is in principle method agnostic as mentioned above. For the purpose of illustration in this manuscript, NEMI uses a hierarchical cluster analysis (HCA), and the clusters found by the ML method will hereafter be referred to as 'HCA clusters'. Specifically, an agglomerative methodology initially assumes that each data point is its own cluster, and pairs of clusters are merged as one moves up the 'hierarchy'. This is a 'bottom-up' approach, whereas a 'divisive' approach would be the opposite ('top-down') and assume that the initial step is to have one cluster represent the whole dataset and proceed to divide the data. Note that the agglomerative clustering methodology is not stochastic. The hierarchical element is useful as it means that running the algorithm on the same data will not introduce uncertainty in what clusters are found. Using a hierarchical method is intuitively useful both for global (for example, the whole earth in the present example) or more local applications (such as a basin or more regional assessment).

The agglomerative hierarchical clustering methodology is presented as a cartoon in Fig. 11. Here, Fig. 11a shows the data points 1 to 6 in a two-dimensional space (here 'UMAP 1' and 'UMAP

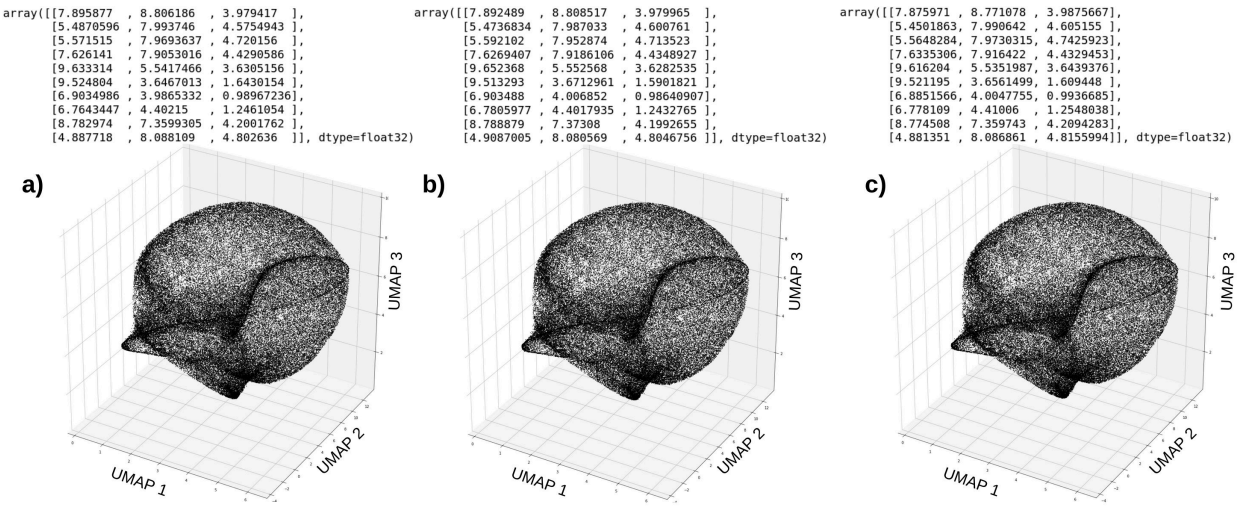

Figure 10. Three different ensemble members, with a part of the associated data. Note that while the manifold renditions look very similar, the data associated highlights the slight differences.

2' for simplicity in relating to the present section, although this is strictly a cartoon and UMAP was not applied). The data points have a certain distance from each other within this space. In Fig. 11b, initially each data point is progressively grouped in relation to the distance between the points in panel a. Points 4 and 5 are initially grouped, as are 3&2, while 6 and 1 remain isolated. At the next aggregation level, 6 is brought into the 5&4 cluster, becoming 6&5&4. The other points remain disaggregated, as the distance between them is still too large (see Fig. 11a)). At the next level, the 3&2 and 6&5&4 clusters are merged into 6&5&4&3&2. Finally, data point 1 is brought into the cluster at the next level, which now includes the entire data set. Note that the level of agglomeration (the yellow lines in Fig. 11b) do not directly correspond to the number of clusters.

Having chosen an agglomerative methodology, I will highlight two hyperparameters that are of greatest relevance to the practitioner. The choices are that of the distance metric and linkage method, and I will discuss the simplest as my goal is to illustrate the principles that can enable a practitioner to choose something appropriate for their problem. Note that the below discussion is specific to the HCA in the example, and see Jenniges et al. (2025) for other demonstrations. The distance metric is an expression of how the separation of the points is quantified. To illustrate, imagine a room of people, such as an auditorium with a lecturer on a podium and students sitting at a distance. If grouping the people using physical distance, the students would be clustered together because the gap between the students would be smaller than the distance to the lecturer. That is, the gap

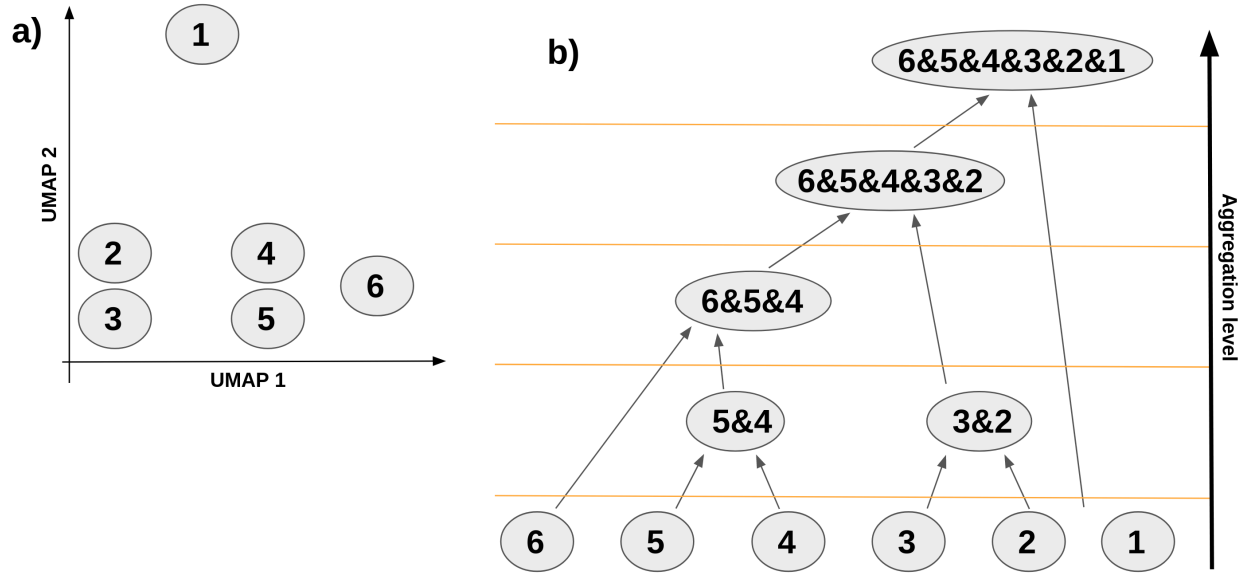

Figure 11. Sketch of the agglomerative clustering functions. Panel a) shows a number of data points 1 to 6 in 2D embedding space, with the dimensions denoted UMAP 1 and UMAP 2. Panel b) shows these points being aggregated by an agglomerative method from single points to one cluster encompassing all points. Note that different 'agglomeration levels' corresponds to different numbers of clusters, as it is the distance between the clusters in embedding space that determines the degree of grouping.

between the teacher and the closest students would be a defining feature of the data. However, if one used a metric such as how well the students know each other (e.g., how many friends they have in common), there would likely be clear groupings within the students. As such, the distance metric chosen should be considered carefully. In NEMI, the use of the manifold methodology and input data consisting of a closed momentum budget, meaning that all terms are accounted for, allows us to directly link the distance in UMAP space (seen in Figs. 10 and 9 as the distance between points) to the clustering. Using a Euclidean metric effectively makes the algorithm use Pythagoras's theorem when operating in Cartesian coordinates. Note what is often used is the squared Euclidean distance.

The second hyperparameter of interest, the linkage method, groups points as described above and illustrated in Fig. 11. The choice of the linkage method is often dictated by computational capacity. Note that methods scale differently with the size of the dataset. Here, the simplest method is single linkage, which defines the distance between clusters as the distance between their closest pair of points. Single linkage guaranties that connected components are preserved, but scales as

$\mathcal{O}(n^3)$ where n is the number of points and should be avoided unless the dataset is very small. In NEMI, the default is the Ward linkage method (Ward, 1963). Ward's method uses a minimum variance criterion that minimizes the total within-cluster variance. Let d be the distance between points i and j in data vector $\mathbf{x}$. The initial distances in Ward's method are squared Euclidean distances between points:

$$d_{ij} = ||\mathbf{x}_i - \mathbf{x}_j||^2. \tag{7}$$

Ward's method works by combining clusters to minimize the increase in total variance within the cluster after merging. This increase is determined as a weighted squared distance between the cluster centers. The success of the HCA is sensitive to the linkage method, so while Ward worked well for the BV example, a manifold featuring concentric circles may require a different method. A key strength of NEMI is that this can be visually assessed.

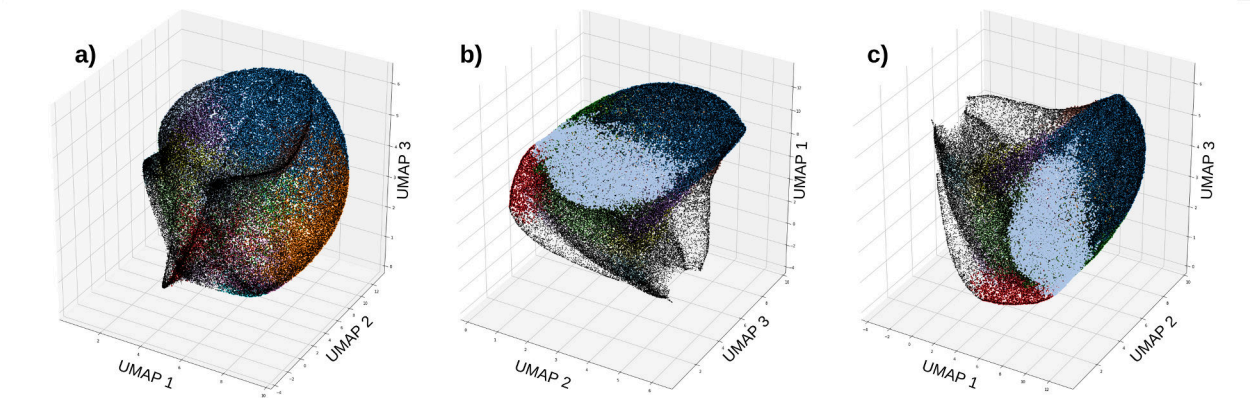

Figure 12. The agglomerative clustering on UMAP embedding. Panels a, b, and c show the same manifold from different angles. See the sub-sampled version of a) in Fig. 13 to highlight shapes that are picked up by NEMI.

In Fig. 12 the application of the hierarchical clustering to a UMAP rendition is illustrated from a few angles (panels are from the same manifold with the same clusters). The colors indicate the different clusters (more detail on this below) and show how the clusters successfully isolate the ridges running along the sides of the data (see Fig. 13 for a sub-sampled version of panel a from Fig. 12 where details are highlighted). Note also that Fig. 12 displays one arbitrary iteration (i.e., ensemble member) of UMAP, with clusters determined on another UMAP ensemble member, illustrating how well the method performs.

It is valuable to contrast the performance in Fig. 12 with the earlier k-means example. In Fig. 14, a k-means rendition with 200 k, which could be seen as reasonable using visual inspection in section 2.3, is displayed on the manifold used in Fig. 12 and 13. In Fig. 14, the pale and translucent colours were chosen to enhance the readability due to the large number of colors. To produce Fig. 14, the k-means clustering was performed on the BV data before the UMAP algorithm, and subsequently projected onto the UMAP manifold. Each data point is projected onto three dimensions from the five-dimensional input space, as the number of data points remain the same, but the number of dimensions change. In Fig. 14 colors do not delineate the areas that are observed to be grouped together; this is a visual demonstration of how k-means fails to identify key regions, and concurrent with the AIC/BIC internal validation methods in Fig. 8c. To illustrate what k-means did in the five-dimensional space further, Fig. 15 repeats the k-means application on a three-dimensional embedding. Fig. 15 illustrates that k-means is forced to artificially separate the data coarsely using 'straight lines' across the entire data volume. Recalling that the UMAP rendition of the BV data are used to 'simplify' and 'clean' the data, it becomes apparent how difficult it would be to apply k-means to the non-transformed data. In supplement to the information criteria, this additional visual appraisal of the performance of the algorithm underscores that the k-means algorithm is a poor choice. This method of validation can be applied widely beyond the examples used here, and further illustrations can be found in Jenniges et al. (2025).

In practice there is no guarantee of finding an optimal solution

As with most clustering and ML applications, there is no guarantee of finding the optimum solution. There might not even be one, and in this case it is especially important to determine this. However, if an optimum does exist for the agglomerated clusters, it is guaranteed to be found via single-linkage. Due to computational costs, the application of single-linkage is largely impractical. Other methods, such as the Density-based spatial clustering of applications with noise (DBSCAN, Ester et al. (1996)) used in Sonnewald et al. (2020) and Jenniges et al. (2025) can be advantageous, especially if the data are more separated. Note that DBSCAN performs considerably better, in terms of scaling to larger datasets, so this method, or the hierarchical version of the same is recommended. I encourage the reader to apply a range of algorithms following the visual inspection of the embedding.

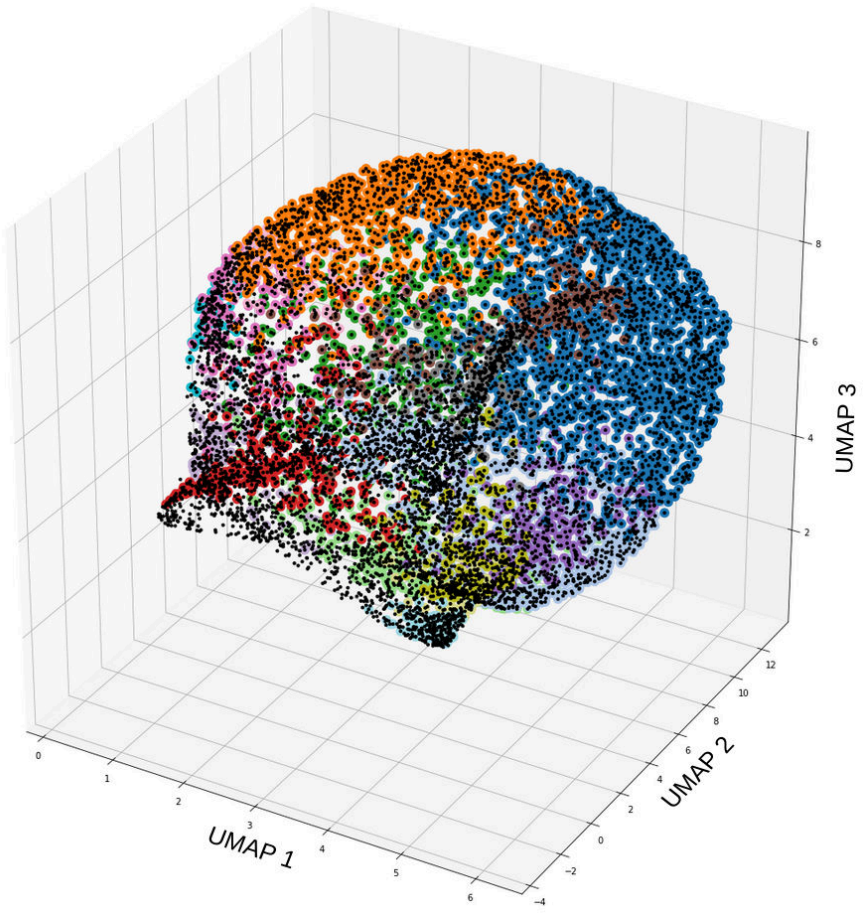

Figure 13. The agglomerative clustering on a UMAP embedding, heavily sub-sampled. Illustration to supplement Fig. 12. To demonstrate how well clusters identified on one embedding fit a different embedding, I chose the cluster assignments and embedding from different ensemble members.

3.2 Outer workflow: Ensemble assessment

The main role of the outer workflow is to provide further validation and quantify uncertainty, which can then be used to either fine-tune parameters, such as the UMAP `min_dist` and `n_neighbors`, or guide exploration of the identified regimes. This validation happens both using a statistical, or probabilistic, approach as well as from a final domain-specific assessment, or external validation. Overall, the threefold approach is: 1) assessing the success of a clustering through visualization of the clusters on the various embeddings, largely happening in the inner workflow, 2) determining if the uncertainty across the ensemble is acceptable, and 3) determining if the final clusters correspond to known patterns in the input data. The visual check of the clustering algorithm chosen in NEMI,

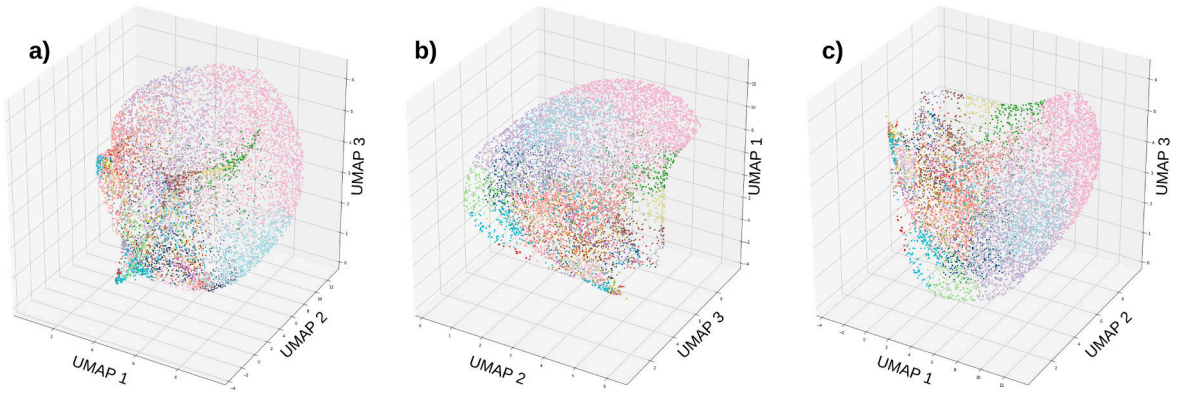

Figure 14. The k-means algorithm with k = 200 result projected onto a UMAP manifold. Panels a, b, and c show the same manifold from different angles. Note that the clusters should be coherent on the manifold if the method is successful. Note there is poor coherence and the clusters are somewhat arbitrarily separating chunks of the space. This confirms earlier suspicions that the k-means algorithm was not succeeding in arriving at a good model representation.

as discussed in section 3.1 in the inner loop, is the first component of the validation. The second validation step is related to the use of embedding and how stochasticity is leveraged to assess uncertainty, discussed in section 3.2.2. In the final validation step, discussed in section 3.3, we use domain-specific expertise, here demonstrated using oceanographic theory.

3.2.1 I: Validation via cluster agreement

For validation and utility, let us return to a concept introduced in Sonnewald et al. (2019) in relation to cluster validation and assessment. It is obvious that having a model that is a good fit to the data can be unhelpful if it is not able to address the research question at hand. In a geoscientific context, it is nominally key that the same regions appear consistently, and that, for example, the Sahara and Antarctic desert areas are not mixed up because they both have low precipitation, unless this is allowable due to the research question as discussed in section 3.3. Here, as in Sonnewald et al. (2019), it is critical that the algorithm can robustly recover and reproduce geographical sub-regions. Namely, if the algorithm does not repeatedly recover the same geographical areas, the identified clusters, however reasonable they may look given statistical checks or other validation, have no utility. Ultimately, a criteria, defined here by the practitioner as finding the same spatial area, is the final objective. From Fig. 10, it may seem surprising that the same area is not recovered precisely after each iteration of this component of NEMI. However, despite the precision apparent in the Fig.

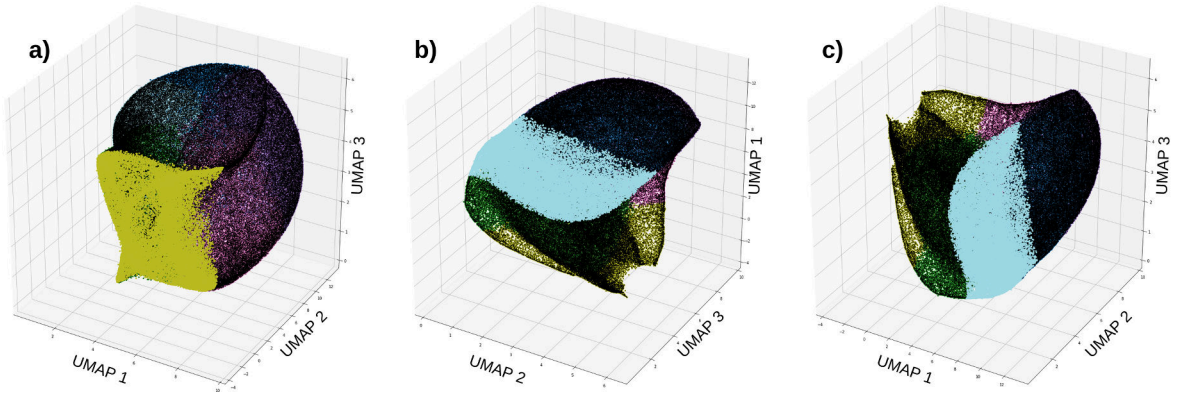

Figure 15. The k-means algorithm applied to a UMAP manifold. Panels a, b, and c show the same manifold from different angles. Here the impact of k-means is illustrated. Note how the manifold is artificially 'chopped' up in ways that clearly do not respect the data.

10, there is geographical variability. This variability, as discussed in the next section, is intrinsically useful because it allows the quantification of uncertainty.

Having an ensemble of runs of the inner workflow, where the necessary number of runs is discussed in section 3.2.2, NEMI now sorts the identified clusters for each ensemble member by spatial similarity. This is necessary because the cluster numbers are arbitrarily assigned within each ensemble member. This similarity is assessed in terms of what amount of geographical overlap there is between clusters in the different ensemble members. For this sorting, weighting by geographical extent is used because large areal extents are seen as a relevant feature to favor. However, appropriate weighting can vary depending on the research question.

The default clustering methodology in NEMI is agglomerative, and while this method does not need to be used to successfully apply NEMI, it has the benefit of allowing the selection of the number of clusters. NEMI is designed to be appropriate both for global and regional applications. Specifically, a practitioner in need of a globally representative set of clusters would select a small level of aggregation, while a regional application would choose a higher one. This feature of aggregation is used in the discussion below, but is not necessary. It is worth stressing that there is not necessarily a globally "correct" number of clusters in the case where agglomerative clustering has been used. The chosen number of clusters can depend on the application and the geographic domain of interest. This thus opens up the possibility of subclassification, where the scope of the research question helps define the level of aggregation required.

Note that it is up to the practitioner to determine a reasonable level and effective number of clusters, as well as acceptable level of uncertainty, for example when using entropy, discussed previously and further below.

The level of aggregation as well as the number of clusters is illustrated in Fig. 16. Three different ensemble members are shown separately (rows), with 6 clusters in the left column and 15 clusters in right column. Note that the three members look very similar, particularly in their global distributions. Note this and subsequent figures of spatial projections were generated with slightly different data from an equivalent MOM6 run.

3.2.2 II: Leveraging and managing noise

In Fig. 17 the product of applying NEMI to the BV data is shown, meaning that it is the result of taking the majority vote across the ensemble with the given number of clusters. An HCA cluster number of 6 (top row) and 15 (bottom row) are demonstrated. Comparing to Fig. 16, we can see that while the figures look somewhat similar, overall the clusters are more spatially coherent and crucially *reproducible*. The entropy associated with the clusters can be seen to vary (Fig. 18) from cluster to cluster, and be higher at the spatial boundaries.

The issue of noise and stochasticity within data and methods may at first appear to be a challenge that only increases the difficulty of building applications interpreting them. In this section, I will describe the final notion and step of NEMI and make a case that stochastic-friendly methods are needed for crafting methodologies applied to 'real' data. By effectively quantifying the uncertainty of a clustering result, NEMI also uses this to optimize the parameter choices, described further below.

No dataset is perfect, and methods, like most from ML, must find optimal ways of approximating the 'underlying' model. However, as demonstrated in Fig. 3, being able to account for the slight variations, for example in the sine curve in the top middle panel, can improve a model's utility. Having a methodology that can reflect the uncertainty of the model fit can be highly beneficial. The two-dimensional examples in Fig. 3 are simple cases, but highly nonlinear BV data pose a more difficult problem. In NEMI, as with any neural network application or optimization algorithm, the method application will determine the best fit given its initial conditions, for example, how internal parameters with the UMAP weights are initialized by including a stochastic method, such as, a random seed. In many cases, a slight perturbation in initial conditions can lead to a different result,

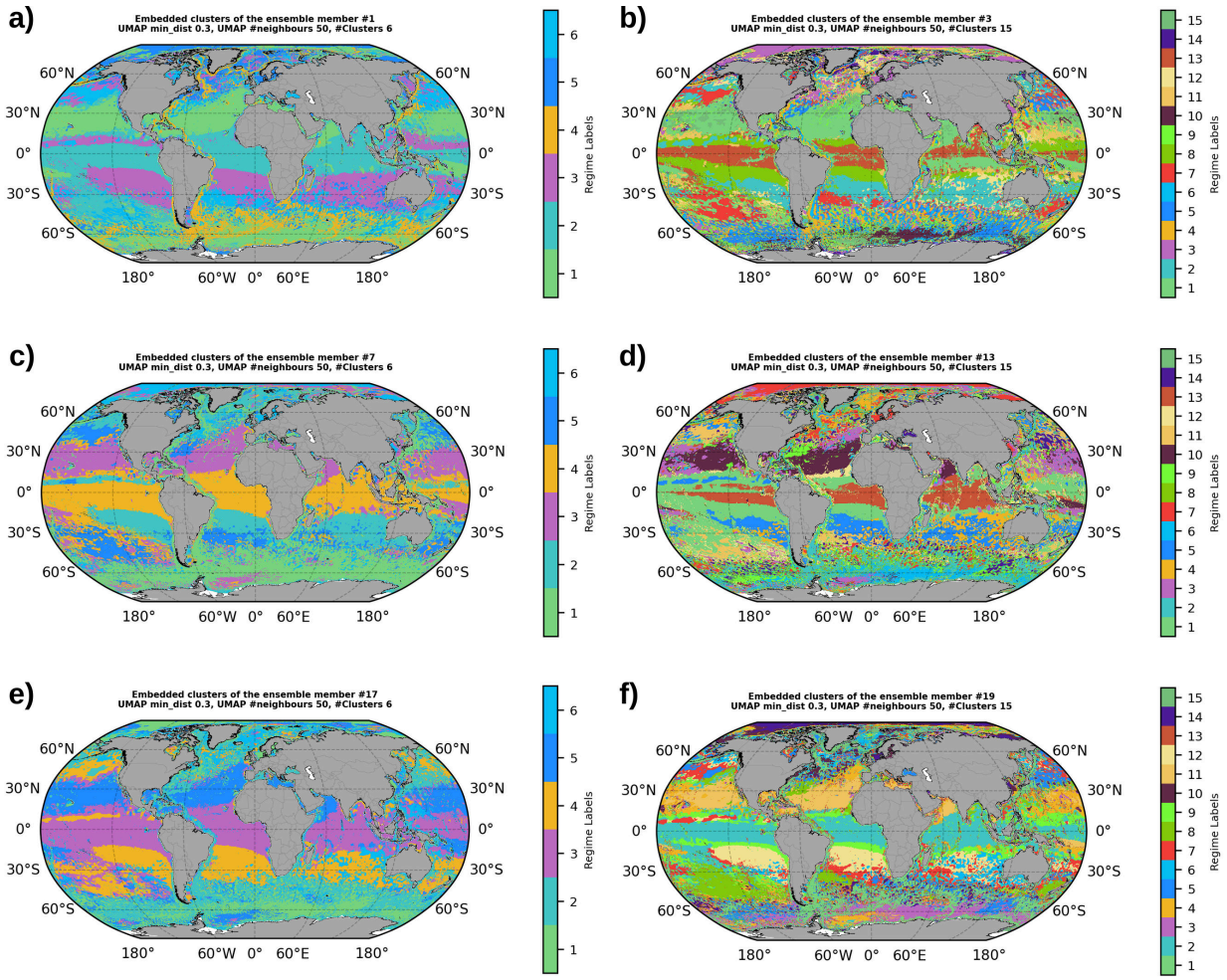

Figure 16. Demonstration of the changes in cluster locations within the ensemble. Three arbitrarily chosen different ensemble members for cluster number cases where 6 (a, c, and e) and 15 (b, d, and f), where the UMAP parameters were kept fixed. Note how there are subtle changes in locations and that the colors are arbitrarily assigned. Figure by Dr. Djeutchouang.

meaning a different model representation. In NEMI, this would be a different manifold, as was demonstrated in Fig. 10. What this sensitivity to initial conditions means in practice is that there are multiple landscapes of possible solutions that the model can converge to and that these different states can be reached given just a small difference in parameters.

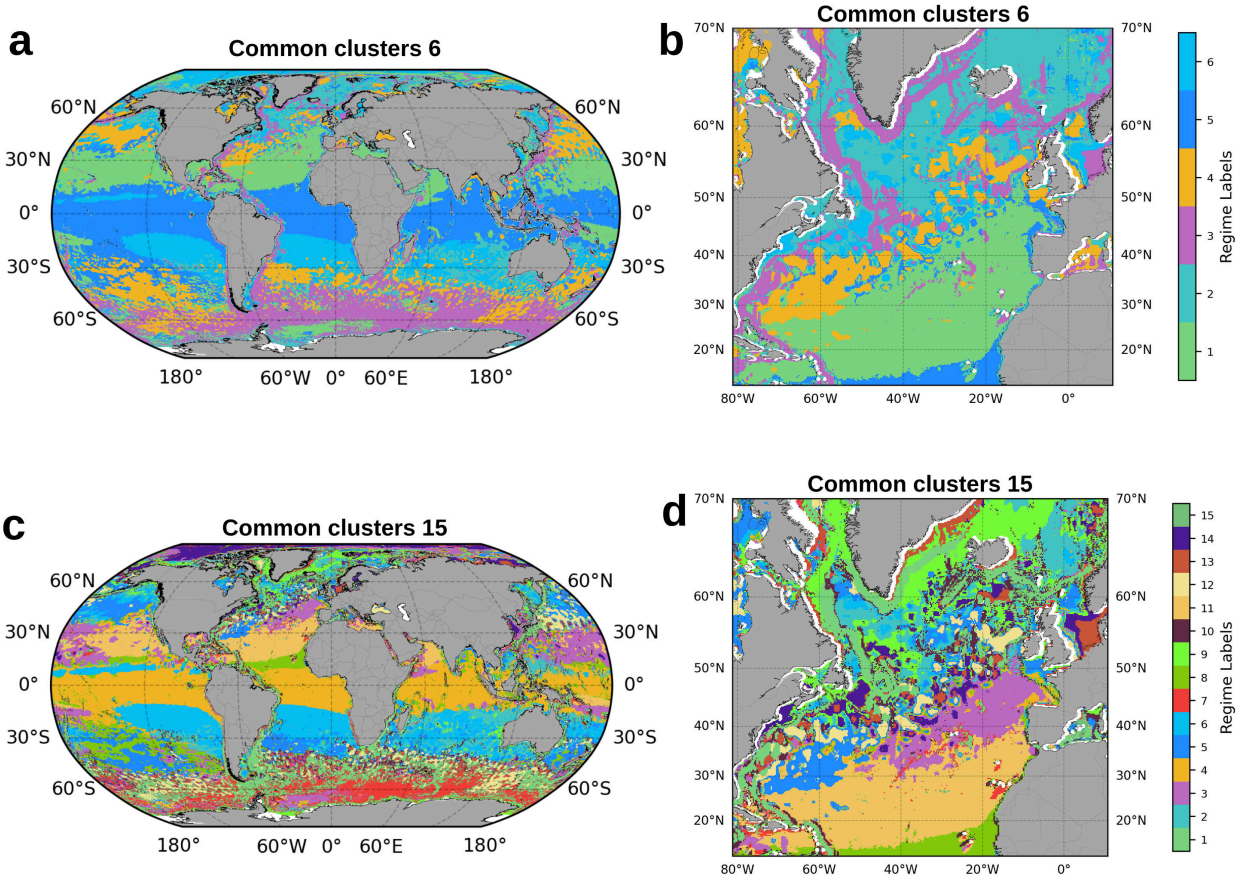

Figure 17. The final clusters of BV dynamical regimes. Cluster numbers of 6 and 15 are shown, which are the result of the majority vote across the ensemble. Note that in comparison to Fig. 16 the clusters here are much smoother. Figure by Dr. Djeutchouang.

	Description	Key Choices
Data Preprocessing & Feature Selection	Normalize and standardize input data; select relevant features based on domain knowledge (e.g., the ocean barotropic vorticity equation terms in section 2.4).	Scaling method (z-score, min-max); Feature selection criteria
Stage 1.I. Manifold Projection	Apply UMAP to reduce data dimensionality, mapping high-dimensional space to a structured 3D space that highlights latent structures: Hidden patterns and relationships between variables.	Hyperparameters `n_neighbors`, `min_dist`, `spread` (chosen based on stability tests)
Stage 1.II. Clustering on the Manifold	Perform agglomerative clustering on the UMAP-embedded space to identify the latent structures.	Clustering method (default: Agglomerative); Distance metric (default: Squared Euclidean)
Stage 2.I. Ensemble Generation (Stochastic Regularization)	Repeat Steps 2–3 multiple times with small perturbations to internal parameters to ensure robustness.	Number of ensemble members; Perturbation strategy
Stage 2.II. Cluster Consensus & Uncertainty Quantification	Align cluster labels across ensemble members using majority voting or entropy-based measures.	Consensus metric (e.g., Majority Vote, Entropy)
Stage 2.III. Domain-Specific Interpretation & Validation	Compare NEMI results against external physical expectations (e.g., known oceanographic structures); adjust parameters if necessary.	External validation criteria (e.g., expert assessment, known physical features)

Table 1. Short summary of NEMI and key choices per step referenced to oceanographic application. After preprocessing, stage 1 (I–II) corresponds to the inner manifold-learning loop, while stage 2 (I–III) corresponds to the outer ensemble-assessment loop (Section 1.1).

The sensitivity to parameters may appear to be a weakness in a methodology and will be if a model of sufficient utility is not arrived at. However, in the application of NEMI to the BV data the slight sensitivity to parameters allows the exploration of the complex covariance space of the BV data. Consequently, NEMI allows an estimation of the uncertainty, which helps us determine how robust the solution is. Using the framework of bias versus variance, having a good approximation of the variance within the covariance space of the data a methodology describes is highly beneficial. The application of a manifold methodology facilitates this. Thus, in practice, NEMI is run iteratively, and for combinations of parameters, the change in the uncertainty can be used to assess if the resulting model is a better fit to the data.

Several methods and approaches can be used to estimate the uncertainty. In the BV example, a geographic majority vote was used to determine the final clusters, and entropy was used as the metric for uncertainty. The majority vote means that for each geographical location, the cluster that was most often flagged throughout the ensemble was the one chosen. This approach allows one to quantify how many different clusters were considered at a particular location, which can be thought of at the entropy of the location as detailed in Appendix A. Fig. 18 illustrated the spatial entropy patterns, produced comparing clusters such as those in Fig. 16. Note that different areas consistently have high or low entropy, and that these can be seen to delineate clusters.

When determining the clusters using the majority vote, note that if the majority was between two different clusters, this could be important information. Especially along the geographical borders, there is often enhanced uncertainty, and this information can be very useful when using the clusters to address research questions.

The relationship between parameter tuning and ensemble generation and uncertainty quantification is important. Using NEMI, we distinguish firstly between the parameters that are user-specified and remain fixed across the ensemble, such as, `min_dist` and `n_neighbors`, and the internal parameters of UMAP that result from the optimization, i.e. the weights. The small perturbation, referred to in Stage 2.I of Table 1, is from this second category. Each ensemble uses the same parameters, but arrived at a unique set of internal parameters representing UMAP's internal weights. This leads to sightly different optimized embeddings (Fig. 10). The entropy quantification is enabled by the ensemble spread arising from these different optimized embeddings. The tuning of parameters takes place at a level above, as if the uncertainty across the ensemble is deemed too high, the practitioner adjusts the parameters given to UMAP to generate a new ensemble. One of

ensemble generation with fixed parameters but varying random initializations, and one of parameter tuning based on ensemble-level uncertainty.

3.3 III: Oceanographic interpretation of regimes

Having determined the desired number of clusters, validation via theory, or field-specific intuition should also occur. If something that is apriori assumed should be present is **not seen**, this does not necessarily invalidate the results, but certainly that the results need to be treated with increased caution, for example, by reevaluating what can be expected from the data. Some pragmatism related to the data is often prudent, and it is worth some time investment before applying NEMI to consider what one realistically can expect the data to contain.

As an example of a strategy reasonable for the BV budget data, I will use a "canonical" balance between equation terms. Certain balances, meaning that the terms add up to zero, are known and expected in certain regions, and have thus been found in other studies possibly using very different methods. Specifically here, Sverdrup balance, which is a canonical balance between the wind stress curl and the planetary vorticity advection (Munk, 1950; Sverdrup, 1947) is expected and has been extensively studied for other model systems (Sonnewald et al., 2019; Sonnewald and Lguensat, 2021; Sonnewald et al., 2023). In the example here, NEMI was applied to a realistic coupled model (Griffies et al., 2024a,b), where intuition and experience strongly suggest the Sverdrup balance should emerge, certainly in the subtropics in the Northern Hemisphere (Munk, 1950; Sverdrup, 1947). In Fig. 16 and 17, such a balance between the wind stress curl and the planetary vorticity advection can be seen in cluster 11 in the case with 15 clusters. For case with 6 clusters, cluster 1 is similar. We can see this balance increasingly take shape going from 6 to 15 regimes. The exact locations where the balance does not hold (where there are other clusters mixed in, for example, over ridges) can lead to new studies and new scientific insight, for example Sonnewald et al. (2023). As such, NEMI is an avenue for generating new knowledge with ML. However, if this were a BV balance in an idealized channel-model setup one would not necessarily flag the absence of this balance as suspicious. As a general tool, this step of NEMI requires field-specific intuition, where ML and scientists should interact to forge and identify new avenues of discovery.

To illustrate the oceanographic context of these results, I will briefly give two examples of interpretation. Note, that the number of clusters is entirely up to the practitioner and will likely depend on the research question at hand. First, in Fig. 17a and c, we can see that clusters have been

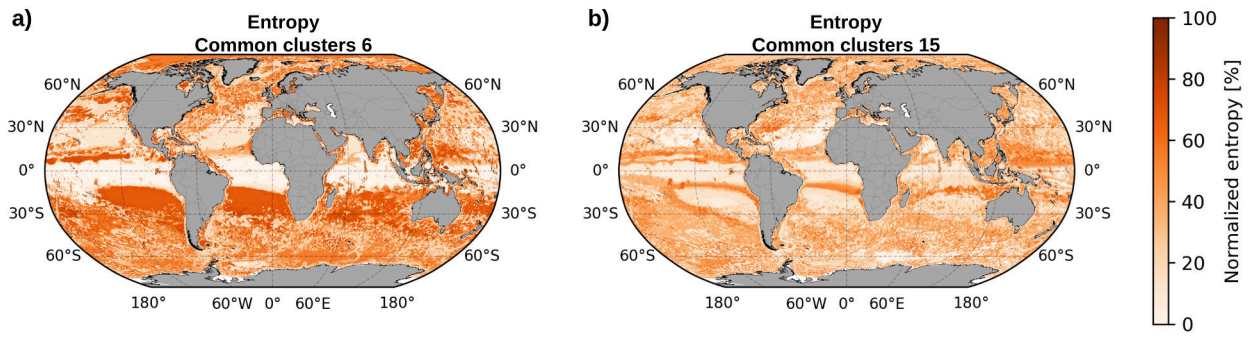

Figure 18. Illustration of the entropy for individual BV regimes. Entropy is represented as a probability, where low entropy indicates low uncertainty. Figure by Dr. Djeutchouang.

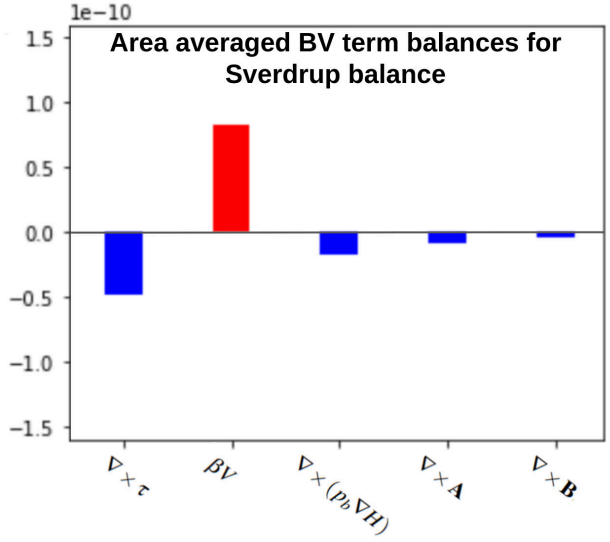

Figure 19. Figure showing canonical and expected balance for validation via expert judgment. The expected balance (meaning that the terms add up to zero) between the wind stress curl and planetary vorticity advection. This is found in cluster 11 for the case with 15 clusters, and a similar balance is present in cluster 1 in the case eith 6 clusters.

identified in regions where the wind stress curl is largely homogeneous, such as in the major ocean gyres. The major gyres have been grouped together, with separate clusters for the southern and northern hemispheres in both the cases with 6 and 15 clusters. The symmetry around the equator, and thus the change of sign in the vorticity, makes this intuitive, as we know from Fig. 19 and from oceanographic intuition that these areas should be similar but have opposite signs in their main drivers. Second, I will use an example where increasing the number of clusters is interesting. Fig. 17a and d show the clusters in the North Atlantic. For the case with 6 clusters, there is one cluster that traces the western side of the basin (purple, cluster 3), Greenland and areas around Iceland, for example. In the case with 15 clusters, this area has been separated into a large variety of clusters. Specifically for the areas that in the 15 cluster case have been broken up, this could indicate that interactions with bathymetry could cause alternating patterns where terms, for example, are similar in magnitude but with alternating sign before and after bathymetry.

4 CONCLUSION AND OUTLOOK

The statistician George Box is attributed with the phrase "all models are wrong, some are useful". This phrase is appropriate for all ML models of the natural world, and the extent to which the ML model can be validated is proportional to how useful the model is. Here, I presented the method Native Emergent Manifold Interrogation (NEMI), which is a methodology for arriving at a statistical/ML model of a given dataset, that is aimed at offering an improvement over the status quo. NEMI offers an improvement over current methodologies in that it is able to accommodate highly complex data, in a manner that many currently mainstream methods cannot. A key strength is that it allows intuitive validation, a core shortcoming of many ML applications. NEMI is designed to quantify uncertainty and thus can better accommodate the inherently noisy and incomplete data that is available in most geoscience applications.

NEMI is a generalization of the methodology presented in Sonnewald et al. (2020), and is designed to find underlying patterns within data. An explicitly hierarchical approach is used in the default, making NEMI less parametric (fewer parameters to tune and less danger of noise interference) and intuitively useful both for global (for example the whole Earth in the present example) or more local applications (for example a basin or more regional assessment). NEMI does not use fixed field-specific benchmark criteria (used in Sonnewald et al. (2020)) but is generalized so a field-agnostic option is available. Lastly, NEMI invites the use of a range of uncertainty

quantification options in the final cluster evaluation, from a majority vote to entropy. I demonstrate NEMI's application to data from a numerical ocean model, namely the time-mean barotropic vorticity balance of the global ocean circulation from MOM6. The data serves as an example of a highly nonlinear and complicated covariance structure, within which reside highly valuable oceanographic patterns. NEMI is used to extract these patterns and facilitate further scientific discovery. However, NEMI is entirely general and can be used on a range of data from the earth sciences and beyond.

The code base for NEMI is being actively developed as of writing, with focus areas including scaling over various HPC systems, embedding alignment, and cluster overlap assessment. I invite readers to contribute to NEMI, as the core motivation behind development is utility for wide varieties of use cases.

AVAILABILITY STATEMENT

The code for the Native Emergent Manifold Interrogation (NEMI) method is available here: https://github.com/maikejulie/NEMI. DOI: 10.5281/zenodo.7764719. The BV data can be found here: Hemant Khatri, Stephen M. Griffies, Benjamin A. Storer, Michele Buzzicotti, Hussein Aluie, Maike Sonnewald, Raphael Dussin, & Andrew Shao. (2023). Barotropic vorticity budget analysis in a global ocean simulation [Data set]. DOI: https://doi.org/10.5281/zenodo.10078539.

ACKNOWLEDGMENTS

I gratefully acknowledge students and researchers who provided the inspiration for this manuscript, including Yvonne Jennings, Will Yik, Dr. Arijeet Dutta, Dr. Jinfei Wang, Makayla McDevitt, Dr. Laique Merlin Djeutchouang and Tinayang Dou. The narrative and content, particularly of the introductory material, was shaped by many conversions with students and colleagues who wanted to use ML in their research and those who requested details regarding NEMI.

Funding: Cooperative Institute for Modeling the Earth System, Princeton University, under Award NA18OAR4320123, and seperately award NA24OARX431C0058-T1-01 from the National Oceanic and Atmospheric Administration, U.S. Department of Commerce. The statements, findings, conclusions, and recommendations are those of the authors and do not necessarily reflect the views of Princeton University, the National Oceanic and Atmospheric Administration, or the U.S.

Department of Commerce.

APPENDIX

A: Entropy for uncertainty estimation

Entropy (H) can be used as a measure of uncertainty. As discussed in Clare et al. (2022): In information theory, entropy is the expected information of a random variable, and for each sample i is given by

$$H_i = -\sum_{j=1}^{N_l} p_{ij} \log(p_{ij}), \tag{8}$$

here N_l is the number of possible outcomes for each location and p_{ij} is the probability of each outcome j for sample i (Goodfellow et al., 2016). The larger the entropy, the less skewed the distribution will be and the more uncertain the outcome. The concept of entropy can be directly applied to manage the potentially different results from NEMI for each geographic location within the ensemble. Wether this is better than a simpler method, such as a majority vote, depends entirely on the application.

REFERENCES

Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J. M., and Perona, I. (2013). An extensive comparative study of cluster validity indices. *Pattern Recognition*, 46(1):243–256.

Beucler, T., Ebert-Uphoff, I., Rasp, S., Pritchard, M. S., and Gentine, P. (2021). Machine learning for clouds and climate (invited chapter for the agu geophysical monograph series "clouds and climate").

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

Bracco, A., Brajard, J., Dijkstra, H., Hassanzadeh, P., Lessig, C., and Monteleoni, C. (2024). Machine learning for the physics of climate. *Nature Reviews Physics*, 7.

Clare, M. C., Sonnewald, M., Lguensat, R., and Deshayes, J. (2022). Explainable artificial intelligence for bayesian neural networks: toward trustworthy predictions of ocean dynamics. *Journal of Advances in Modeling Earth Systems*, 14(11):e2022MS003162.

Davis, M. (2001). *Late Victorian Holocausts: El Niño Famines and the Making of the Third World*. Verso.

Dramsch, J. S. (2020). Chapter one - 70 years of machine learning in geoscience in review. In Moseley, B. and Krischer, L., editors, *Machine Learning in Geosciences*, volume 61 of *Advances in Geophysics*, pages 1–55. Elsevier.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise.

Fleming, S., Watson, J., Ellenson, A., Cannon, A., and Vesselinov, V. (2021). Machine learning in earth and environmental science requires education and research policy reforms. *Nature Geoscience*, 14.

Furtado, J. C., Molina, M. J., Arcodia, M. C., Anderson, W., Beucler, T., Callahan, J. A., Ciasto, L. M., Gensini, V. A., L'Heureux, M., Pegion, K., Pérez-Carrasquilla, J. S., Sonnewald, M., Takahashi, K., Xiang, B., and Zimmerman, B. G. (2025). Taking the garbage out of data-driven prediction across climate timescales.

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

Griffies, S. M., Adcroft, A., Beadling, R. L., Bushuk, M., Chang, C.-Y., Drake, H. F., Dussin, R., Hallberg, R. W., Hurlin, W., Khatri, H., Krasting, J. P., Lobo, M., MacGilchrist, G., Reichl, B. G., Sane, A., Sergienko, O. V., Sonnewald, M., Steinberg, J. M., Tesdal, J.-E., Thomas, M. D., Turner, K. E., Ward, M. L., Winton, M., Zadeh, N., Zanna, L., Zhang, R., Zhang, W., and Zhao, M. (2024a). The gfdl-cm4x climate model hierarchy, part i: model description and thermal properties.

Griffies, S. M., Adcroft, A., Beadling, R. L., Bushuk, M., Chang, C.-Y., Drake, H. F., Dussin, R., Hallberg, R. W., Hurlin, W., Khatri, H., Krasting, J. P., Lobo, M., MacGilchrist, G., Reichl, B. G., Sane, A., Sergienko, O. V., Sonnewald, M., Steinberg, J. M., Tesdal, J.-E., Thomas, M. D., Turner, K. E., Ward, M. L., Winton, M., Zadeh, N., Zanna, L., Zhang, R., Zhang, W., and Zhao, M. (2024b). The gfdl-cm4x climate model hierarchy, part ii: case studies.

Harvey, A. C. (1982). Spectral analysis and time series, m. b. priestly. two volumes, 890 pages plus preface, indexes, references and appendices, london: Academic press, 1981. price in the uk: Vol. i, £49-60: Vol. ii, £20-60. *Journal of Forecasting*, 1(4):422–423.

Hughes, C. W. and de Cuevas, B. A. (2001). Why Western Boundary Currents in Realistic Oceans are Inviscid: A Link between Form Stress and Bottom Pressure Torques. *Journal of Physical Oceanography*, 31(10):2871–2885.

Jenniges, Y., Sonnewald, M., Maneth, S., Olsen, A., and Koch, B. P. (2025). Unveiling 3d ocean biogeochemical provinces in the north atlantic: A systematic comparison and validation of clustering methods.

Kaiser, B. E., Saenz, J. A., Sonnewald, M., and Livescu, D. (2022). Automated identification of dominant physical processes. *Engineering Applications of Artificial Intelligence*, 116:105496.

Kaur, J., Parmar, K., and Singh, S. (2023). Autoregressive models in environmental forecasting time series: a theoretical and application review. *Environmental Science and Pollution Research*, 30:19617–19641.

Khatri, H., Griffies, S. M., Storer, B. A., Buzzicotti, M., Aluie, H., Sonnewald, M., Dussin, R., and Shao, A. (2024). A scale-dependent analysis of the barotropic vorticity budget in a global ocean simulation. *Journal of Advances in Modeling Earth Systems*, 16(6):e2023MS003813.

Kohonen, T. (2004). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69.

Lai, C.-Y., Hassanzadeh, P., Sheshadri, A., Sonnewald, M., Ferrari, R., and Balaji, V. (2025). Machine learning for climate physics and simulations. *Annual Review of Condensed Matter Physics*, 16(Volume 16, 2025):343–365.

Lorenz, E. (1956). *Empirical Orthogonal Functions and Statistical Weather Prediction*. Scientific report. Massachusetts Institute of Technology, Department of Meteorology.

Mackay, D. J. C. (2003). *Information Theory, Inference and Learning Algorithms*.

MacQueen, J. (1965). Some methods for classification and analysis of multivariate observations. In *Proc. 5th Berkeley Symposium on Math., Stat., and Prob*, page 281.

Mann, M. E., Bradley, R. S., and Hughes, M. K. (1999). Northern hemisphere temperatures during the past millennium: Inferences, uncertainties, and limitations. *Geophysical Research Letters*, 26(6):759–762.

McInnes, L., Healy, J., Saul, N., and Grossberger, L. (2018). Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3:861.

Munk, W. H. (1950). ON THE WIND-DRIVEN OCEAN CIRCULATION. *Journal of Meteorology*, 7(2):80–93.

Nanga, S., Bawah, A. T., Acquaye, B., Billa, M.-I., Baeta, F., Odai, N., Obeng, S. K., and Nsiah, A. D. (2021). Review of dimension reduction methods. *Journal of Data Analysis and Information Processing*.

Schlake, G. S. and Beecks, C. (2024). Validating arbitrary shaped clusters - a survey. In *2024 IEEE 11th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–12.

Sonnewald, M., Dutkiewicz, S., Hill, C., and Forget, G. (2020). Elucidating ecological complexity: Unsupervised learning determines global marine eco-provinces. *Science advances*, 6(22):eaay4740.

Sonnewald, M. and Lguensat, R. (2021). Revealing the impact of global heating on north atlantic circulation using transparent machine learning. *Journal of Advances in Modeling Earth Systems*, 13(8):e2021MS002496. e2021MS002496 2021MS002496.

Sonnewald, M., Lguensat, R., Jones, D., Düben, P., Brajard, J., and Balaji, V. (2021). Bridging observations, theory and numerical simulation of the ocean using machine learning. *Environmental Research Letters*, 16.

Sonnewald, M., Reeve, K. A., and Lguensat, R. (2023). A southern ocean supergyre as a unifying dynamical framework identified by physics-informed machine learning. *Communications Earth & Environment*, 4(1):153.

Sonnewald, M., Wunsch, C., and Heimbach, P. (2018). Linear predictability: A sea surface height case study. *Journal of Climate*, 31:2599–2611.

Sonnewald, M., Wunsch, C., and Heimbach, P. (2019). Unsupervised learning reveals geography of global ocean dynamical regions. *Earth and Space Science*, 6(5):784–794.

Stommel, H. (1948). The westward intensification of wind-driven ocean currents. *Transactions, American Geophysical Union*, 29(2):202–206.

Sun, Z., Ten Brink, T., Carande, W., Koren, G., Cristea, N., Jorgenson, C., Janga, B., Asamani, G., Achan, S., Mahoney, M., Huang, Q., Mehrabian, A., Munasinghe, T., Liu, Z., Margolis, A., Webley, P., Gong, B., Rao, Y., Burgess, A., and Duzgun, S. (2024). Towards practical artificial intelligence in earth sciences. *Computational Geosciences*, 28:1305–1329.

Sverdrup, H. U. (1947). Wind-driven currents in a baroclinic ocean; with application to the equatorial currents of the eastern pacific. *Proceedings of the National Academy of Sciences*, 33(11):318–326.

van der Maaten, L. and Hinton, G. (2008). Viualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605.

Walker, G. (1928). World weather. vol. 54. *QJR Meteorol Soc*, pages 79–87.

Ward, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American*

Statistical Association, 58:236–244.

Yang, Y. (2005). Can the strengths of AIC and BIC be shared? A conflict between model indentification and regression estimation. *Biometrika*, 92(4):937–950.

Zebiak, S. E. and Cane, M. A. (1987). A model el niño–southern oscillation. *Monthly Weather Review*, 115(10):2262 – 2278.