Nonlinear Dynamics manuscript No.
(will be inserted by the editor)

What are the most informative data points for predicting extreme events?

Likelihood-weighted data selection to improve the prediction of extreme events in complex dynamical systems

Bianca Champenois (Corresponding Author) · Themistoklis P. Sapsis

Received: date / Accepted: date

Abstract The growing availability of large datasets that describe complex dynamical systems, such as climate models and turbulence simulations, has made machine learning an increasingly popular tool for modeling and analysis, but the inherent low representation of extreme events poses a major challenge for model accuracy in the tails of the distribution. This raises a fundamental question: Given a large dataset, which data points should we use to train machine learning models that effectively learn extremes? To address this question, we study a likelihood-weighted active data selection framework that identifies the most informative data points for model training. The framework improves predictions of extreme values of a target observable, scales to high-dimensional systems, and is model-agnostic. Unlike traditional active learning, which assumes the ability to query new data, our method is designed for problems where the dataset is fixed but vast, focusing on selection rather than acquisition. Points are scored using a likelihood-weighted uncertainty sampling criterion that prioritizes samples expected to reduce model uncertainty and improve predictions in the tails of the distribution for systems with non-Gaussian statistics. When applied to a machine learning climate model with input dimensionality on the order of tens of thousands, we find that the likelihood-weighted active data selection algorithm most accurately captures the statistics of extreme events using only a fraction of the original dataset. We also introduce analysis techniques to further interpret the optimally selected points. Looking ahead, the approach can serve as a compression algorithm that preserves information associated with extreme events in vast datasets.

Keywords active learning · active data selection · extreme events · complex dynamical systems · climate modeling · machine learning

Mathematics Subject Classification (2020) MSC 86A08 · MSC 86-08 · MSC 62G32 · MSC 60G70 · MSC 62L05

B. Champenois (Corresponding Author)
Massachusetts Institute of Technology
E-mail: bchamp@mit.edu

T. P. Sapsis
Massachusetts Institute of Technology

1 Introduction

Many important scientific and engineering problems involve complex dynamical systems, such as turbulent flows or the climate. Often in such settings, we now have access to terabytes or even petabytes of simulation, observational, or experimental data. This abundance of data, coupled with the complexity of the underlying physics, has made machine learning (ML) an increasingly popular tool for modeling and analysis of dynamical systems [10, 60, 26, 37]. However, extreme events, which are high-impact events that lie in the tails of the probability distribution (e.g., rogue waves or extreme weather), are typically underrepresented in datasets not explicitly designed to capture extremes [1, 23, 73]. Standard ML models trained on available datasets tend to prioritize the regions of the domain where most points exist. As a result, they often fail to capture extremes or converge slowly, resulting in poor generalization where accurate predictions matter most [56]. From this perspective, not all points in a given dataset carry the same *value* of information, so using the full dataset or a random subset of the dataset without proper selection can be inefficient or ineffective for training [33, 86, 68]. Therefore, identifying a subset of training data that is most informative for predicting extreme events can improve the model's ability to capture the full statistics while reducing computational cost of training on large datasets.

To address the challenges of training on large or randomly sampled datasets, active learning and data selection methods have emerged as effective approaches to efficiently choosing training data by iteratively identifying points that maximize information gain or reduce model uncertainty [47, 17, 78]. These methods are valuable in scientific applications where data acquisition or simulation is expensive or time-consuming, and in settings with vast datasets requiring subsampling. Among these methods, likelihood-weighted sampling, which emphasizes samples that balance high uncertainty with a high likelihood of extreme outcomes, has been identified as particularly effective for capturing the tails of the distribution [52, 72, 74, 7, 8]. Likelihood-weighted active learning has been successfully applied for modeling prototypical nonlinear systems [72, 74], hydrological systems [8], pandemics [57], and offshore structures [52, 57, 31, 38].

Despite their promise, likelihood-weighted criteria have mainly been applied in active learning settings, where data can be queried on demand, rather than in active data selection, where datasets are fixed yet vast. Previous work has explored the potential of likelihood-weighted criteria for quantifying the value of individual data points within a given dataset in a posterior manner (i.e., after training the model using the full dataset) [56], but did not formulate data selection algorithms focused on improving training to better capture extreme events with less data. Furthermore, the application of likelihood-weighted active learning and active selection methods to extremely high-dimensional systems — such as those with input dimensions on the order of tens of thousands or more — remains largely understudied.

Here, we explore these directions by formalizing a likelihood-weighted active data selection framework for large, fixed datasets obtained from systems with non-Gaussian statistics [73]. Additionally, we address the challenge of high-dimensional inputs to make the framework practical for real-world applications. Overall, the framework (i) prioritizes the most informative points in large datasets, (ii) enables efficient data compression, (iii) reduces training costs, (iv) improves model

generalization for extreme events, and (v) allows for the interpretation of the selected points. A key advantage of the approach is that it is model-agnostic and adaptable to any ML architecture. Although the examples in this paper focus on spatiotemporal datasets, the method is broadly applicable to other systems that exhibit extreme events. However, the current formulation is not designed to identify precursors to extreme events, but rather to detect the events themselves, though the framework could be modified to target precursors [15, 69]. We demonstrate the utility of the method through an application to climate modeling, where it is used to improve the predictions of extreme weather events in a system with dimensionality on the order of 10^5, using a fraction of the full dataset.

The structure of the paper is as follows. In Section 2, we introduce the active data selection algorithm and the likelihood-weighted sampling criterion. In Section 3, we explain how to quantify the uncertainty needed to evaluate this selection criterion. In Section 4, we describe how to extend the method to systems for which the inputs are high-dimensional functionals. In Section 5, we apply the framework to predict extreme events in the Majda–McLaughlin–Tabak (MMT) model, a one-dimensional model for dispersive wave turbulence. Finally, in Section 6, we apply the framework to a real-world problem of learning a correction operator for the outputs of a coarse resolution climate model, in which accurately capturing extreme weather events is critical. In both applications, we introduce methods to interpret the selected points and gain insight into the data selection algorithm.

2 Likelihood-Weighted Data Selection

2.1 Supervised Machine Learning for Complex Dynamical Systems

We consider the setting of supervised ML in which a model $\mathcal{M}_\theta : \mathcal{X} \to \mathcal{Y}$ maps input features $x \in \mathcal{X}$ to targets $y \in \mathcal{Y}$. The model architecture (e.g., neural network, Gaussian process) defines the functional form of $\mathcal{M}_\theta$, parameterized by θ, which is trained to minimize a loss function $\mathcal{L}$ over a *training set* of size P

$$\mathcal{D}_{\text{train}} = \{(x_j, y_j)\}_{j=1}^{P}. \tag{1}$$

The goal is to find parameters θ^* that minimize the loss:

$$\theta^* = \arg\min_\theta \sum_{j=1}^{P} \mathcal{L}\left(\mathcal{M}_\theta(x_j), y_j\right). \tag{2}$$

In the context of complex dynamical systems, Figure 1 highlights the setting

$$\mathcal{M}_\theta : \mathcal{X} \to \mathcal{U} \to \mathcal{Y} \tag{3}$$

where $\mathcal{X}$ is a high-dimensional input field, $\mathcal{U}$ is a field that is a function of $\mathcal{X}$, and $\mathcal{Y}$ is a scalar observable extracted from $\mathcal{U}$. The ML model can be configured to predict $\mathcal{U}$ or $\mathcal{Y}$ depending on the task. To illustrate, in the climate modeling application (Section 6), the input $\mathcal{X}$ is data from a coarse-resolution climate model. The model $\mathcal{M}_\theta$ is a debiasing operator (correction operator) trained to map the coarse-resolution model outputs $\mathcal{X}$ to their debiased counterpart $\mathcal{U}$. The scalar observable $\mathcal{Y}$ is then extracted from the corrected field $\mathcal{U}$, such as, for example, maximum temperature over space or temperature at a specific location.

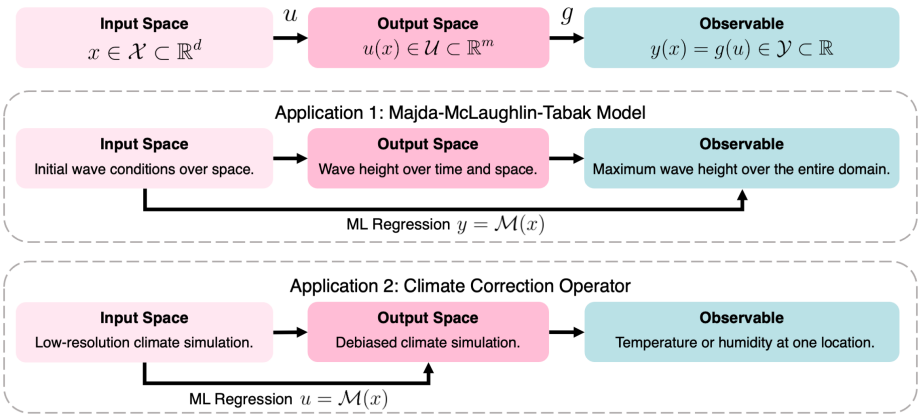

Fig. 1 Schematic of the mapping from input space to observable. Input fields $x \in \mathcal{X}$ are mapped to output fields $u \in \mathcal{U}$ through a forward model (ML model or physics-based numerical model), and a scalar observable $y \in \mathcal{Y}, \mathcal{Y} \subset \mathbb{R}$ is extracted from u via an observable-specific operator $g(u)$. This framework is illustrated for two applications: in the MMT case (Section 5), the inputs are the initial conditions of the system, and the observable is the maximum wave height over a specified time horizon; in the climate correction case (Section 6), the input is data from a coarse-resolution climate simulation, he model $\mathcal{M}_\theta$ is a debiasing operator (correction operator) trained to map the coarse-resolution model outputs $\mathcal{X}$ to their corresponding, debiased counterpart $\mathcal{U}$, and the observable is a scalar quantity of interest such as temperature or humidity at a specific location.

2.2 Active Data Selection

Active data selection is a method for choosing training points for supervised ML from a fixed, labeled dataset to improve model performance. In this setting, the labels y_j corresponding to inputs x_j are already known for a predefined *candidate set* of size M [61, 3, 29]:

$$\mathcal{D}_{\text{cand}} = \{(x_j, y_j)\}_{j=1}^{M}. \tag{4}$$

A selection criterion (or acquisition function, in the language of active learning) is used to sequentially identify a subset of informative samples $\mathcal{D}_{\text{select}} \subset \mathcal{D}_{\text{cand}}$, which are then added to the training set [47]:

$$\mathcal{D}_{\text{train}} \leftarrow \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{select}}. \tag{5}$$

This approach is also referred to as greedy approximation, active sampling, optimal sampling, and active search; here, we use the term *active data selection*. In the next section, we introduce *likelihood-weighted* data selection, a variant that prioritizes points expected to improve model performance in the tails of the output distribution.

Active data selection differs from active learning, where labels are not initially known. In active learning, the algorithm sequentially selects new input points x_t according to the acquisition function, queries their corresponding labels y_t through experiment or simulation, and adds the new labeled pairs (x_t, y_t) to the training set [47, 17, 28, 78]. Likelihood-weighted strategies have been widely applied in this setting to improve predictions of extreme events. Active learning belongs to a broader family of sequential data acquisition strategies, including optimal

experimental design, Bayesian experimental design, and Bayesian optimization, which also aim to identify the most informative data points [70, 13, 28, 30, 38].

In the context of extreme events, we evaluate the data selection algorithms based on how well the resulting models, trained on the selected data, capture the tails of the target observable's distribution. Specifically, we consider the log-pdf error (LPE), which measures the integrated difference between the log of the true probability density function (pdf) obtained from the true y and the log of the conditional pdf from the model's posterior mean $\hat{y}$.

$$\mathrm{LPE} = \int |\log p_y(y) - \log p_{\hat{y}}(y)| \mathrm{d}y. \tag{6}$$

This loss function is similar to the Kullback-Leibler (KL) divergence, but unlike KL, is not weighted by the output distribution p_y, so extreme events contribute more, and errors in the tails are penalized more heavily. In cases where only the left or right tail is of interest, we adapt the integration limits to target that side of the distribution. We benchmark against a Monte Carlo (MC) selection criterion, which randomly chooses points from the candidate set, providing a relevant baseline given the common use of random data splitting in ML.

2.3 Algorithm Overview

We start with a dataset of candidate points, denoted $\mathcal{D}_{\text{cand}} = \{(x_j, y_j)\}_{j=1}^{M}$, consisting of all available labeled input-output pairs from which training points can be drawn. We initialize the data selection algorithm (illustrated in Figure 2) with a small training set of size $P^{(0)}$ randomly selected points from $\mathcal{D}_{\text{cand}}$,

$$\mathcal{D}_{\text{train}}^{(0)} = \{(x_j, y_j)\}_{j=1}^{P^{(0)}}, \tag{7}$$

where (0) corresponds to the initial iteration and (t) corresponds to the t^{th} iteration. At iteration $t = 0$, the remaining points form the candidate set for the first iteration of the algorithm

$$\mathcal{D}_{\text{cand}}^{(0)} = \mathcal{D}_{\text{cand}} \setminus \mathcal{D}_{\text{train}}^{(0)}. \tag{8}$$

At each iteration t, a model $\mathcal{M}^{(t)}$ is trained on the current training set $\mathcal{D}_{\text{train}}^{(t)}$. The selection criterion $q(x)$ is then evaluated on all points in the current candidate set $\mathcal{D}_{\text{cand}}^{(t)}$, using the predictive mean and epistemic uncertainty of the target observable estimated by the probabilistic model $\mathcal{M}^{(t)}$. The point (or batch of points) that yields the highest value according to the selection criterion $q(x)$ (acquisition function) is selected and added to the training set of size $P^{(t)}$:

$$\mathcal{B}^{(t)} = \{(x_j, y_j)\}_{j=1}^{b} = \left\{ (x_j, \mathcal{M}^{(t)}(x_j)) \;\middle|\; x_j \in \underset{x \text{ (top } b)}{\arg\max}\, q(x) \right\} \tag{9}$$

$$\mathcal{D}_{\text{train}}^{(t+1)} = \mathcal{D}_{\text{train}}^{(t)} \cup \mathcal{B}^{(t)}, \quad \mathcal{D}_{\text{cand}}^{(t+1)} = \mathcal{D}_{\text{cand}}^{(t)} \setminus \mathcal{B}^{(t)}. \tag{10}$$

Here, $\mathcal{B}^{(t)} \subset \mathcal{D}_{\text{cand}}^{(t)}$ denotes the selected batch, which may contain a single point or multiple points depending on the batch size b. Batching reduces the frequency of

retraining the model, which lowers computational cost. Although this may slightly reduce optimality in selection, previous work shows that it performs comparably for moderate batch sizes [57].

The loop is repeated — retraining the model, evaluating the criterion, and updating the selected dataset — until the model error converges or falls below a predefined threshold. Through the described process, the selected data points, optimized for capturing extreme events, are chosen according to both the observable and the model architecture.

The final output of the algorithm is a high-value subset of the full dataset, together with a model trained using this subset. The dataset is selected to optimize model performance, with particular emphasis on extremes. Moreover, the points selected at each iteration can be analyzed to identify the most informative regions of the input space, and the final subset of optimal points can serve as a compressed representation of the full dataset, retaining the information relevant to extremes.

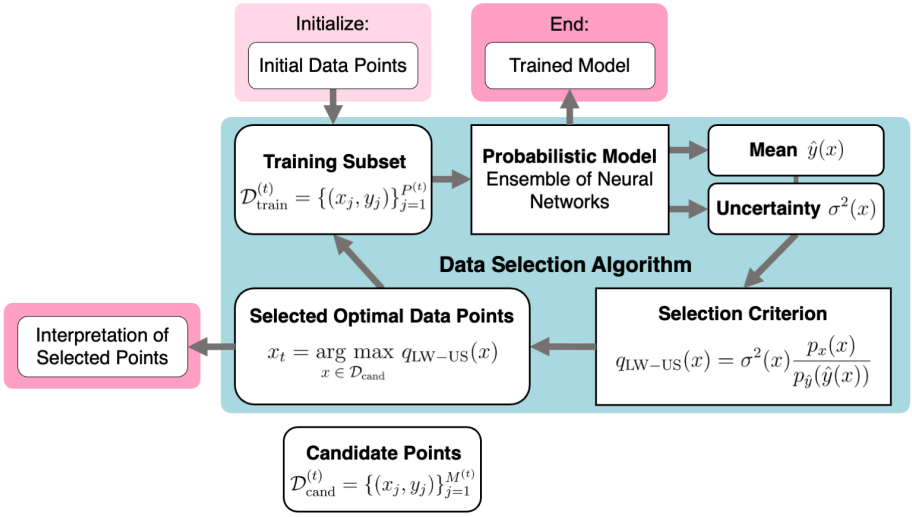

Fig. 2 Data selection algorithm. Points are sequentially selected from the candidate dataset according to the selection criterion and added to the training set to improve model prediction. The output of the algorithm is a model that has been trained on an optimal subset of the data with respect to predicting the target observable's statistics. At each iteration of the algorithm, the selected points can be interpreted to gain insights into which points are more valuable.

2.4 Selection Criterion

A key element of the data selection algorithm is the selection criterion, which identifies the most valuable points for model training. The choice of the selection criterion depends on the nature of the system (e.g., non-Gaussian, high-dimensional), the goal of the modeling problem (e.g., optimization, extreme event identification), and other constraints (e.g., computational costs). In general, the selection criterion should strike a balance between exploration and exploitation. A common choice

for the selection criterion is uncertainty sampling (US) where points are ranked by their epistemic variance [47, 55, 42].

$$q_{\mathrm{US}}(x) = \sigma^2(x). \tag{11}$$

This choice is well-motivated when using mean squared error (MSE) as a performance metric, as reducing the predictive variance directly lowers the expected MSE across the input space. However, this criterion does not take into account the magnitude of the output and, therefore, fails to account for the importance of accurately representing extreme events.

For problems involving extreme events, we use a likelihood-weighted criterion to select optimal training points and quantify the value of individual points in the dataset [52, 72]. The likelihood-weighted selection criterion, like US, targets input points that reduce the model uncertainty, but it also prioritizes points expected to produce extreme outputs. Sampling criteria that incorporate information about the output distribution were first introduced by Mohamad and Sapsis [52] and further improved in Sapsis [72] and Sapsis and Blanchard [74]. In the original formulation (from Mohamad and Sapsis [52]), the criterion compares the distribution of the model output with and without a hypothetical new sample to guide data acquisition. Specifically, the criterion aims to minimize the integrated absolute difference between the logarithms of the estimated output pdf, $p_y(y)$, and a perturbed version $p_y^+(y \mid x)$ that accounts for the effect of adding a new sample corresponding to input x:

$$q(x) = \int_{S_y} \left| \log p_y(y) - \log p_y^+(y \mid x) \right| \, dy. \tag{12}$$

Here, $p_y(y)$ is the estimated output pdf based on the current samples, $p_y^+(y \mid x)$ is the output pdf after hypothetically adding sample x, and S_y is the support of the output variable y. Maximizing $q(x)$ selects the input x that most improves accuracy in low-probability regions of the output space. For bounded S_y, this approach asymptotically converges to the true output statistics, even in regions with low probability of occurrence [74]. However, Equation 12 is computationally expensive, and its lack of smooth gradients makes it unsuitable for gradient-based optimization, which is needed in active learning settings.

Sapsis [72] derived an upper bound (which was shown to be optimal for Gaussian process regression in Sapsis and Blanchard [74]) that has lower computational cost and is analytically differentiable.

$$q(x) \leq \kappa \int \sigma^2(z; x) \frac{p_x(z)}{p_y(y(z))} \, dz, \tag{13}$$

where $\sigma^2(z; x)$ is the predictive variance at input z after including a sample at x, p_x is the input probability density, κ is a scaling coefficient, and the weighting factor $1/p_y$ emphasizes extreme outputs. This criterion quantifies the expected global reduction in uncertainty, weighted by the likelihood of extreme outputs.

Building on this, Blanchard and Sapsis [8] proposed a simplified *likelihood-weighted uncertainty sampling* (LW-US) criterion that avoids the integral by evaluating the uncertainty and weights locally at the candidate point:

$$q_{\text{LW-US}}(x) = \sigma^2(x) \frac{p_x(x)}{p_y(y(x))}. \tag{14}$$

This selection criterion balances (i) the epistemic variance σ^2, which quantifies uncertainty reduction by favoring points where the model is least confident; (ii) the input probability p_x, which emphasizes points likely to occur by assigning higher weight to inputs that are representative of the underlying distribution; and (iii) the inverse output probability $1/p_y$, which emphasizes points leading to extreme outcomes, since low-probability events in the tails of the distribution have large inverse values and are thus prioritized. This balance ensures that, among likely inputs, the criterion promotes those with a disproportionate effect on the output, and, among equally impactful inputs, it favors those that are more representative of the underlying input distribution. Together, these terms guide the exploration-exploitation trade-off, prioritizing candidate points that reduce uncertainty, are likely to occur, and most importantly, are likely to result in extreme events. When used for model training, these points produce models that are better able to represent the tails of the distribution in non-Gaussian settings. The criterion can also be thought of as a *scoring* function (i.e., a function that assigns value) because it prioritizes data points with the highest *value* in terms of improving the statistics of the target observable.

3 Model Uncertainty Quantification

The LW-US selection criterion requires an estimate for the epistemic uncertainty $\sigma^2(x)$ (in Equation 14) of the model. In many settings, the uncertainty is quantified using traditional Bayesian supervised learning methods such as Bayesian regression or Gaussian process regression [72, 8, 7, 87]. However, these methods are limited: Bayesian regression may fail when modeling nonlinear systems, and Gaussian process regression suffers from performance issues on high-dimensional or large datasets. Newer neural network architectures that can quantify uncertainty can overcome these problems [54, 36, 32, 90, 58, 91, 45]. We will focus on ensembles of neural networks (E-NN) and dropout neural networks (D-NN), but a comprehensive study of various computational methods for quantifying the uncertainty of surrogate models for complex dynamical systems can be found in Guth et al. [32], and further methods are described in Angelopoulos and Bates [2]. A key advantage of using E-NN or D-NN is that they leverage existing neural networks developed for the application of interest, eliminating the need to construct a separate surrogate model for uncertainty quantification and allowing users to retain the architecture that best suits their system, regardless of its complexity.

3.1 Ensemble of Neural Networks

In the E-NN, multiple models with identical architectures and hyperparameters are trained on the same dataset, each initialized with different random weights. The resulting prediction $\hat{y}$ is the mean of the n neural network predictions $\hat{y}_i$

$$\hat{y}(x) = \frac{1}{n}\sum_{i=1}^{n} \hat{y}_i(x) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{M}_i(x) \tag{15}$$

where $\hat{y}_i$ is the prediction of the i^{th} neural network in the ensemble. Similarly, the model uncertainty can be quantified using the variance of the predictions

$$\sigma^2(x) = \frac{1}{n-1}\sum_{i=1}^{n}\left(\hat{y}_i(x) - \hat{y}(x)\right)^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(\mathcal{M}_i(x) - \hat{y}(x)\right)^2. \tag{16}$$

Guth et al. [32] and Pickering and Sapsis [56] showed that even small ensembles of neural networks, as low as $N = 2$, can still perform well for the purpose of quantifying uncertainty for active learning.

3.2 Ensemble of Dropout Neural Networks

In a D-NN, only one model is trained, but the model includes dropout layers during both training and inference to introduce stochasticity [82]. During the prediction step, multiple predictions are made with different randomly dropped nodes, resulting in higher variance predictions [25]. Again, the resulting prediction is the mean of all the predictions, and the variance can be used to quantify uncertainty in the model. The dropout layers require additional training time, but only one model is trained, so the overall computation time is lower for the D-NN.

4 Practical Considerations for Evaluating the Likelihood-Weighted Selection Criterion in High-Dimensional Systems

This section presents the complete procedure for evaluating the LW-US selection criterion for high-dimensional or functional inputs, with a focus on practical implementation. We discuss how to handle cases where the training and inference datasets differ — an important consideration in real-world applications. We also explain how to compute probability densities.

As a reminder, the LW-US selection criterion depends on both the input x and the corresponding predicted observable $\hat{y}(x)$, and it incorporates three terms: (i) the predictive uncertainty $\sigma^2(x)$, (ii) the pdf of the input space $p_x(x)$, and (iii) the pdf of the observable $p_{\hat{y}}(\hat{y}(x))$:

$$q_{\text{LW-US}}(x) = \sigma^2(x)\frac{p_x(x)}{p_{\hat{y}}(\hat{y}(x))}. \tag{17}$$

We consider a labeled dataset $\mathcal{D}_{\text{cand}}$ from which training (tr) pairs $\{\mathcal{X}_{\text{tr}}, y_{\text{tr}}\}$ are selected, and a separate set of inputs $\{\mathcal{X}_{\text{inf}}\}$ for which the model will be deployed and evaluated during inference (inf). Here, $\mathcal{X}$ denotes an infinite-dimensional input field. Because the observable's true value in the inference setting is unknown during training, we optimize the selection of training points with respect to the predicted observable. This affects the output weighting term, which is evaluated over candidate training points but incorporates knowledge of the output distribution anticipated at inference time:

We define $\mathcal{M}^i_{\mathcal{S}}$ as the i^{th} member of a probabilistic ensemble of size n trained on $\mathcal{S} = [\mathcal{X}_1, \ldots, \mathcal{X}_P]$, a subset of P selected training inputs from the full labeled dataset $\mathcal{D}_{\text{cand}}$. We evaluate the ensemble on both the inputs of $\mathcal{D}_{\text{cand}}$ as well as $\mathcal{X}_{\text{inf}}$ to obtain $\hat{y}_{\text{tr}}$ and $\hat{y}_{\text{inf}}$, respectively. From these, we get:

(1) Uncertainty The predictive uncertainty at the candidate training points $\mathcal{X}$ is estimated as the variance of the predicted observable across the ensemble of models $\mathcal{M}_{\mathcal{S}}^i$:

$$\sigma^2(\mathcal{X}) = \frac{1}{n-1}\sum_{i=1}^{n}\left(\hat{y}_{\text{tr}}^i(\mathcal{X}) - \hat{y}_{\text{tr}}(\mathcal{X})\right)^2 \tag{18}$$

(2) Input Probability Estimating the input distribution p_x is a necessary step in applying the likelihood-weighted selection criterion, but it is generally intractable in high-dimensional settings. To address this, we reduce the dimensionality of the input fields $\mathcal{X}$ using weighted Principal Component Analysis (PCA) described in Appendix A, a widely used and computationally efficient technique in physics-based applications [11, 83].

The inputs are projected onto the modes ψ_i of the first k principal components, yielding a reduced representation $\mathbf{x}$, from which the density $p_{\mathbf{x}}(\mathbf{x})$ is estimated:

$$\mathcal{X} \approx \mathbf{x} = \left\langle \mathcal{X}, \{\psi_i\}_{i=1}^{k} \right\rangle_w . \tag{19}$$

The brackets correspond to the inner product operation of the PCA reconstruction, and the number of components k can be selected according to the PCA reconstruction error based on the decay of eigenvalues. The resulting low-dimensional space allows for density estimation using a Gaussian approximation, kernel density estimation (KDE) with a Gaussian kernel, or other suitable methods.

The purpose of estimating the input distribution is to prioritize likely scenarios in the training set. Although the density is approximated in a reduced space, the full high-dimensional structure of the inputs can still be represented within the machine learning model. While we use PCA in this work, other reduced-order modeling techniques can be used, as long as they preserve the key statistical properties of the input space [18].

(3) Output Probability The density of the observable during inference, p_{inf}, is estimated using KDE with a Gaussian kernel applied to the model predictions for the inference set $\hat{y}_{\text{inf}}$. KDE is fast and easy to compute for one-dimensional variables, and can be further accelerated using Fast Fourier Transform (FFT)-based methods. This density is then evaluated at the predicted observable for the training points $\hat{y}_{\text{tr}}$:

$$p_{\text{inf}}(\hat{y}_{\text{tr}}(\mathcal{X})) \tag{20}$$

We note that this density is approximated using predictions for all input samples $\mathcal{X}_{\text{inf}}$, so the resulting estimate is well-resolved and considered representative of the predictive distribution of the observable at inference time.

The final selection criterion can be written as

$$q_{\text{LW-US}}(\mathcal{X}) = \sigma^2(\mathcal{X})\frac{p_{\mathbf{x}}(\mathbf{x})}{p_{\text{inf}}(\hat{y}_{\text{tr}}(\mathcal{X}))} \tag{21}$$

by modifying Equation 17 according to Equations 18, 19, and 20.

5 Application to the Majda-McLaughlin-Tabak (MMT) Model

We first apply the described method to the MMT model, a one-dimensional dispersive nonlinear wave model that is useful for studying turbulence and rogue waves [48]. More details on the overall system can be found in Cai et al. [12], Zakharov et al. [88], Pushkarev and Zakharov [59], Cousins and Sapsis [19], Zakharov et al. [89]. The system is described by the governing equation

$$iu_t = |\partial_x|^{\alpha} u + \lambda |\partial_x|^{-\beta/4} \left(\left| |\partial_x|^{-\beta/4} u \right|^2 |\partial_x|^{-\beta/4} u \right) + iDu \tag{22}$$

where the output u is a complex scalar representing the wave amplitude, α and β are parameters of the system, and D is a selective Laplacian that eliminates high wave numbers. For $\alpha = 1/2$ and $\beta = 0$, the equation can be rewritten in the wave number space with forcing $f(k)$

$$\hat{u}(k)_t = -i|k|^{1/2}\hat{u}(k) - i\lambda |\hat{u}(k)|^2 \hat{u}(k) + \widehat{Du}(k) + f(k) \tag{23}$$

where the selective Laplacian is defined as

$$\widehat{Du}(k) = \begin{cases} -(|k| - k^*)^2 \hat{u}(k) & \text{if } |k| > k^* \\ 0 & \text{if } |k| \leq k^* \end{cases}. \tag{24}$$

This operator $\widehat{Du}(k)$ prevents wave numbers above a threshold k^*: for small wave amplitudes, the output pdf appears to be Gaussian, but for large wave amplitudes, the output pdf is very heavy-tailed. The stochastic complex initial conditions $u(x, t = 0)$, which are Gaussian, are obtained from the covariance function

$$k(x, x') = \sigma_u^2 \exp\left(i2 \sin^2(\pi(x - x')) \right) \exp\left(-\frac{2 \sin^2(\pi(x - x'))}{l_u^2} \right) \tag{25}$$

with $\sigma_u = 1$ and $l_u = 0.35$, and they are reduced to $2m$ dimensions, with m real and m imaginary components, using the Karhunen-Loève expansion

$$u(x, t = 0) \approx \sum_{j=1}^{m} \alpha_j \sqrt{\lambda_j} \phi_j(x), \ \forall \ x \in [0, 1) \tag{26}$$

to transform the original high-dimensional data into a set of orthogonal components. In this step, the Karhunen-Loève expansion is the continuous analogue of PCA described in Section A. The grid is periodic over $[0, 1)$ and discretized into 512 points, m is set to 4, the timestep is $dt = 0.001$, the parameters of the equation are $\lambda = -0.5$ and $k^* = 20$, and there is no forcing, $f(k) = 0$. As in Pickering et al. [57] and Guth et al. [32], we seek to train a standard fully-connected NN (FC-NN) to predict the maximum future wave amplitude over a given time horizon, an extreme event, as a function of the $2m$ stochastic initial conditions $\vec{\alpha}$

$$y(\vec{\alpha}) = ||\text{Re}(u(x, T = 50; \vec{\alpha}))||. \tag{27}$$

5.1 Datasets

To better mimic the characteristics of datasets that are found in the real world, we make use of two datasets: points obtained from inputs that follow a Gaussian distribution, $\mathcal{D}_{p_X}$, and points obtained with Latin hypercube sampling (LHS), $\mathcal{D}_{\text{LHS}}$. For Monte Carlo sampling, we select candidate training points from $\mathcal{D}_{p_X}$ because this distribution more closely resembles naturally-occurring datasets. As a result, we compare our proposed method to a more rigorous benchmark (the Monte Carlo sampling performs worse when applied to points from $\mathcal{D}_{\text{LHS}}$, as LHS more evenly covers the input space, including the tails). For US and LW-US sampling, we select candidate training points from $\mathcal{D}_{\text{LHS}}$ because this dataset more completely represents all the achievable values, including the tails of the distribution. We compute the error metrics on $\mathcal{D}_{\text{LHS}}$ to evaluate the model's ability to capture the tails of the distribution.

5.2 Machine Learning Architecture and Active Learning Hyperparameters

The ML model learns a mapping from the initial conditions to the maximum wave height over a given time horizon. We test both the E-NN and the D-NN described in Section 3. For the E-NN, we use an ensemble of size 2, and for the D-NN, we use an ensemble of size 5 (both sufficient as shown in Guth et al. [32] and Pickering and Sapsis [56]). Even though the size of the D-NN ensemble is higher, the overall process takes less time because only one model is trained. Based on the results of a hyperparameter grid search, we set the number of layers to eight, the number of neurons to 250, the activation to ReLU, the number of epochs to 3000, and the batch size to the floor of half the number of points in the training set. For the D-NN, we set the dropout rate to 50%, a value shown to be effective at estimating uncertainty in Guth et al. [32]. The batch size is adjusted at each iteration to accommodate the growing dataset size while keeping the training error within a reasonable range. We initialize the algorithm with a training set of 10 randomly chosen points. At each iteration, we add a batch of 10 points to the training set (points that correspond to the maximum value of the selection criterion), and we re-initialize the model to avoid getting stuck in any bad local minima found during early iterations. The first iteration of the algorithm, using 10 points, took about 1 minute on a standard CPU, while the final iteration, with 1500 points, took about 5 minutes.

5.3 Results

The results obtained from carrying out the algorithm for 150 iterations (up to 1500 points — 1.5% of the full dataset) for randomly chosen points (MC), input-weighted uncertainty sampling (US), and likelihood-weighted uncertainty sampling (LW-US) are shown in Figure 3. Because we compute the mean squared error (MSE) with the Latin hypercube sampling dataset $\mathcal{D}_{\text{LHS}}$, we weight the error by the input distribution as follows

$$\text{MSE} = \sum_{i=1}^{N} (y_i - \hat{y}_i(\vec{\alpha}_i))^2 p_X(\vec{\alpha}_i). \tag{28}$$

The LW-US outperforms MC and US in minimizing the error in the tail of the pdf, and this is seen again in Figure 4. While the E-NN outperforms the D-NN in accuracy, the D-NN has faster training times, making it a better option for more computationally expensive problems.

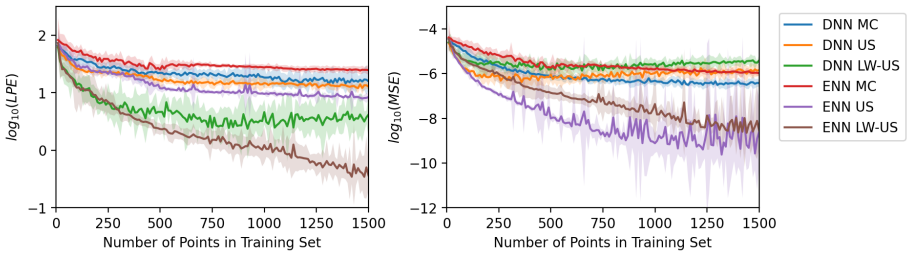

Fig. 3 Error convergence curves of the MMT predictions. The log of the LPE error (left) and log of the MSE (right) are plotted as a function of the number of points in the training set for both the E-NN and D-NN implementations of the MC, US, and LW-US selection criteria. The shading represents ±1 standard deviation of the five experiments. LW-US (both E-NN and D-NN) significantly outperforms the other selection criteria with respect to LPE. E-NN US initially achieves a better MSE, but E-NN LW-US eventually achieves a similar error.

5.4 Interpreting the Selected Points: Multidimensional Scaling

To gain insights into the behavior of the LW-US active search algorithm, we visualize the eight-dimensional selected input points using multidimensional scaling (MDS) [20, 21, 43, 44, 71, 77]. MDS projects high-dimensional points to a two-dimensional subspace with the requirement that a chosen distance metric be preserved between points — points that are more spread apart in the original space must be spread apart in the lower-dimensional space, and vice versa. Here, we use the standard Euclidean distance as the distance metric. The two-dimensional projection shown in Figure 4 reveals that points chosen by the LW-US selection criterion are more spread out in the two-dimensional projection than points chosen by other selection criteria. The results of MDS suggest that drawing points that are more "spread out" can be valuable for predicting extreme events.

6 Application to Debiasing Operator for Coarse-Resolution Climate Model Outputs

We now apply the likelihood-weighted active selection framework to outputs from a ML-based climate model to accelerate training and improve the prediction of extreme weather events [34, 66]. Forecasting future extreme weather events requires the generation of high-resolution climate model outputs for a range of potential scenarios [66, 67, 5, 24, 79]. Historically, such simulations have been produced from numerical solvers based on physical equations and parameterizations [80, 49, 51, 85, 22, 75, 27]. However, these solvers must resolve turbulent dynamics

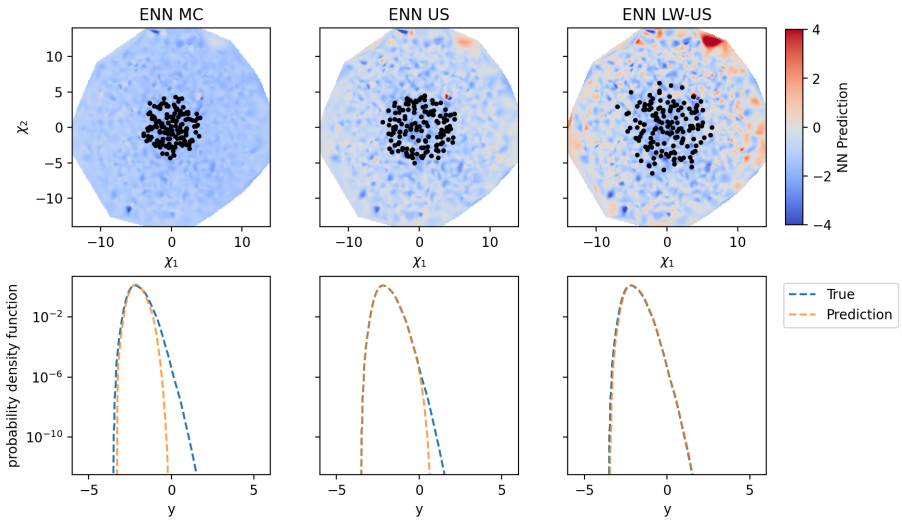

Fig. 4 Visualization of selected MMT input points. In the top row, the 8-dimensional space is projected to a 2-dimensional space with multi-dimensional scaling. Each plot shows the spread of the optimally selected points in black over the prediction made from the neural network trained after 150 iterations with training data obtained from MC, US, and LW-US (left to right). The rightmost plot suggests that points chosen by LW-US are more spread out. In the bottom row, the predicted pdf is compared to the true pdf after 150 iterations for MC, US, and LW-US, and LW-US best matches the tail of the distribution.

across scales from millimeters to thousands of kilometers, making them computationally expensive and reliant on extensive parameter tuning and complicated closure terms.

Recent advances in ML architectures, algorithms, and hardware have enabled alternative data-driven climate models [62, 64, 63, 40, 76, 53, 6, 14, 4, 16, 65, 41, 50]. To build on this growing field, we apply our likelihood-weighted selection approach to a recently proposed debiasing framework [4] that learns a correction operator between low-resolution free-running simulations and high-resolution reanalysis data. This operator improves coarse models without requiring full-resolution computation. In this setting, the likelihood-weighted data selection framework improves the prediction of extreme weather events in ML climate models. Although we focus on this particular model, our approach is model-agnostic and broadly applicable to ML-based climate modeling with large candidate training sets.

6.1 Datasets

The coarse-resolution simulations are obtained from version 2 of the Energy Exascale System Model (E3SM) Atmosphere Model (EAMv2) [22, 27, 84]. The dataset consists of temperature (T), specific humidity (Q), zonal velocity (U), and meridional velocity (V) at a 1° (approximately 110km) resolution, and we only consider the vertical layer closest to the surface of the Earth. The high-resolution target dataset is the European Centre for Medium-Range Weather Forecasts (ECMWF)

Reanalysis version 5 (ERA5) [35]. ERA5 has a resolution of 0.25° (approximately 31km), but it is projected onto the E3SM grid for the purpose of this debiasing framework. For all datasets, we use 10 years, from January 2007 to December 2017, sampled 8 times per day.

During training, the machine learning correction model learns a mapping from the coarse-resolution simulation nudged toward reanalysis (denoted $\mathcal{X}^{\mathrm{NUDG}}$) to the high-fidelity reanalysis fields (denoted $\mathcal{U}^{\mathrm{ERA5}}$). The nudging procedure is described in detail in Barthel Sorensen et al. [4]. We define the candidate dataset of potential training pairs as

$$\mathcal{D}_{\mathrm{cand}} = \{(\mathcal{X}^{\mathrm{NUDG}}, \mathcal{U}^{\mathrm{ERA5}})\}, \tag{29}$$

where $\mathcal{X}^{\mathrm{NUDG}}$ are nudged model fields and $\mathcal{U}^{\mathrm{ERA5}}$ are the corresponding reanalysis fields. During inference, the trained model acts as a debiasing operator, correcting free-running (i.e., *un-nudged*) coarse-resolution simulations under different climate scenarios (denoted $\mathcal{X}^{\mathrm{CR}}$). The inference dataset is

$$\mathcal{D}_{\mathrm{inf}} = \{(\mathcal{X}^{\mathrm{CR}}, \hat{\mathcal{U}}^{\mathrm{CR}})\} = \{(\mathcal{X}^{\mathrm{CR}}, \mathcal{M}(\mathcal{X}^{\mathrm{CR}}))\}, \tag{30}$$

where $\mathcal{X}^{\mathrm{CR}}$ are coarse-resolution model fields from a new scenario and $\mathcal{M}(\mathcal{X}^{\mathrm{CR}})$ is the model's debiased prediction. Each sample consists of a snapshot in time of the full spatial field. The observable of interest, $\mathcal{Y}$, can be extracted from $\mathcal{U}$. To compute the selection criterion, the dimensionality of the inputs is reduced using weighted PCA as in Section A with the weight $w(\xi) = w(\theta, \phi) = \sqrt{\sin\left(\frac{90^\circ - \theta}{180^\circ}\pi\right)}$.

6.2 Machine Learning Architecture and Active Learning Hyperparameters

The NN architecture, shown in Figure 5, is an encoder-decoder consisting of two-dimensional convolutional layers. The globe is divided into 25 sections (a 5×5 grid), and each section is padded to maintain spherical periodicity. The encoding convolutions are then applied independently to each section. The encoder is made up of one layer to split the globe, one layer to spherically pad the sections, three convolutional layers applied to each section to capture anisotropic local features, and one layer to merge the sections. Next, the decoder applies "deconvolutional" layers (or transpose convolutional layers) to map the latent space back to the desired dimension. Finally, the 25 sections are combined to recreate the full field. The batch size is set to 8, and the number of epochs is set to 150, as was done in [4]. The loss function used to guide the ML optimization is the MSE for which spatial points are weighted by latitude θ: $w(\theta) = \sqrt{\sin\left(\frac{90^\circ - \theta}{180^\circ}\pi\right)}$.

The probabilistic architecture is a two-member ensemble neural network. We initialize the algorithm with a training set of ten randomly chosen points and, at each iteration, add ten points that maximize the selection criterion to the training set. For the MC case, we add twenty random points at each iteration because future iterations do not depend on previous iterations. There are 29,200 (10 years $\times$ 365.25 days $\times$ 8 measurements per day) candidate samples that can be chosen by the selection criterion, and we only evaluate the method up to 750 points in the training set (2.6% of all data). The first iteration of the algorithm, using 10 points, took about 7 to 8 minutes on a standard CPU, while the final iteration, with 750

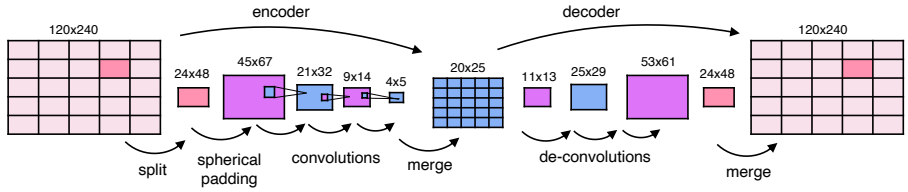

Fig. 5 Climate debiasing operator neural network architecture. The NN architecture splits the Earth into sections that are individually passed through convolutional encoder-decoder layers.

points, took about 50 to 60 minutes, highlighting the increased costs associated with larger training datasets. Training could have been made faster with a GPU.

To evaluate the framework, we generate "ground truth" data by training a model with 100% of the samples in $\mathcal{X}^{\mathrm{NUDG}}$ and $\mathcal{U}^{\mathrm{ERA5}}$: we call this model $\mathcal{M}_{100}$. Then, we use the model $\mathcal{M}_{100}$ to make a prediction from the un-nudged coarse-resolution dataset $\mathcal{X}^{\mathrm{CR}}$: we call this prediction $\mathcal{M}_{100}(\mathcal{X}^{\mathrm{CR}})$. Figure 6 shows the mean of the reference reanalysis dataset and the mean of the model output given the test data CR as input. At each iteration, we compute error metrics for

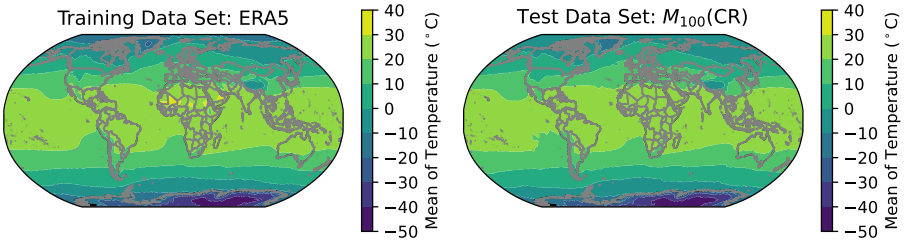

Fig. 6 Mean of ERA5 and Mean of $\mathcal{M}_{100}(\mathcal{X}^{\mathrm{CR}})$ for temperature computed over the ten years of data. During the training phase, the nudged dataset is mapped to the ERA5 dataset. During the testing phase, the coarse-resolution dataset is provided as input to the trained model.

$\mathcal{M}_{\mathcal{S}}(\mathcal{X}^{\mathrm{CR}})$, which refers to the model trained on a subset of the data, with respect to $\mathcal{M}_{100}(\mathcal{X}^{\mathrm{CR}})$, the model trained on the full dataset. For each test case, we perform five or six experiments to evaluate the statistics of the MSE and LPE over the different experiments.

6.3 Results

We test the method on three cases: (i) the first PCA coefficient for temperature over the entire globe, obtained using weighted PCA (Figure 7), (ii) temperature in Paris (Figure 10), and (iii) specific humidity in Ankara (Figure 13). These last two locations were chosen randomly from a list of cities that have experienced extreme heat waves (in the case of temperature) or extreme floods (in the case of specific humidity) in the last few decades. For each example, we plot the true

distribution in green, the predicted distribution obtained from MC sampling in blue, and the predicted distribution obtained from LW-US sampling in orange. We also plot a Gaussian distribution with the same mean and standard deviation as the true distribution, shown as a dashed gray line. When both tails are heavy or non-Gaussian, we compute the LPE over the full distribution; however, if only one tail is heavy (as in the case of humidity in Ankara), we compute the LPE for only the half of the distribution corresponding to the heavy tail.

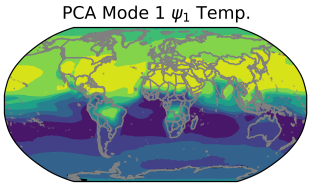

Fig. 7 First weighted PCA mode of the global temperature field.

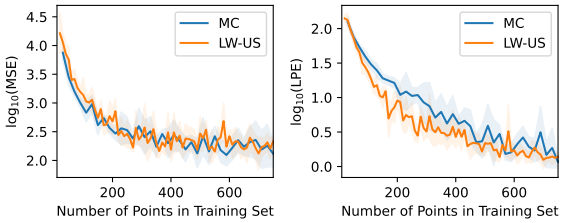

Fig. 8 The log of the MSE and LPE are shown for MC and LW-US with respect to predicting the first PCA coefficient of global temperature. The shading represents ± 1 standard deviation across five experiments.

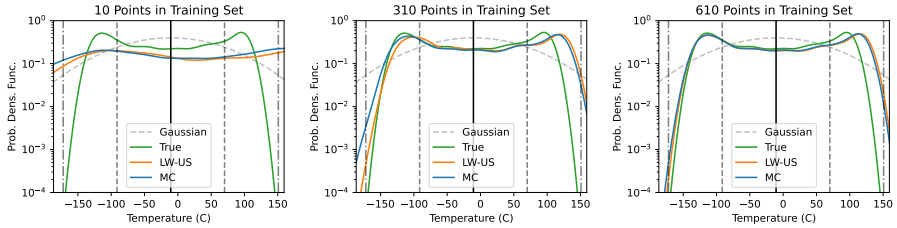

Fig. 9 The true pdf (green) of the first PCA coefficient for temperature is compared to the pdf obtained from predictions made with MC (blue) and LW-US (orange). The Gaussian pdf is also shown in the dashed gray line. The black vertical line denotes the mean of the true distribution, and the dashed lines denote $\pm 1\sigma$ and $\pm 2\sigma$. LW-US is able to better match the left tail of the true pdf.

Looking at the LPE as a function of the number of points in the training set in Figures 8, 11, and 14, we see that LW-US outperforms MC in all cases. The improvement obtained from using LW-US can also be seen in the pdf plots (Figures 9, 12, and 15) - LW-US does a better job matching the tails of the distribution, especially in cases where the distribution is non-Gaussian or exhibits heavy tails. We also observe that the improvement obtained from using LW-US occurs at a different number of iterations for the different test cases. Looking at the MSE, the error is similar to MC in the cases involving temperature (Figures 8 and 11), but worse in cases involving humidity (Figure 14). However, LPE, not MSE, is the primary metric of interest in our framework, as it better reflects the performance on the tails of the distribution. It is not expected that our method performs well in minimizing the MSE.

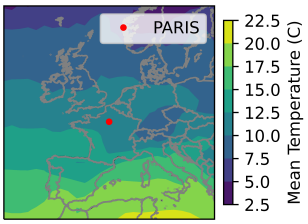

Fig. 10 Mean temperature in the region surrounding Paris, France.

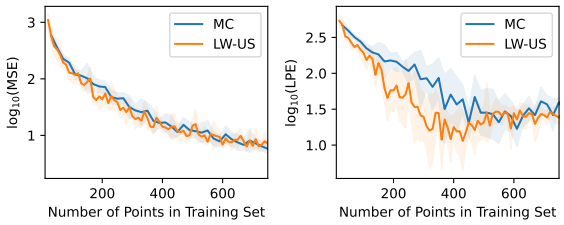

Fig. 11 The log of the MSE and LPE are shown for MC and LW-US with respect to predicting the temperature in Paris given a global model. The shading represents ± 1 standard deviation across six experiments.

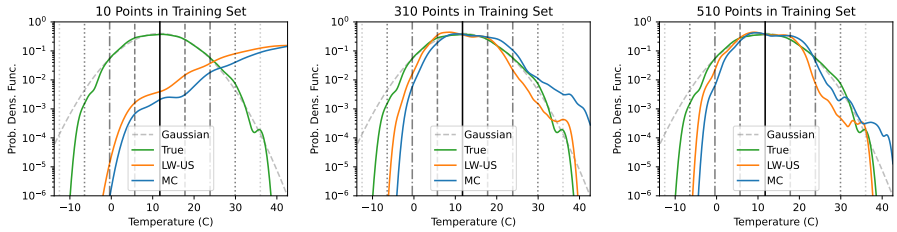

Fig. 12 The true pdf (green) of the temperature in Paris is compared to the pdf obtained from predictions made with MC (blue) and LW-US (orange). The Gaussian pdf is also shown in the dashed gray line. The black vertical line denotes the mean of the true distribution, and the dashed lines denote the $\pm 1\sigma$, $\pm 2\sigma$, $\pm 3\sigma$, and $\pm 4\sigma$. LW-US is able to better match the tails of the true pdf with just 310 points.

6.4 Interpreting the Selected Points: Clustering

After selecting the training points, the next goal is to determine whether the points chosen by the algorithm have any relevant physical meaning. For example, scientists could be interested in determining if these points are related to important system dynamics, if they can be attributed to physical phenomena (e.g., turbulence, atmospheric rivers, tropical cyclones, etc.), or if their physical interpretation depends on the target's predicted output. Understanding why the optimal points were selected also reduces some of the "black box" nature of the ML-based algorithm.

We present a clustering framework to mechanistically identify and define the dynamics of these points of interest. Clustering, a form of reduced-order modeling in which observations are clustered around centroids, has been used for climate datasets in other applications [46]. In the case of a dynamic system like the climate, the observations (or samples) are snapshots in time of the system. We select cluster centroids from the entire reference dataset of PCA time coefficients $\alpha_j(t) = \langle \mathcal{U}^{\mathrm{ERA5}}, \psi_j^{\mathrm{ERA5}} \rangle_w$, using the standard k-means algorithm. We set the number of clusters to six in all cases. The resulting cluster centroids are projected back onto the PCA modes to visualize the spatial patterns. These cluster centroids are then used to predict the cluster labels of the new subset of the data chosen by the algorithm. This step assigns the optimal points to relevant cluster centers which allows us to determine if the points chosen by the algorithm are associ-

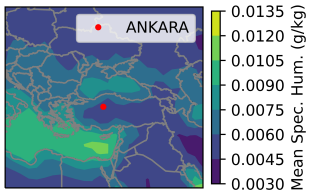

Fig. 13 Mean specific humidity in the region surrounding Ankara, Turkey.

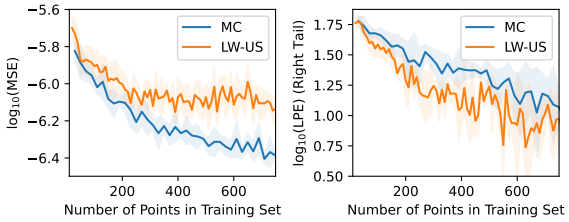

Fig. 14 The log of the MSE and LPE (over the right tail) are shown for MC and LW-US with respect to predicting the specific humidity in Ankara given a global model. Here, we report the LPE for the right tail only, as it exhibits heavy-tailed behavior. The shading represents ±1 standard deviation across five experiments. LW-US is better for minimizing LPE, but worse for minimizing MSE in the case of specific humidity.

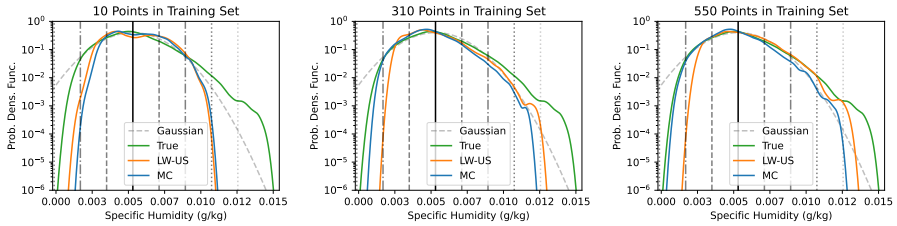

Fig. 15 The true pdf (green) of the specific humidity in Ankara is compared to the pdf obtained from predictions made with MC (blue) and LW-US (orange). The Gaussian pdf is also shown in the dashed gray line as a way to assess how heavy each tail is. The black vertical line denotes the mean of the true distribution, and the dashed lines denote the $\pm 1\sigma$, $\pm 2\sigma$, $\pm 3\sigma$, and $\pm 4\sigma$. LW-US is able to better match the heavy tail (right tail) of the true pdf.

ated with noteworthy dynamical phenomena. The ultimate goal is to interpret the physical meaning of the points that were chosen for training.

In Figures 16 and 17, the six clusters are ordered from the most frequent (Cluster #1) to the least frequent (Cluster #6) in the full dataset. For temperature in Paris (Figure 16), we found that points belonging to Cluster #6 are more relevant to the dynamics of extreme weather events. Upon further examination, the shape of Cluster #6 suggests a potential heat dome over Paris and the surrounding region [39]. In the case of specific humidity in Ankara (Figure 17), Cluster #4 contains most of the optimal training points, and the distribution of points between clusters is significantly more skewed. By applying clustering to the points selected by the algorithm, we can identify extreme weather events in an unsupervised manner.

7 Conclusions

To address the challenge of training ML models from large, high-dimensional candidate datasets, we introduce a likelihood-weighted active data selection framework that improves the prediction of extreme event statistics. At each iteration, the framework selects training points using a criterion that emphasizes samples likely

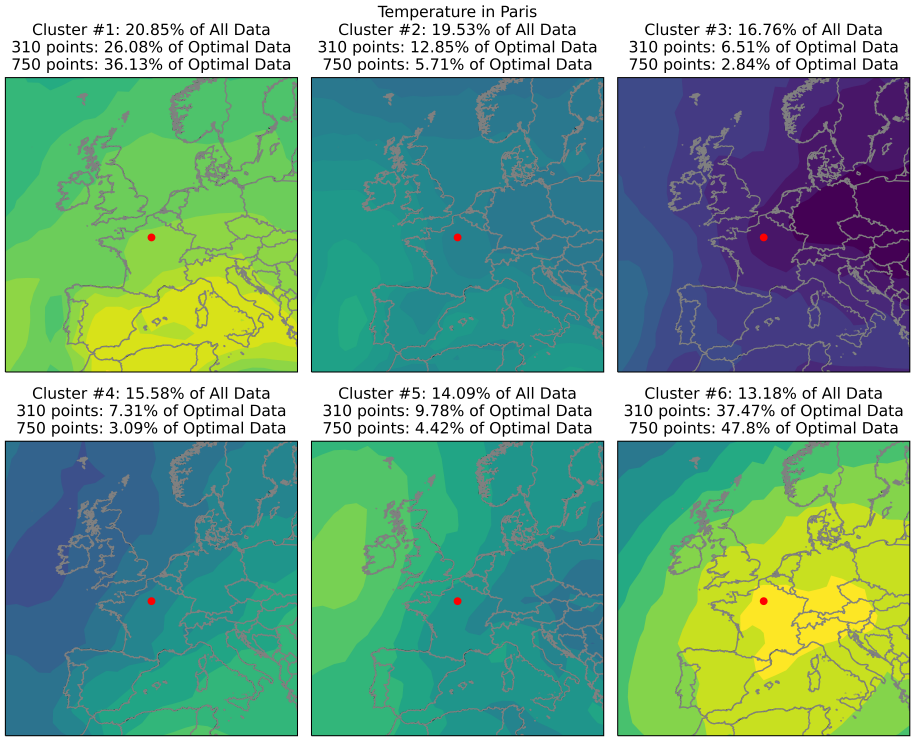

Fig. 16 Clusters for temperature in Paris. With 310 points in the training set, Cluster #6 only represents 13.18% of all data but 37.47% of the optimal data. Cluster #6 exhibits a blocking pattern over most of France. The next most occurring cluster is Cluster #1 that represents standard zonal flow, typical for normal weather events.

to produce extreme events. The uncertainty in the model is quantified according to the variance of an ensemble of neural networks, and the dimensionality of the high-dimensional inputs is reduced using weighted principal component analysis. The framework is model-agnostic and suitable for high-dimensional datasets. We note that the method requires training at each iteration and may not perform effectively if the datasets of candidate points do not consist of sufficient data to capture the true underlying statistics. We demonstrated the success of the framework on both a synthetic problem (dispersive wave turbulence) and a real-world problem (a correction operator for coarse-resolution climate models). In both cases, likelihood-weighted active data selection achieved a lower error in the tails of the probability distribution with fewer training points. In the real-world problem, clustering showed that the method was selecting points relevant to extreme weather events. Future work could consider reformulating the algorithm to identify precursors to extreme events. Down the line, the developed approach has the potential to be used as a compression algorithm that preserves the information associated with extreme events in vast datasets.

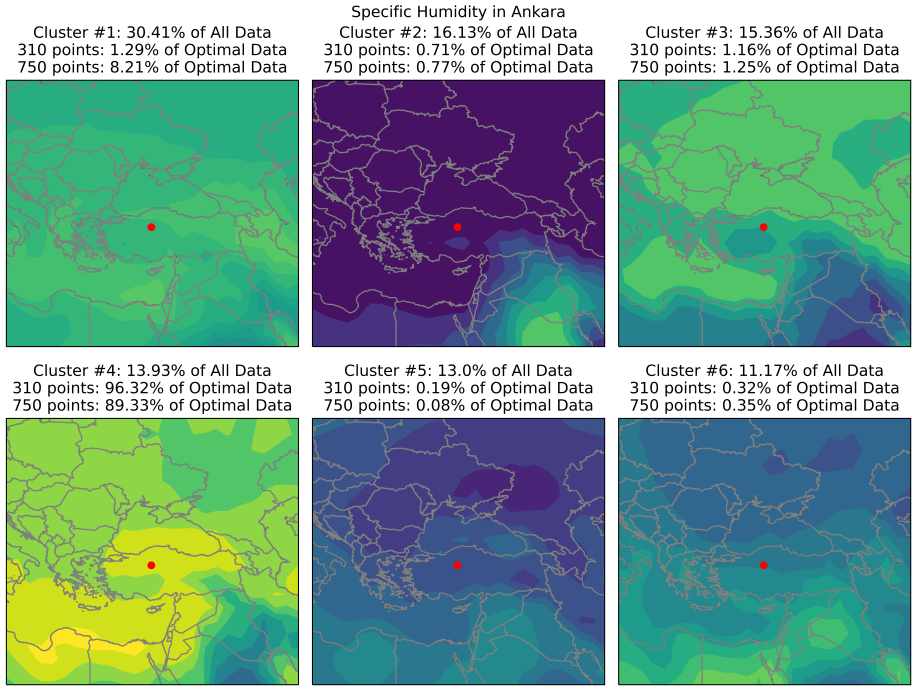

Fig. 17 Clusters for specific humidity in Ankara. With 310 points in the training set, Cluster #4 only represents 13.93% of all data but 96.32% of the optimal data. The next most occurring cluster is Cluster #1. Between 310 points and 750 points, more points are chosen from Cluster #1.

Nomenclature

Machine Learning Model

$\mathcal{X}$	Input space; often high-dimensional fields
$\mathcal{Y}$	Output space (target or observable)
$\mathcal{U}$	Intermediate space $\mathcal{U}$, derived from the input $\mathcal{X}$, used to obtain the scalar observable $\mathcal{Y}$
$\mathcal{M}_\theta$	ML model mapping inputs $\mathcal{X}$ to outputs $\mathcal{Y}$ (or sometimes $\mathcal{X}$ to $\mathcal{U}$), parameterized by θ
$\hat{y}$	ML model prediction
$\mathcal{L}(y, \hat{y})$	Loss function used to train the ML model
$\mathcal{D}_{\text{train}}$	Training set consisting of pairs (x_j, y_j) used to train ML model

Active Sampling Algorithm

$\mathcal{D}_{\text{train}}^{(t)}$	Dataset of points used for training at iteration t
$\mathcal{D}_{\text{cand}}^{(t)}$	Dataset of potential candidate points at iteration t
$\mathcal{B}^{(t)}$	Batch of b selected points added at iteration t
$\hat{y}_{\text{tr}}$	Model predictions on training dataset
$\hat{y}_{\text{inf}}$	Model predictions on inference dataset

Selection Criterion

$q(x)$	Selection criterion (acquisition function) used in active learning
$q_{\text{US}}(x)$	Uncertainty sampling criterion
$q_{\text{LW-US}}(x)$	Likelihood-weighted uncertainty sampling criterion
$\sigma^2(x)$	Epistemic variance at input x, estimated from ensemble predictions
$\sigma^2(z; x)$	Epistemic variance at z after adding sample at x

Probability Density Function

p_x	Input distribution
p_y	Output distribution
$p_{\hat{y}}$	Output distribution of model predictions
$p_y^+(y \mid x)$	Output distribution after hypothetically adding x
p_{inf}	Output distribution of predictions made during inference
LPE	Log-PDF error; measures difference between true and predicted log-pdfs

Ensemble of Neural Networks

n	Number of ensemble members in E-NN
$\mathcal{M}_i$	The i-th ensemble model
$\hat{y}_i$	Prediction of the i-th neural network in the ensemble

Principal Component Analysis

ψ_i	Modes from weighted PCA
$\langle \mathcal{X}, \{\psi_i\} \rangle_w$	Weighted inner product between the input $\mathcal{X}$ and PCA modes ψ_i, also known as the PCA coefficients.

Funding Declaration

The work was funded through the AFOSR Award FA9550-23-1-0517 and the National Science Foundation Graduate Research Fellowship Grant No. 2141064. This work used Anvil at Purdue University through allocation MTH240010 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (AC-

CESS) program, which is supported by U.S. National Science Foundation grants No. 2138259, 2138286, 2138307, 2137603, and 2138296 [9, 81].

Author Contributions

B.C. and T.S. both contributed to the methodology. B.C. developed the software, conducted the investigation and analysis, prepared the figures, and wrote the original draft. T.S. contributed to the conceptualization, data curation, and resources, and provided input during analysis and editing. All authors reviewed and approved the final manuscript.

Competing Interests

The authors have no competing interests, or other interests that might be perceived to influence the results and/or discussion reported in this paper.

Data Availability

The data and code are available at `https://github.com/biancach/LikelihoodWeightedActiveSampling`.

Acknowledgements We implement multidimensional scaling with the `MATLAB` `mdscale` function. We implement the kernel density estimate with the python function `FFTKDE` from the package `KDEpy`. The neural networks are designed using the `tensorflow` library. We implement `k-means++` with `scikit-learn`.

A Weighted Principal Component Analysis for High-Dimensional Systems

Our input data consists of high- or infinite-dimensional fields, which makes direct estimation of input densities intractable. To overcome this, we first reduce the dimensionality of the input space using weighted principal component analysis (PCA) or, when the covariance function is known *a priori*, a Karhunen-Loève decomposition.

Formally, let $\mathbf{x}(\xi, t)$ denote a vector field defined over a spatial domain indexed by ξ, with temporal mean $\overline{\mathbf{x}}(\xi)$. We define the centered field

$$\mathbf{z}(t, \xi) = \mathbf{x}(t, \xi) - \overline{\mathbf{x}}(\xi), \tag{31}$$

and represent it in terms of orthonormal spatial modes $\{\psi_j(\xi)\}_{j=1}^{k}$, weighted according to the domain geometry:

$$\mathbf{z}(t, \xi) = \sum_{j=1}^{N} \alpha_j(t)\psi_j(\xi) \approx= \sum_{j=1}^{k} \alpha_j(t)\psi_j(\xi), \tag{32}$$

where $\alpha_j(t)$ are time-dependent PCA coefficients, and k is the number of retained modes.

To incorporate the spatial geometry, we define a weighted inner product:

$$\langle \mathbf{x}_1, \mathbf{x}_2 \rangle_w = \int_{\xi} w^2(\xi)\mathbf{x}_1(\xi)\mathbf{x}_2(\xi)\, d\xi, \tag{33}$$

with a spatial weighting function $w(\xi)$. For example, the MMT application uses uniform weights $w(\xi) = 1$ whereas the climate application accounts for spherical geometry by choosing

$$w(\theta, \phi) = \sqrt{\sin\left(\frac{90^\circ - \theta}{180^\circ}\pi\right)}, \tag{34}$$

where θ and ϕ denote latitude and longitude, respectively.

The covariance operator is estimated by time-averaging:

$$\mathbf{R}(\xi_1, \xi_2) = \frac{1}{T}\int_0^T \mathbf{z}(t, \xi_1)\mathbf{z}(t, \xi_2)\, dt \approx \frac{1}{n_t}\mathbf{Z}\mathbf{Z}^T, \tag{35}$$

where $\mathbf{Z} \in \mathbb{R}^{N \times n_t}$ is the snapshot matrix. The PCA modes ψ_j can be found by solving the eigenvalue problem

$$\langle \mathbf{R}(\cdot, \xi), \psi_j(\cdot)\rangle_w = \lambda_j \psi_j(\xi), \tag{36}$$

with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{n_x} \geq 0$.

The PCA coefficients are computed as projections:

$$\alpha_j(t) = \langle \mathbf{z}(t, \cdot), \psi_j(\cdot)\rangle_w. \tag{37}$$

The primary purpose of this dimensionality reduction is to enable the use of the data selection algorithm even for systems with very high-dimensional inputs, by projecting the input field onto a lower-dimensional basis that facilitates density estimation. Additionally, the same methodology allows us to define a scalar observable y by selecting appropriate modes or functions ψ. For example, if the quantity of interest is the first PCA coefficient, we set $j = 1$; alternatively, ψ may represent spatial averages, maximum values, or values at specific points. In such cases, we use $\langle x, \cdot\rangle$, as in equation 36, to denote the corresponding extraction operator.

In our applications, this manifests as follows: In the climate modeling application (Section 6), weighted PCA is used first to reduce the input field dimensionality from the low-resolution climate model output. Optionally, it is also used to define the scalar observable y when this corresponds to a dominant mode of variability. In the MMT application (Section 5), the input initial conditions have a known covariance function and are obtained from a Karhunen-Loève expansion (equivalent to PCA under these conditions). In both cases, this approach projects the high-dimensional spaces onto a reduced basis, enabling a tractable evaluation of the input probability density and a flexible definition of the observable. In general, we note that the ML models themselves remain high-dimensional.

References

1. Albeverio S, Jentsch V, Kantz H, Dragoman D, Dragoman M, Elitzur AC, Silverman MP, Tuszynski J, Zeh HD (eds) (2006) Extreme Events in Nature and Society. The Frontiers Collection, Springer Berlin Heidelberg, Berlin, Heidelberg, DOI 10.1007/3-540-28611-X, URL `http://link.springer.com/10.1007/3-540-28611-X`
2. Angelopoulos AN, Bates S (2022) A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification. DOI 10.48550/arXiv.2107.07511, URL `http://arxiv.org/abs/2107.07511`, arXiv:2107.07511 [cs, math, stat]
3. Banerjee A, Dunson D, Tokdar S (2011) Efficient Gaussian Process Regression for Large Data Sets. URL `http://arxiv.org/abs/1106.5779`
4. Barthel Sorensen B, Charalampopoulos A, Zhang S, Harrop BE, Leung LR, Sapsis TP (2024) A Non-Intrusive Machine Learning Framework for Debiasing Long-Time Coarse Resolution Climate Simulations and Quantifying Rare Events Statistics. Journal of Advances in Modeling Earth Systems 16(3):e2023MS004122
5. Bauer P, Stevens B, Hazeleger W (2021) A digital twin of Earth for the green transition. Nature Climate Change 11(2):80–83, DOI 10.1038/s41558-021-00986-y, URL `https://www.nature.com/articles/s41558-021-00986-y`, publisher: Nature Publishing Group
6. Bi K, Xie L, Zhang H, Chen X, Gu X, Tian Q (2023) Accurate medium-range global weather forecasting with 3D neural networks. Nature 619(7970):533–538, DOI 10.1038/s41586-023-06185-3, URL `https://doi.org/10.1038/s41586-023-06185-3`

7. Blanchard A, Sapsis T (2021) Bayesian optimization with output-weighted optimal sampling. Journal of Computational Physics 425:109901, DOI 10.1016/j.jcp.2020.109901
8. Blanchard A, Sapsis T (2021) Output-Weighted Optimal Sampling for Bayesian Experimental Design and Uncertainty Quantification. SIAM/ASA Journal on Uncertainty Quantification 9(2):564–592, DOI 10.1137/20M1347486, URL https://epubs.siam.org/doi/10.1137/20M1347486
9. Boerner TJ, Deems S, Furlani TR, Knuth SL, Towns J (2023) ACCESS: Advancing Innovation: NSF's Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support. In: Practice and Experience in Advanced Research Computing, ACM, Portland OR USA, pp 173–176, DOI 10.1145/3569951.3597559, URL https://dl.acm.org/doi/10.1145/3569951.3597559
10. Brunton SL, Noack BR, Koumoutsakos P (2020) Machine Learning for Fluid Mechanics. Annual Review of Fluid Mechanics 52(1):477–508
11. Béranger B, Duong T, Perkins-Kirkpatrick SE, Sisson SA (2019) Tail density estimation for exploratory data analysis using kernel methods. Journal of Nonparametric Statistics 31(1):144–174, DOI 10.1080/10485252.2018.1537442, URL https://www.tandfonline.com/doi/full/10.1080/10485252.2018.1537442
12. Cai D, Majda AJ, McLaughlin DW, Tabak EG (1999) Spectral bifurcations in dispersive wave turbulence. Proceedings of the National Academy of Sciences 96(25):14216–14221, DOI 10.1073/pnas.96.25.14216, URL https://www.pnas.org/doi/abs/10.1073/pnas.96.25.14216, publisher: Proceedings of the National Academy of Sciences
13. Chaloner K, Verdinelli I (1995) Bayesian Experimental Design: A Review. Statistical Science 10(3), DOI 10.1214/ss/1177009939, URL https://projecteuclid.org/journals/statistical-science/volume-10/issue-3/Bayesian-Experimental-Design-A-Review/10.1214/ss/1177009939.full
14. Charalampopoulos AT, Zhang S, Harrop B, Leung LyR, Sapsis T (2023) Statistics of extreme events in coarse-scale climate simulations via machine learning correction operators trained on nudged datasets. DOI 10.48550/arXiv.2304.02117, URL http://arxiv.org/abs/2304.02117, arXiv:2304.02117 [physics]
15. Chattopadhyay A, Nabizadeh E, Hassanzadeh P (2020) Analog Forecasting of Extreme-Causing Weather Patterns Using Deep Learning. Journal of Advances in Modeling Earth Systems 12(2):e2019MS001958, DOI 10.1029/2019MS001958, URL https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2019MS001958
16. Chen L, Du F, Hu Y, Wang Z, Wang F (2023) SwinRDM: Integrate SwinRNN with Diffusion Model towards High-Resolution and High-Quality Weather Forecasting. Proceedings of the AAAI Conference on Artificial Intelligence 37(1):322–330, DOI 10.1609/aaai.v37i1.25105, URL https://ojs.aaai.org/index.php/AAAI/article/view/25105
17. Cohn DA, Ghahramani Z, Jordan MI (1996) Active Learning with Statistical Models. DOI 10.48550/arXiv.cs/9603104, URL http://arxiv.org/abs/cs/9603104
18. Constantine PG (2015) Active Subspaces: Emerging Ideas for Dimension Reduction in Parameter Studies. Society for Industrial and Applied Mathematics, Philadelphia, PA, DOI 10.1137/1.9781611973860, URL http://epubs.siam.org/doi/book/10.1137/1.9781611973860
19. Cousins W, Sapsis TP (2014) Quantification and prediction of extreme events in a one-dimensional nonlinear dispersive wave model. Physica D: Nonlinear Phenomena 280-281:48–58, DOI https://doi.org/10.1016/j.physd.2014.04.012
20. Cox T, Cox M (2000) Multidimensional Scaling, 0th edn. Chapman and Hall/CRC, DOI 10.1201/9780367801700, URL https://www.taylorfrancis.com/books/9781420036121
21. Davison ML (1992) Multidimensional scaling. xiv, 242 p., Krieger Pub. Co., Malabar, Fla., URL https://catalog.hathitrust.org/Record/009131389
22. Dennis JM, Edwards J, Evans KJ, Guba O, Lauritzen PH, Mirin AA, St-Cyr A, Taylor MA, Worley PH (2012) CAM-SE: A scalable spectral element dynamical core for the Community Atmosphere Model. The International Journal of High Performance Computing Applications 26(1):74–89
23. Farazmand M, Sapsis TP (2019) Extreme Events: Mechanisms and Prediction. Applied Mechanics Reviews 71(5):050801, DOI 10.1115/1.4042065, URL https://asmedigitalcollection.asme.org/appliedmechanicsreviews/article/doi/10.1115/1.4042065/629878/Extreme-Events-Mechanisms-and-Prediction
24. Fiedler T, Pitman AJ, Mackenzie K, Wood N, Jakob C, Perkins-Kirkpatrick SE (2021) Business risk and the emergence of climate analytics. Nature Climate Change

11(2):87–94, DOI 10.1038/s41558-020-00984-6, URL https://www.nature.com/articles/s41558-020-00984-6, publisher: Nature Publishing Group
25. Gal Y, Ghahramani Z (2016) Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: international conference on machine learning, PMLR, pp 1050–1059
26. Ghattas O, Willcox K (2021) Learning physics-based models from data: perspectives from inverse problems and model reduction. Acta Numerica 30:445–554, DOI 10.1017/S0962492921000064, URL https://www.cambridge.org/core/product/identifier/S0962492921000064/type/journal_article
27. Golaz JC, Van Roekel LP, Zheng X, Roberts AF, Wolfe JD, Lin W, Bradley AM, Tang Q, Maltrud ME, Forsyth RM, Zhang C, Zhou T, Zhang K, Zender CS, Wu M, Wang H, Turner AK, Singh B, Richter JH, Qin Y, Petersen MR, Mametjanov A, Ma PL, Larson VE, Krishna J, Keen ND, Jeffery N, Hunke EC, Hannah WM, Guba O, Griffin BM, Feng Y, Engwirda D, Di Vittorio AV, Dang C, Conlon LM, Chen CCJ, Brunke MA, Bisht G, Benedict JJ, Asay-Davis XS, Zhang Y, Zhang M, Zeng X, Xie S, Wolfram PJ, Vo T, Veneziani M, Tesfa TK, Sreepathi S, Salinger AG, Reeves Eyre JEJ, Prather MJ, Mahajan S, Li Q, Jones PW, Jacob RL, Huebler GW, Huang X, Hillman BR, Harrop BE, Foucar JG, Fang Y, Comeau DS, Caldwell PM, Bartoletti T, Balaguru K, Taylor MA, McCoy RB, Leung LR, Bader DC (2022) The DOE E3SM Model Version 2: Overview of the Physical Model and Initial Model Evaluation. Journal of Advances in Modeling Earth Systems 14(12):e2022MS003156, DOI 10.1029/2022MS003156, URL https://doi.org/10.1029/2022MS003156
28. Gramacy RB, Lee HKH (2009) Adaptive Design and Analysis of Supercomputer Experiments. Technometrics 51(2):130–145, DOI 10.1198/TECH.2009.0015, URL https://doi.org/10.1198/TECH.2009.0015, publisher: Taylor & Francis _eprint: https://doi.org/10.1198/TECH.2009.0015
29. Guerra N, Nelsen NH, Yang Y (2025) Learning Where to Learn: Training Distribution Selection for Provable OOD Performance. DOI 10.48550/ARXIV.2505.21626, URL https://arxiv.org/abs/2505.21626, version Number: 1
30. Guth S, Champenois B, Sapsis TP (2022) Application of gaussian process multi-fidelity optimal sampling to ship structural modeling. In: 34th Symp. on Naval Hydrodynamics, Washington, DC, June
31. Guth S, Katsidoniotaki E, Sapsis TP (2024) Statistical modeling of fully nonlinear hydrodynamic loads on offshore wind turbine monopile foundations using wave episodes and targeted CFD simulations through active sampling. Wind Energy 27(1):75–100, DOI 10.1002/we.2880, URL https://onlinelibrary.wiley.com/doi/10.1002/we.2880
32. Guth S, Mojahed A, Sapsis TP (2024) Quality measures for the evaluation of machine learning architectures on the quantification of epistemic and aleatoric uncertainties in complex dynamical systems. Computer Methods in Applied Mechanics and Engineering 420:116760
33. Haixiang G, Yijing L, Shang J, Mingyun G, Yuanyue H, Bing G (2017) Learning from class-imbalanced data: Review of methods and applications. Expert Systems with Applications 73:220–239, DOI 10.1016/j.eswa.2016.12.035, URL https://linkinghub.elsevier.com/retrieve/pii/S0957417416307175
34. Hense A, Friederichs P (2006) Wind and Precipitation Extremes in the Earth's Atmosphere. In: Dragoman D, Dragoman M, Elitzur AC, Silverman MP, Tuszynski J, Zeh HD, Albeverio S, Jentsch V, Kantz H (eds) Extreme Events in Nature and Society, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 169–187, DOI 10.1007/3-540-28611-X_8
35. Hersbach H, Bell B, Berrisford P, Hirahara S, Horányi A, Muñoz-Sabater J, Nicolas J, Peubey C, Radu R, Schepers D, Simmons A, Soci C, Abdalla S, Abellan X, Balsamo G, Bechtold P, Biavati G, Bidlot J, Bonavita M, De Chiara G, Dahlgren P, Dee D, Diamantakis M, Dragani R, Flemming J, Forbes R, Fuentes M, Geer A, Haimberger L, Healy S, Hogan RJ, Hólm E, Janisková M, Keeley S, Laloyaux P, Lopez P, Lupu C, Radnoti G, de Rosnay P, Rozum I, Vamborg F, Villaume S, Thépaut JN (2020) The ERA5 global reanalysis. Quarterly Journal of the Royal Meteorological Society 146(730):1999–2049, DOI https://doi.org/10.1002/qj.3803, URL https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.3803
36. Hüllermeier E, Waegeman W (2021) Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. Machine learning 110(3):457–506

37. Karniadakis GE, Kevrekidis IG, Lu L, Perdikaris P, Wang S, Yang L (2021) Physics-informed machine learning. Nature Reviews Physics 3(6):422–440, DOI 10.1038/s42254-021-00314-5, URL https://www.nature.com/articles/s42254-021-00314-5
38. Katsidoniotaki E, Guth S, Göteman M, Sapsis TP (2025) Reduced order modeling of wave energy systems via sequential Bayesian experimental design and machine learning. Applied Ocean Research 155:104439, DOI 10.1016/j.apor.2025.104439, URL https://linkinghub.elsevier.com/retrieve/pii/S0141118725000276
39. Kautz LA, Martius O, Pfahl S, Pinto JG, Ramos AM, Sousa PM, Woollings T (2022) Atmospheric blocking and weather extremes over the Euro-Atlantic sector – a review. Weather and Climate Dynamics 3(1):305–336, DOI 10.5194/wcd-3-305-2022, URL https://wcd.copernicus.org/articles/3/305/2022/
40. Keisler R (2022) Forecasting Global Weather with Graph Neural Networks. URL https://arxiv.org/abs/2202.07575
41. Kochkov D, Yuval J, Langmore I, Norgaard P, Smith J, Mooers G, Klöwer M, Lottes J, Rasp S, Düben P, Hatfield S, Battaglia P, Sanchez-Gonzalez A, Willson M, Brenner MP, Hoyer S (2024) Neural general circulation models for weather and climate. Nature DOI 10.1038/s41586-024-07744-y, URL https://doi.org/10.1038/s41586-024-07744-y
42. Krause A, Singh A, Guestrin C (2008) Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies. Journal of Machine Learning Research 9(8):235–284, URL http://jmlr.org/papers/v9/krause08a.html
43. Kruskal JB (1964) Multidimensional Scaling by Optimizing Goodness of Fit to a Nonmetric Hypothesis. Psychometrika 29(1):1–27, DOI 10.1007/BF02289565
44. Kruskal JB (1964) Nonmetric Multidimensional Scaling: A Numerical Method. Psychometrika 29(2):115–129, DOI 10.1007/BF02289694
45. Lakshminarayanan B, Pritzel A, Blundell C (2017) Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. URL https://arxiv.org/abs/1612.01474
46. Li H, Fernex D, Semaan R, Tan J, Morzyński M, Noack BR (2021) Cluster-based network model. Journal of Fluid Mechanics 906:A21, DOI 10.1017/jfm.2020.785
47. MacKay DJC (1992) Information-Based Objective Functions for Active Data Selection. Neural Computation 4(4):590–604, DOI 10.1162/neco.1992.4.4.590
48. Majda AJ, McLaughlin DW, Tabak EG (1997) A one-dimensional model for dispersive wave turbulence. Journal of Nonlinear Science 7(1):9–44, DOI 10.1007/BF02679124, URL https://doi.org/10.1007/BF02679124
49. MANABE S, SMAGORINSKY J, STRICKLER RF (1965) Simulated climatology of a general circulation model with a hydrologic cycle. Monthly Weather Review 93(12):769 – 798, DOI 10.1175/1520-0493(1965)093⟨0769:SCOAGC⟩2.3.CO;2
50. Materia S, García LP, Van Straaten C, O S, Mamalakis A, Cavicchia L, Coumou D, De Luca P, Kretschmer M, Donat M (2024) Artificial intelligence for climate prediction of extremes: State of the art, challenges, and future perspectives. WIREs Climate Change 15(6):e914, DOI 10.1002/wcc.914, URL https://wires.onlinelibrary.wiley.com/doi/10.1002/wcc.914
51. Mintz Y (1968) Very Long-Term Global Integration of the Primitive Equations of Atmospheric Motion: An Experiment in Climate Simulation. In: Billings DE, Broecker WS, Bryson RA, Cox A, Damon PE, Donn WL, Eriksson E, Ewing M, Fletcher JO, Hamilton W, Jerzykiewicz M, Kutzbach JE, Lorenz EN, Mintz Y, Mitchell JM, Saltzman B, Serkowski K, Shen WC, Suess HE, Tanner WF, Weyl PK, Worthington LV, Mitchell JM (eds) Causes of Climatic Change: A collection of papers derived from the INQUA—NCAR Symposium on Causes of Climatic Change, August 30–31, 1965, Boulder, Colorado, American Meteorological Society, Boston, MA, pp 20–36
52. Mohamad MA, Sapsis TP (2018) Sequential sampling strategy for extreme event statistics in nonlinear dynamical systems. Proceedings of the National Academy of Sciences 115(44):11138–11143, DOI 10.1073/pnas.1813263115, URL https://www.pnas.org/doi/full/10.1073/pnas.1813263115, publisher: Proceedings of the National Academy of Sciences
53. Mukkavilli SK, Civitarese DS, Schmude J, Jakubik J, Jones A, Nguyen N, Phillips C, Roy S, Singh S, Watson C, Ganti R, Hamann H, Nair U, Ramachandran R, Weldemariam K (2023) AI Foundation Models for Weather and Climate: Applications, Design, and Implementation. DOI 10.48550/arXiv.2309.10808, URL http://arxiv.org/abs/2309.10808, arXiv:2309.10808 [physics]
54. Murphy KP (2022) Probabilistic machine learning: an introduction. MIT press

55. Park C, Qiu P, Carpena-Núñez J, Rao R, Susner M, Maruyama B (2023) Sequential adaptive design for jump regression estimation. IISE Transactions 55(2):111–128, DOI 10.1080/24725854.2021.1988770, URL https://www.tandfonline.com/doi/full/10.1080/24725854.2021.1988770
56. Pickering E, Sapsis TP (2024) Information FOMO: The Unhealthy Fear of Missing Out on Information—A Method for Removing Misleading Data for Healthier Models. Entropy 26(10):835, DOI 10.3390/e26100835, URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11507899/
57. Pickering E, Guth S, Karniadakis GE, Sapsis TP (2022) Discovering and forecasting extreme events via active learning in neural operators. Nature Computational Science 2(12):823–833, DOI 10.1038/s43588-022-00376-0, URL https://doi.org/10.1038/s43588-022-00376-0
58. Psaros AF, Meng X, Zou Z, Guo L, Karniadakis GE (2023) Uncertainty quantification in scientific machine learning: Methods, metrics, and comparisons. Journal of Computational Physics 477:111902, DOI https://doi.org/10.1016/j.jcp.2022.111902
59. Pushkarev A, Zakharov VE (2013) Quasibreathers in the MMT model. Physica D: Nonlinear Phenomena 248:55–61, DOI 10.1016/j.physd.2013.01.003
60. Qi D, Majda AJ (2020) Using machine learning to predict extreme events in complex systems. Proceedings of the National Academy of Sciences 117(1):52–59, DOI 10.1073/pnas.1917285117, URL https://pnas.org/doi/full/10.1073/pnas.1917285117
61. Rasmussen CE, Williams CKI (2005) Gaussian Processes for Machine Learning. The MIT Press, DOI 10.7551/mitpress/3206.001.0001, URL https://doi.org/10.7551/mitpress/3206.001.0001
62. Rasp S, Lerch S (2018) Neural Networks for Postprocessing Ensemble Weather Forecasts. Monthly Weather Review 146(11):3885 – 3900, DOI 10.1175/MWR-D-18-0187.1
63. Rasp S, Thuerey N (2021) Data-Driven Medium-Range Weather Prediction With a Resnet Pretrained on Climate Simulations: A New Model for WeatherBench. Journal of Advances in Modeling Earth Systems 13(2):e2020MS002405, DOI https://doi.org/10.1029/2020MS002405, URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2020MS002405
64. Rasp S, Dueben PD, Scher S, Weyn JA, Mouatadid S, Thuerey N (2020) WeatherBench: A Benchmark Data Set for Data-Driven Weather Forecasting. Journal of Advances in Modeling Earth Systems 12(11):e2020MS002203
65. Rasp S, Hoyer S, Merose A, Langmore I, Battaglia P, Russell T, Sanchez-Gonzalez A, Yang V, Carver R, Agrawal S, Chantry M, Ben Bouallegue Z, Dueben P, Bromberg C, Sisk J, Barrington L, Bell A, Sha F (2024) WeatherBench 2: A Benchmark for the Next Generation of Data-Driven Global Weather Models. Journal of Advances in Modeling Earth Systems 16(6):e2023MS004019, DOI https://doi.org/10.1029/2023MS004019
66. Raymond C, Horton RM, Zscheischler J, Martius O, AghaKouchak A, Balch J, Bowen SG, Camargo SJ, Hess J, Kornhuber K, Oppenheimer M, Ruane AC, Wahl T, White K (2020) Understanding and managing connected extreme events. Nature Climate Change 10(7):611–621, DOI 10.1038/s41558-020-0790-4, URL https://www.nature.com/articles/s41558-020-0790-4, publisher: Nature Publishing Group
67. Robinson A, Lehmann J, Barriopedro D, Rahmstorf S, Coumou D (2021) Increasing heat and rainfall extremes now far outside the historical climate. npj Climate and Atmospheric Science 4(1):1–4, DOI 10.1038/s41612-021-00202-w, URL https://www.nature.com/articles/s41612-021-00202-w, publisher: Nature Publishing Group
68. Rolf E, Worledge TT, Recht B, Jordan M (2021) Representation Matters: Assessing the Importance of Subgroup Allocations in Training Data. In: Meila M, Zhang T (eds) Proceedings of the 38th International Conference on Machine Learning, PMLR, Proceedings of Machine Learning Research, vol 139, pp 9040–9051, URL https://proceedings.mlr.press/v139/rolf21a.html
69. Rudy SH, Sapsis TP (2023) Output-weighted and relative entropy loss functions for deep learning precursors of extreme events. Physica D: Nonlinear Phenomena 443:133570, DOI 10.1016/j.physd.2022.133570, URL https://linkinghub.elsevier.com/retrieve/pii/S0167278922002743
70. Sacks J, Welch WJ, Mitchell TJ, Wynn HP (1989) Design and Analysis of Computer Experiments. Statistical Science 4(4), DOI 10.1214/ss/1177012413, URL https://projecteuclid.org/journals/statistical-science/volume-4/issue-4/Design-and-Analysis-of-Computer-Experiments/10.1214/ss/1177012413.full

71. Sammon J (1969) A Nonlinear Mapping for Data Structure Analysis. IEEE Transactions on Computers C-18(5):401–409, DOI 10.1109/T-C.1969.222678
72. Sapsis TP (2020) Output-weighted optimal sampling for Bayesian regression and rare event statistics using few samples. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 476(2234):20190834, URL https://royalsocietypublishing.org/doi/10.1098/rspa.2019.0834
73. Sapsis TP (2021) Statistics of Extreme Events in Fluid Flows and Waves. Annual Review of Fluid Mechanics 53(Volume 53, 2021):85–111
74. Sapsis TP, Blanchard A (2022) Optimal criteria and their asymptotic form for data selection in data-driven reduced-order modelling with Gaussian process regression. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 380(2229):20210197, DOI 10.1098/rsta.2021.0197, URL https://royalsocietypublishing.org/doi/10.1098/rsta.2021.0197
75. Schneider T, Lan S, Stuart A, Teixeira J (2017) Earth System Modeling 2.0: A Blueprint for Models That Learn From Observations and Targeted High-Resolution Simulations. Geophysical Research Letters 44(24):12,396–12,417, DOI 10.1002/2017GL076101, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/2017GL076101
76. Schneider T, Behera S, Boccaletti G, Deser C, Emanuel K, Ferrari R, Leung LR, Lin N, Müller T, Navarra A, Ndiaye O, Stuart A, Tribbia J, Yamagata T (2023) Harnessing AI and computing to advance climate modelling and prediction. Nature Climate Change 13(9):887–889, DOI 10.1038/s41558-023-01769-3, URL https://doi.org/10.1038/s41558-023-01769-3
77. Seber GAF (1984) Multivariate Observations, 1st edn. Wiley Series in Probability and Statistics, Wiley, DOI 10.1002/9780470316641, URL https://onlinelibrary.wiley.com/doi/book/10.1002/9780470316641
78. Settles B (2009) Active Learning Literature Survey. Technical Report, University of Wisconsin-Madison Department of Computer Sciences, URL https://minds.wisconsin.edu/handle/1793/60660
79. Slingo J, Bates P, Bauer P, Belcher S, Palmer T, Stephens G, Stevens B, Stocker T, Teutsch G (2022) Ambitious partnership needed for reliable climate prediction. Nature Climate Change 12(6):499–503, DOI 10.1038/s41558-022-01384-8, URL https://www.nature.com/articles/s41558-022-01384-8, publisher: Nature Publishing Group
80. SMAGORINSKY J, MANABE S, HOLLOWAY JL (1965) Numerical results from a nine-level general circulation model of the atmosphere1. Monthly Weather Review 93(12):727 – 768, DOI 10.1175/1520-0493(1965)093⟨0727:NRFANL⟩2.3.CO;2
81. Song XC, Smith P, Kalyanam R, Zhu X, Adams E, Colby K, Finnegan P, Gough E, Hillery E, Irvine R, Maji A, St John J (2022) Anvil - System Architecture and Experiences from Deployment and Early User Operations. In: Practice and Experience in Advanced Research Computing, ACM, Boston MA USA, pp 1–9, DOI 10.1145/3491418.3530766, URL https://dl.acm.org/doi/10.1145/3491418.3530766
82. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research 15(56):1929–1958, URL http://jmlr.org/papers/v15/srivastava14a.html
83. Stein ML (2021) A parametric model for distributions with flexible behavior in both tails. Environmetrics 32(2):e2658, DOI 10.1002/env.2658, URL https://onlinelibrary.wiley.com/doi/10.1002/env.2658
84. Taylor MA, Cyr AS, Fournier A (2009) A Non-oscillatory Advection Operator for the Compatible Spectral Element Method. In: Allen G, Nabrzyski J, Seidel E, van Albada GD, Dongarra J, Sloot PMA (eds) Computational Science – ICCS 2009, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 273–282
85. Tomita H, Miura H, Iga S, Nasuno T, Satoh M (2005) A global cloud-resolving simulation: Preliminary results from an aqua planet experiment. Geophysical Research Letters 32(8)
86. Vodrahalli K, Li K, Malik J (2018) Are All Training Examples Created Equal? An Empirical Study. DOI 10.48550/ARXIV.1811.12569, URL https://arxiv.org/abs/1811.12569, version Number: 1
87. Yang Y, Blanchard A, Sapsis T, Perdikaris P (2022) Output-weighted sampling for multi-armed bandits with extreme payoffs. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 478(2260):20210781, DOI 10.1098/rspa.2021.0781, URL https://royalsocietypublishing.org/doi/10.1098/rspa.2021.0781

88. Zakharov V, Dias F, Pushkarev A (2004) One-dimensional wave turbulence. Physics Reports 398(1):1–65, DOI 10.1016/j.physrep.2004.04.002
89. Zakharov VE, Guyenne P, Pushkarev AN, Dias F (2001) Wave turbulence in one-dimensional models. Physica D: Nonlinear Phenomena 152-153:573–619, DOI 10.1016/S0167-2789(01)00194-4
90. Zhou X, Liu H, Pourpanah F, Zeng T, Wang X (2022) A Survey on Epistemic (Model) Uncertainty in Supervised Learning: Recent Advances and Applications. Neurocomputing 489:449–465, DOI 10.1016/j.neucom.2021.10.119, URL http://arxiv.org/abs/2111.01968
91. Zou Z, Meng X, Psaros AF, Karniadakis GE (2024) NeuralUQ: A Comprehensive Library for Uncertainty Quantification in Neural Differential Equations and Operators. SIAM Review 66(1):161–190, DOI 10.1137/22M1518189, URL https://doi.org/10.1137/22M1518189, publisher: Society for Industrial and Applied Mathematics