entropy

Article

Information FOMO: The Unhealthy Fear of Missing Out on Information—A Method for Removing Misleading Data for Healthier Models

Ethan Pickering * and Themistoklis P. Sapsis *

Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

* Correspondence: pickering@mit.edu (E.P.); sapsis@mit.edu (T.P.S.)

Abstract: Misleading or unnecessary data can have out-sized impacts on the health or accuracy of Machine Learning (ML) models. We present a Bayesian sequential selection method, akin to Bayesian experimental design, that identifies critically important information within a dataset while ignoring data that are either misleading or bring unnecessary complexity to the surrogate model of choice. Our method improves sample-wise error convergence and eliminates instances where more data lead to worse performance and instabilities of the surrogate model, often termed sample-wise "double descent". We find these instabilities are a result of the complexity of the underlying map and are linked to extreme events and heavy tails. Our approach has two key features. First, the selection algorithm dynamically couples the chosen model and data. Data is chosen based on its merits towards improving the selected model, rather than being compared strictly against other data. Second, a natural convergence of the method removes the need for dividing the data into training, testing, and validation sets. Instead, the selection metric inherently assesses testing and validation error through global statistics of the model. This ensures that key information is never wasted in testing or validation. The method is applied using both Gaussian process regression and deep neural network surrogate models.

Keywords: double descent; Bayesian sequential selection; machine learning; deep neural networks; Gaussian process regression; sample-wise error convergence; misleading data

Citation: Pickering, E.; Sapsis, T.P. Information FOMO: The Unhealthy Fear of Missing Out on Information—A Method for Removing Misleading Data for Healthier Models. *Entropy* **2024**, *26*, 835. https://doi.org/10.3390/e26100835

Academic Editor: Éloi Bossé

Received: 7 July 2024
Revised: 27 September 2024
Accepted: 27 September 2024
Published: 30 September 2024

1. Introduction

What we term, "Information FOMO", or Fear Of Missing Out, is the unhealthy tendency to use all available data for fear of missing out on information. However, more data are not always better. Data may present misleading information, or even worse, lead to overfitting and, as we show here, induce instabilities in the surrogate map. These cases may be linked to the concept of sample-wise "deep double descent" [1–3], where more data does not result in better models. However, this phenomenon can more generally be attributed to slow sample-wise convergence.

The general concept of double descent, Figure 1, refers to test errors that first undergo a descent, then an *ascent* in error, followed by a second and final descent in error. This phenomenon is observed with respect to model complexity (i.e., number of layers or layer width), training epochs, or training samples [4], and has been both theoretically appreciated and empirically observed in numerous studies [1–3,5–9]. Although all three manifestations of double descent can lead to significant errors, the first two, model complexity and training epochs, can be avoided through straightforward model parameter studies, such as "early stopping" strategies [10] in either training time [11] or model size. Training sample size, however, is nearly always fixed and often too small for sufficient cross-validation studies. As such, double descent with small and fixed samples presents a substantial threat to ML techniques in real-world applications.

Despite this threat to ML techniques, sample-wise double descent in deep models has received much less attention than model complexity and training time. This is likely due to the limitation that fixed datasets are not large enough to perform validation studies. The majority of the literature surrounding sample-wise double descent is recognized in ridgeless regression [8,12] or random features regression [13]. In ridgeless regression, [12], double descent is directly related to an instability of the solution to an ill-posed and noisy dataset when dimensions d are equal to fitting parameters n. This notion of the instability of the surrogate model is precisely what we find here.

Data-driven modeling often addresses sample-wise instabilities through appropriate ℓ_2 regularization of the model in ridge regression [8,12] or data-dependent regularizers in deep learning models [14,15]. However, the latter provides modest improvements of a few percent for the majority of presented cases. Our work follows in the vein of data-dependent regularizers, where the data themselves are seen as a component to the level of complexity of the model, to eliminate instabilities of the surrogate model and improve error convergence.

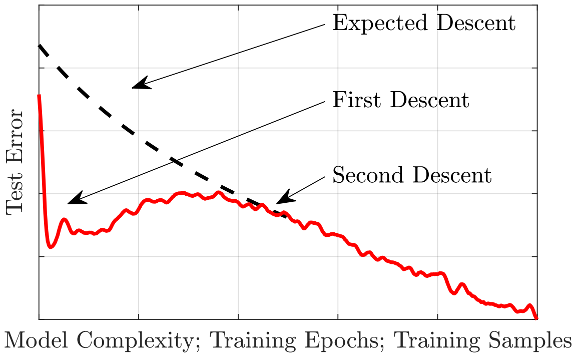


Figure 1. Double Descent does not follow the expected descent of modern ML techniques. Modern ML expects test error to decrease with model complexity, training epochs, and training samples, yet, in practice, the descent is not monotonic.

While not applied to concepts of sample-wise double descent, there are several approaches for selecting subsets of training data to reduce training cost on large datasets that draw parallels to our approach. Random uniform sampling [16,17], importance sampling [18–22], and adaptive importance sampling [23–25] reduce neural network training costs by selecting only a fraction of the data for training. Adaptive importance sampling is closest to our approach, using probability distributions to define sample importance and update the training set at each iteration [26]. While quite effective, these approaches are designed for efficiently updating and training the network weights, not for neglecting unnecessarily complex data or improving the stability of a general surrogate model. Our application considers not only neural networks, but also Gaussian processes, which do not undergo stochastic gradient descent for training.

In Section 2.4 we propose a method, inspired by Bayesian experimental design, for eliminating double descent through identifying, and ignoring, data that bring unnecessary complexity to the model. This is conducted by iteratively selecting a subset of data for improving the model by scoring the dataset with respect to its likely information gain and predictive uncertainty. This idea allows us to ignore data whose predictive uncertainty and likely information gain are small. We find that doing so brings substantial improvements to the convergence of test mean-squared error (MSE), as well as the often unappreciated log-PDF error, which emphasizes accurate prediction of extreme events.

The method also converges without the need for testing or validation data while only using a fraction of the data. As the method's selection criteria ignores much of the data, convergence in the training set directly implies convergence in test error. The approach also demonstrates that only a small subset of the data, e.g., 1 out of every 20 samples in one example, provide useful information. Together, the unnecessary need for splitting data into training, testing, and validation sets and the minimal selection of data for optimal error properties suggest this approach may unlock the use of many small and fixed datasets throughout various applications.

These findings are demonstrated first in Section 3.1 on a simple, nonlinear $1D$ function, with a tunable extreme event parameter, approximated by Gaussian process (GP) regression that presents a visual example of sample-wise double descent. To our knowledge, sample-wise double descent has yet to be observed for GP regression and our tuning from non-extreme to extreme behavior underscores how data can induce instabilities in the surrogate map. We then extend this approach in Section 3.2 to a larger and significantly more complex problem with 20,000 training samples and Deep Neural Networks (DNNs) taking the role of a surrogate model (specifically, DeepONet [27]). Section 4 concludes our study discussing the generality and implications of the approach to limited and sparse datasets as well as any Bayesian or ensemble surrogate model.

2. Methods

2.1. General Map Approximation Task

We wish to predict an output, y, from an observed input, x , by approximating the underlying map, $y = f(x)$ via a surrogate model. Learning this map may require substantial data depending on the complexity and dimension of the input space (e.g., when x is a multi-dimensional vector of inputs, $\mathbf{x}$). We seek to accomplish this learning task through the identification of only a subset of available data for training. We quantify performance by measuring errors in both the normalized mean square error (MSE) and the log-PDF error. MSE is calculated between the approximated output, μ, and true value, y, with n samples and is calculated as,

$$e_{\text{MSE}} = \frac{\sum_{i=1}^{n}(y_i - \mu_i)^2}{\sum_{i=1}^{n} y_i^2}, \tag{1}$$

while the log-PDF error is calculated by,

$$e_{\log-\text{PDF}} = \int |\log_{10} p_{\mu}(y) - \log_{10} p_f(y)| \mathrm{d}y, \tag{2}$$

where both the true PDF, $p_f(y)$, and approximated PDF, $p_\mu(y)$, are found via a kernel density estimator. Details on computing errors and application to each presented use case are provided in the appendix, Appendix A.3.

We test two surrogate models on two approximation tasks of differing dimensionality and complexity. A Gaussian Process regression (GP) surrogate approximates a simple one-dimensional map with heavy-tailed statistics and a Deep Neural Network (DNN) surrogate approximates a more complex dispersive nonlinear wave model that takes the form of a one-dimensional partial differential equation. GPs and DNNs are detailed in the Appendix A.2, while the high-level details of each task are presented next.

2.2. $1D$ *Piece-Wise Nonlinear Function*

Figure 2a presents the scalar, $1D$ piece-wise nonlinear function of varying degrees (see Appendix A.1.1 for more details) with a linear core and nonlinear edges to emulate rare dynamical instabilities initiated at high magnitudes (varied by a nonlinear coefficient, $C = 0, 5, 20, 50$). The input variable, x, is a Gaussian random variable with mean 0 and standard deviation of 1, whose probability distribution function (PDF), p_x, is shown in Figure 2b. We may then calculate the PDF of the function, with output y, via a standard weighted Gaussian kernel density estimators (KDE) as,

$$p_f(y) = \text{KDE}(\text{data} = y, \text{weights} = p_x(x)), \tag{3}$$

which displays a Gaussian core and heavy tails in Figure 2c, indicating rare and extreme events. The PDF of the surrogate model with output μ is calculated similarly, replacing y for μ.

To emulate experimental datasets, where the independent variable is controlled rather than stochastic, the training data are sampled from a uniform distribution from $x \in [-6, 6]$. While this may appear counter-intuitive, sampling from p_x would constitute the "in the wild" case and require an improbably large training dataset to span the domain. We do not test this case here. Experimentation of stochastic systems often aims to explore the entire domain of possibilities, rather than repeating the most likely scenarios. For example, resources on large-scale weather simulations are often spent to span the input space of parameters, rather than resample likely scenarios, to quantify weather risk. This sampling approach is also used for our second test case, the dispersive nonlinear wave model.

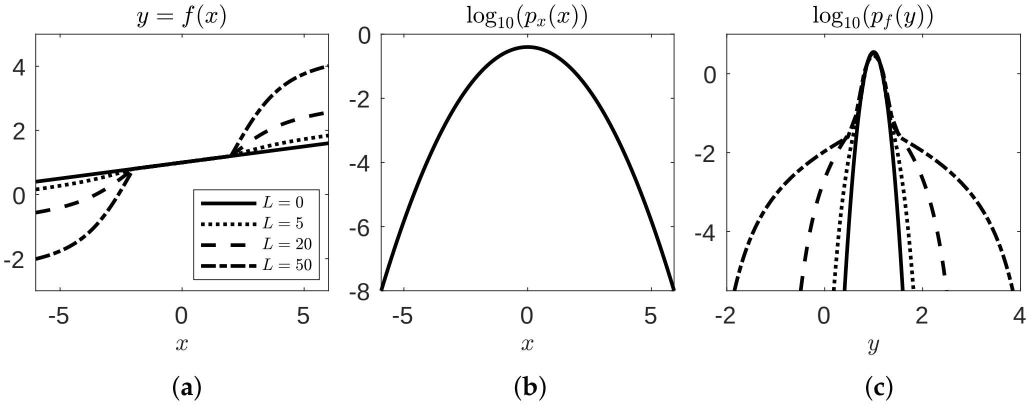


Figure 2. (**a**) the true nonlinear solutions $y = f(x)$, with respect to random variable x for nonlinear coefficients of $C = 0, 5, 20, 50$, (**b**) the Gaussian PDF of x, and (**c**) the non-Gaussian PDF of the function with respect to the response variable, y, displaying heavy tails for each nonlinear case.

2.3. Dispersive Nonlinear Wave Model

Our second task is based on a dispersive, nonlinear wave model that has the form of a partial differential equation originally proposed by Majda, McLaughlin, and Tabak (MMT) [28] for the study of one-dimensional (i.e., one spatial dimension) wave turbulence. In contrast to the previous $1D$ task, the MMT model evolves a wave, $u(x_L, t)$ over space ($1D$ spatial variable x_L) and time (t), resulting in an infinite dimensional task that is reduced to the dimension of the chosen computational grid (i.e., $n_{x_L} \times n_t$). The initial conditions of the wave, $u(x_L, 0)$, provide the input values for the model. They are randomly chosen from an $8D$ subspace: $u(x_L, 0) = \mathbf{x}\mathbf{\Phi}(x_L)$, where $\mathbf{x}$ represents the random coefficients (an $8D$ vector) that are assumed to follow a known distribution. The output variable of interest, which we wish to map the initial wave conditions to, is the maximum future wave height $|Re(u(x_L, t = \tau))|_\infty$, where τ is a finite time horizon and set at $\tau = 50$ for this study (see Appendices A.1.2 and A.1 for details on MMT and the wave height map).

2.4. FOMO Algorithm

The "FOMO" algorithm aims to identify a subset of a dataset for training that results in improved error metrics. For each of the tasks and surrogate models, we vary the size of the dataset to search as well as the input variable distributions. Details on each of the datasets used to test the FOMO algorithm are detailed and discussed in the results, Section 3.

The heart of the FOMO algorithm is the acquisition function used to select the performance-improving subset. We use an acquisition function proposed by Blanchard and Sapsis [29] to optimally learn the log-PDF for extreme events through sequential experimentation (i.e., Bayesian Experimental Design (BED)). The acquisition function,

$$a(\mathbf{x}) = w(\mathbf{x})\sigma^2(\mathbf{x}), \tag{4}$$

is the product of the likelihood ratio,

$$w(\mathbf{x}) = \frac{p_{\mathbf{x}}(\mathbf{x})}{p_{\mu}(\mu(\mathbf{x}))}, \tag{5}$$

where $p_{\mathbf{x}}$ and p_{μ} are the input and approximated output PDFs, respectively, and $\sigma^2(\mathbf{x})$ is the predictive variance of the surrogate model. The likelihood ratio, or information metric, ranks input-output (I-O) pairs with respect to their potential for contributing information to efficiently learn the underlying probability distribution of the map. This metric becomes optimal when p_{μ} approaches p_f in the log-PDF metric. The acquisition function combines this metric with the predictive variance, identifying input regions where the model is uncertain of the output. This balances the acquisition of samples that resolve the error in the map (i.e., MSE) and the map's output distribution (i.e., log-PDF error) [29].

The "FOMO" sequential selection method for a fixed dataset is outlined in Algorithm 1 with the following steps. (1) Initialize with a small subset, n_{init}, I-O pairs from the dataset and train the surrogate model. For reference, the GP and DNN tasks are initialized with 8% and 0.1% of the datasets, respectively. The initial samples may be chosen randomly or through some proposed sampling bias. (2) A predictive variance emitting surrogate model (e.g., GP or an ensemble of DNNs) trains on the selected dataset and emits a predictive mean, μ, and variance, σ^2, for all observed I-O pairs. The new surrogate model is then leveraged to construct the approximate output PDF (the output PDF is approximated by 10^7 Latin Hypercube samples (LHS) whose output is predicted by the surrogate model). The acquisition values are calculated and the highest scoring I-O pairs of batch size, n_b, amongst the whole dataset, are selected. (3) Augment training dataset with selected I-O pairs and, as *all* I-O pairs are given acquisition scores, duplicates, i.e., a current training I-O sample that is chosen again, are removed. (4) Retrain the surrogate model on the augmented dataset and repeat (2)–(4) until n_{iter} is achieved.

Algorithm 1 FOMO sequential selection algorithm for fixed data

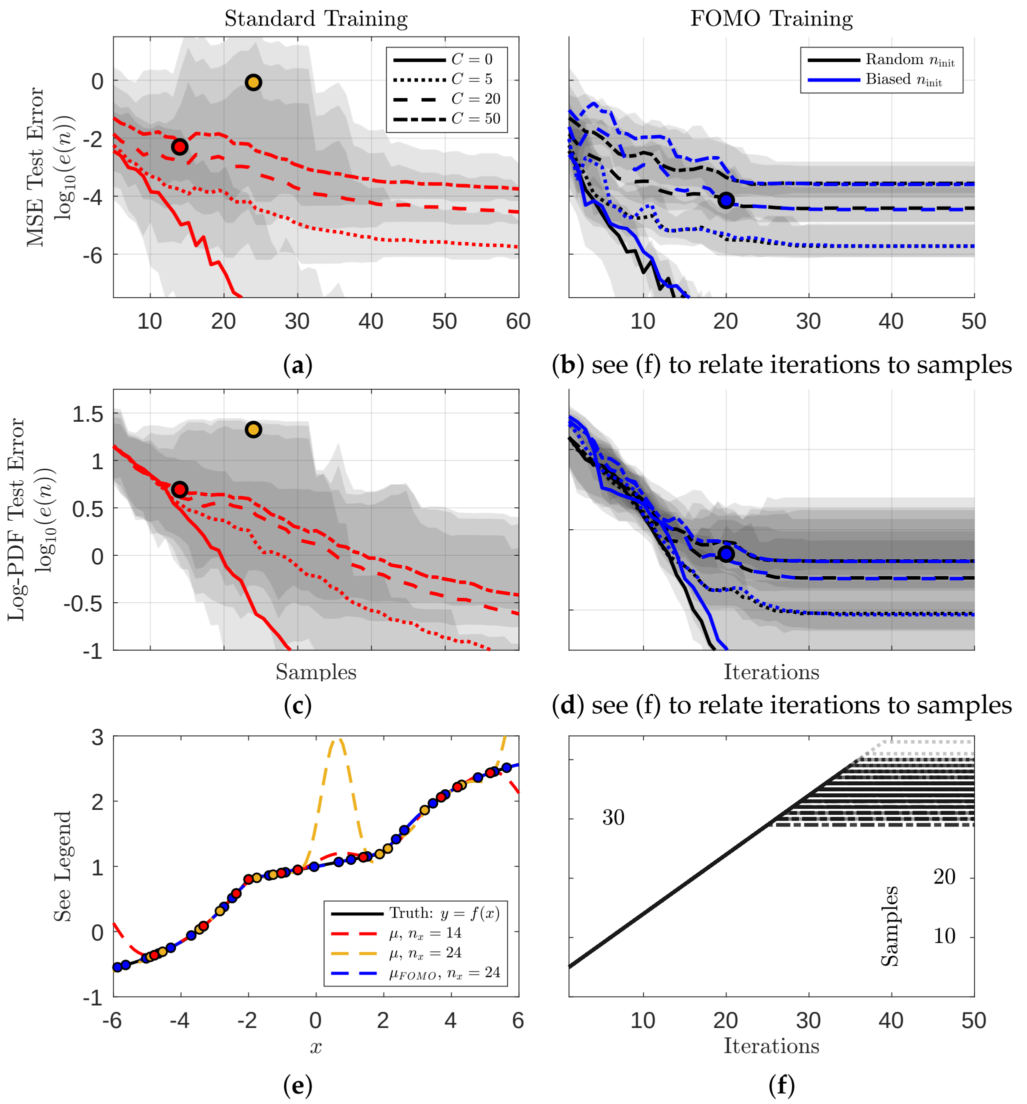
```
1: Initialize: Select $n_{\text{init}}$ I-O pairs, randomly or otherwise, and train surrogate model.
   for $n = 1$ to $n_{\text{iter}}$ do
2:     Calculate acquisition values, $a(\mathbf{x})$, for the entire dataset, select $n_b$ (i.e., batch size) highest scoring I-O pairs.
3:     Augment training dataset and remove duplicate I-O pairs.
4:     Retrain surrogate model on augmented dataset.
5: end for
6: return Final surrogate model and training dataset.
```

3. Results

We show that the FOMO approach improves error convergence in both mean-square error (MSE) and log-PDF error and eliminates model instabilities for GP and DNN surrogate models in both low and high dimensional problems, without the need for test or validation data.

3.1. 1*D Example with Gaussian Process Regression*

In Figure 3, we test and observe training errors for 100 independent experiments for our two training approaches approximating the function defined by (A1)–(A3), shown in Figure 2a with a GP surrogate model, detailed in Appendix A.2.1, given n samples (sampled from a uniform distribution), ranging from 5 to 60, of observed input and output data, $\mathcal{D} = \{x_i, y_i\}_{i=1}^{n}$. Figure 3a,c present the normalized MSE and log-PDF test errors (see Appendix A.3 for error calculation details), respectively, against the number of samples in standard training, while Figure 3b,d report the errors versus iteration using the FOMO algorithm initialized with random samples and biased I-O samples. In each figure the median error is highlighted with shaded regions indicating the minimum and maximum errors.



(**a**)

(**b**) see (f) to relate iterations to samples

(**c**)

(**d**) see (f) to relate iterations to samples

(**e**)

(**f**)

Figure 3. GP FOMO model improves error convergence, is superior to early stopping, and converges without testing or validation data. The mean normalized MSE and log-PDF errors (min and max values shaded) of 100 experiments with randomly chosen data samples (**a**,**c**), and with FOMO sequential selections (**b**,**d**) over four nonlinear coefficients $C = 0, 5, 20, 50$ and two initializations, random and biased. (**e**) provides a comparison of the approximated and true solution for the errors denoted in (**a**–**d**,**f**) is the number of chosen data samples by iteration for 100 independent sequential searches from (**b**,**d**).

Poor error convergence, i.e., sample-wise double descent, is observed for both metrics under standard training using GPs for the simple $1D$ problem for nonzero nonlinear coefficients, while the FOMO approach clearly improves convergence. For this $1D$ case, we will forego detailed early-stopping/training/testing/validation implications for the following DNN example and focus on the $C = 20$ 1D representations of sample-wise double descent and the instabilities experienced by the surrogate model. We highlight 3 points in plots a–d. The red, gold, and blue points present representative examples of early stopping point, double descent, and FOMO training, respectively, from the *same* training set. Figure 3e provides the mean solution, $\mu(x)$, for each and their associated training samples. While the red, $n_x = 14$, solution is relatively accurate, the addition of 10 more samples leads to an unstable solution at $n_x = 24$. The FOMO approach avoids this instability and presents a substantially superior model solution by sample 10, though $n_x = 24$ is shown to compare against the double descent solution. This observation of model instabilities through standard training is also observed for the other two nonlinear coefficients, $C = 50$ and $C = 5$, with more and less severity, respectively. These coefficients lead to heavier tails and stronger extreme events, while $C = 0$ emits no extreme events and already possesses fast error convergence with standard training.

Differences in initialization, biased or random, do not change FOMO performance, as seen in Figure 3b,d. Our investigation of biased samples aims to ask whether the initial subset of chosen data alters the FOMO trajectory. The biased I-O samples are chosen as the n_{init} highest likelihood ratios after training the surrogate model with *all* data to approximate the output PDF. This assumes the "all-data-trained" model provides a reasonable, but not necessarily converged, approximation of the output statistics. Other initial approximations of the output PDF could also be used to determine initial values. Regardless, the random initialization performs similarly to the biased approach. This observation reiterates that the FOMO algorithm seeks to build a subset that does not trigger model instabilities. As long as the initialization does not trigger a model instability, it will not appreciably change the FOMO trajectory. Initializing with a small training subset, when compared to model capacity or the input space, helps ensure stability of the first iteration.

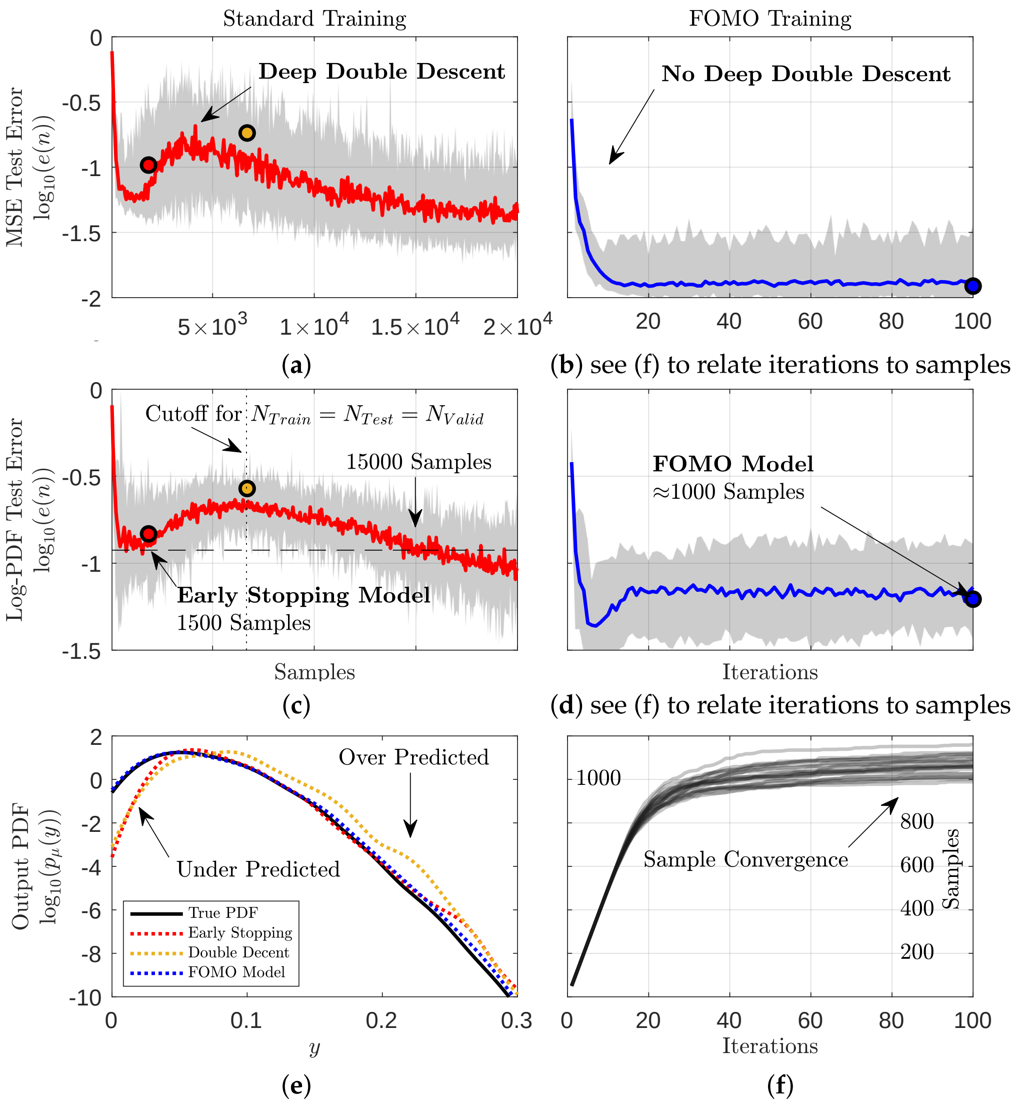
Figure 3f, presents another key feature of the FOMO algorithm, choosing only about half of the available samples. The algorithm only adds samples that bring stable improvements to the model and ignores those that do not. We now expand on this idea in the higher-dimensional case using DNNs.

3.2. *Dispersive Nonlinear Wave Model with Deep Neural Networks*

We move to a more complex case that seeks to learn a dispersive nonlinear wave model that has the form of a one-dimensional partial differential equation originally proposed by Majda, McLaughlin, and Tabak (MMT) [28] for the study of 1D wave turbulence. While GPs could be used as a surrogate, a companion study [9] showed DNNs are superior for the complexity of this problem and are used here, see Appendix A.2.2 for DNN details. Additionally, this example requires many training samples, at least at an order of magnitude of 1000 or more, and are computationally intractable for standard off-the-shelf GP applications compared to off-the-shelf DNNs.

Similar to the previous example, Figure 4 presents the training errors of 25 independent experiments of both training approaches for approximating the underlying map with a DNN given n samples (sampled via Latin Hypercube sampling, LHS), ranging from 50 to 20,000, of observed input and output data, $\mathcal{D} = \{\boldsymbol{u}_i(x_L), y_i\}_{i=1}^{n}$. Figure 4a,c present the normalized MSE and log-PDF test errors, respectively, against the number of samples, while Figure 4b,d report the errors versus iteration using the FOMO algorithm. In Figure 4a,c, early stopping only considers 1500 samples and a similarly accurate error is not observed again until an order of magnitude higher, at 15,000 samples. Even at the early stopping error, Figure 4e shows a drastic underprediction of the most probable states. More concerning, if an unconservative splitting of the 20,000 samples is taken, i.e., 6700 samples for training, testing, and validation, then full training leads to errors near the peak of double descent. In addition to the early stopping errors, the peak of double decent also leads to extreme over predictions of high magnitude events, shown in Figure 4e.

The FOMO results shown in Figure 4b,d, and found via Algorithm 1, do not suffer from these difficulties. Instead, the FOMO approach provides the most accurate model (see Figure 4e), does not undergo sample-wise double descent, converges quickly, and only uses 1/20th of the data to achieve these results. We stress the latter observation as it is a direct result of Algorithm 1, which, at each iteration, only selects data that specifically eliminate uncertainty in the model or provides essential information. Consequently, data that do not bear beneficial information to the model are deemed unnecessary and ignored.



(**a**)

(**b**) see (f) to relate iterations to samples

(**c**)

(**d**) see (f) to relate iterations to samples

(**e**)

(**f**)

Figure 4. DNN FOMO model eliminates double descent, is superior to early stopping, and converges without testing or validation data. The mean normalized MSE and log-PDF errors (with min and max values shaded) of 25 experiments with randomly chosen data points on a Latin Hypercube (**a**,**c**), and with sequential selections (**b**,**d**), with biased initialization. (**e**) provides a comparison of the predicted and true output distributions for three errors denoted in (**a**–**d**,**f**) the number of chosen data samples by iteration for 25 independent sequential searches from (**b**,**d**).

Figure 4f demonstrates the ability of the algorithm to ignore data and converge to a set of optimal training data that, critically, emits converged error metrics in both MSE and log-PDF error. At each iteration, the algorithm finds a user-specified n_b samples (here $n_b = 50$) that provide the greatest information gain amongst the entire dataset. This means that even those samples that are amongst the training set are considered. As more samples are added to the training set, the algorithm finds that the remaining samples provide *less* information than *known* samples, permitting an ability to disregard those remaining samples. The convergence of the training set results in a convergence of information, which ultimately leads to a convergence in the test error (Figure 4b,d). This implies that the FOMO approach does not require data to be split into training, testing, and validation groups. Rather, all pertinent information may be extracted from the entire dataset for the most accurate model.

Optimal data selection through a dynamic partitioning of samples as informative or unnecessary. Figure 5 demonstrates how Algorithm 1 selects the most informative samples while ignoring the rest. The y-axis denotes our metric for information, the likelihood ratio [29], while the x-axis presents the predictive variance, $\sigma^2(\mathbf{x})$, found amongst the ensemble of the independently weight-initialized DNN predictions. The product of these two quantities is the acquisition function proposed by Blanchard and Sapsis [29], Sapsis [30], $a(\mathbf{x}) = w(\mathbf{x})\sigma^2(\mathbf{x})$. As shown in Sapsis and Blanchard [31], the adopted acquisition function balances information with the uncertainty that guarantees optimal convergence in the context of Gaussian Process Regression. Here, it is employed as a measure to optimally acquire samples for training the DNN. In effect, this approach allows us to measure model risk versus model capacity [2,32–35].

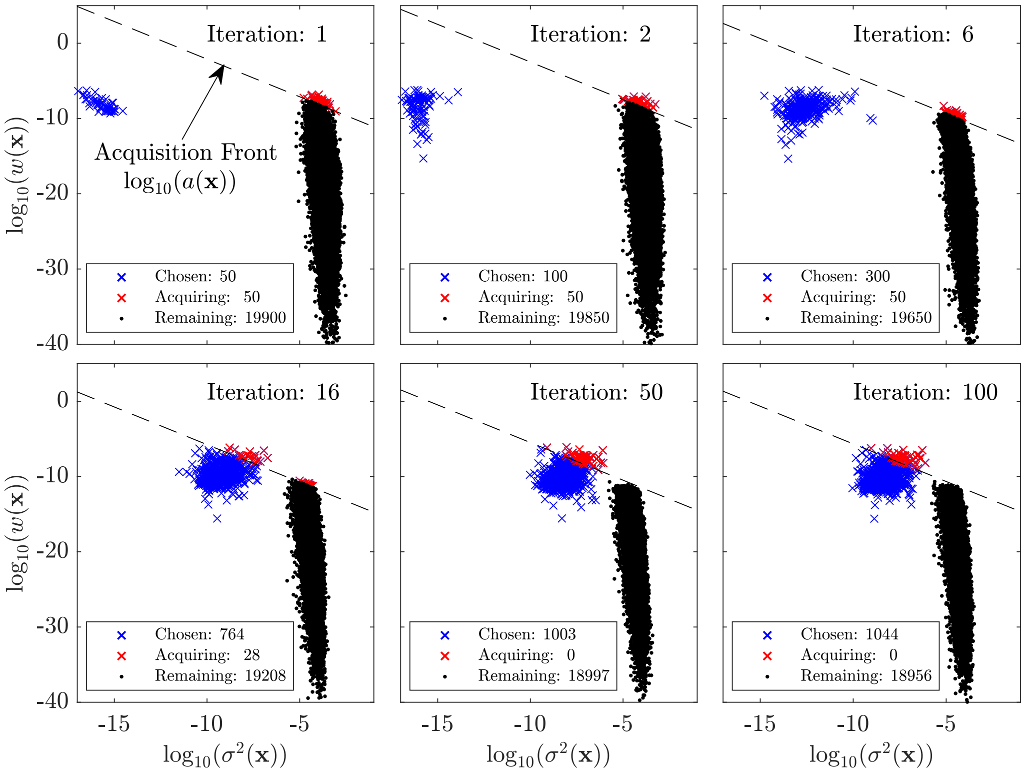
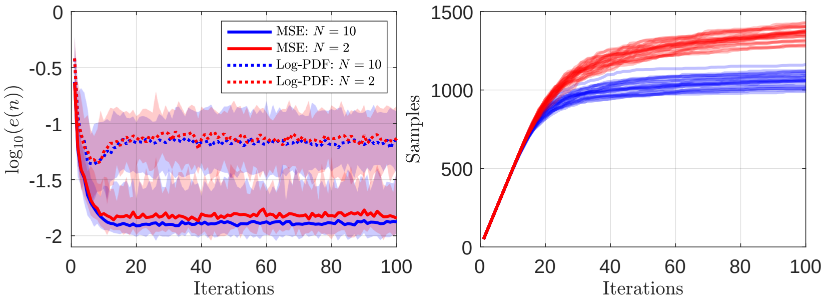


Figure 5. Necessary/Informative and unnecessary/misleading datasets show clear separation. A representative example of the iterative selection process, where data are sequentially acquired. The acquisition front indicates the acquisition score of the 50th highest scoring I-O sample at each iteration. Those above are acquired and those already chosen remain in the training set.

Walking through the iterations of Figure 5, we can see how the approach dynamically partitions the data into informative and unnecessary categories. Starting at iteration 1, only 50 samples, denoted in blue are used to train the model. Unsurprisingly, such little data allow each DNN to train these samples to machine precision and gives a predictive uncertainty amongst the training samples of $\log_{10}(\sigma^2_{Chosen}(\mathbf{x})) \approx -16$, while the predictions of the remaining samples are 12 orders of magnitude larger, as expected, and vary widely in the likelihood ratio metric. We can then acquire the largest n_b samples, with respect to the acquisition function, retrain the model with the addition of the new samples, and repeat the process. Here, we chose $n_b = 50$ and the 50th point sets which we define as the "acquisition front", denoting a line of constant acquisition value between I-O samples the model wishes to acquire and those it does not. Iterations 2 and 6 show that as more data are acquired into the chosen/training set, the uncertainty in the chosen set increases, while the uncertainty in the remaining set decreases. The latter observation is simply due to the increased training data that reduce the generalization error and thus uncertainty. The former observation, although expected, has significant implications for the FOMO method. As the training set increases, the degree of over-parameterization of each DNN reduces and discrepancies between the DNN training errors increase. This behavior is critical to identifying when the DNN parameterization becomes stressed and susceptible to instabilities related to slow error convergence or double descent. This is shown in the following iterations.

Iterations 16, 50, and 100, demonstrate the ability of the model to ignore data when the DNN parameterization has reached a critical threshold. At iteration 16 the uncertainty of the chosen data has substantially increased, while the information contained in the remaining data has decreased, leading to an acquisition front that intersects both the chosen and remaining sets. For this iteration, only 28 of the 50 samples are novel, while 22 samples already exist in the chosen data. This means that the model begins to recognize that it already contains the most pertinent information and its uncertainty in these data is approaching an unacceptable level. By iteration 50 the model has begun to entirely ignore the remaining data, as the model is sufficiently uncertain about its own, more informative training data and does not see a reason to add further stress into the model. Over the next 50 iterations, the stochastic nature of the DNN training permits 43 new samples to join the chosen dataset. However, considering the average of acquisition is less than one for the 50 potential acquisition samples at each iteration, the algorithm is easily converged by iteration 50.

Shallow ensembles are cheap and perform well. An ensemble of DNNs, differed only in their weight initialization [36], are employed to calculate the predictive uncertainty. Although the validity of such approaches is hotly debated, Wilson and Izmailov [37] and Pickering and Sapsis [38] have argued that DNN ensembles provide a very good approximation of the posterior. While the previous results have all been for an ensemble of $N = 10$ DNNs, Figure 6 shows that shallow ensembles of only $N = 2$ perform nearly as well at 1/5 the computational cost. Figure 6a shows that the median and range of both the MSE and log-PDF errors are nearly identical, with small advantages going to the larger ensemble. The greatest difference between the two is the number of acquired samples in Figure 6b, where $N = 2$ chooses approximately 25% more samples. The surprising ability of shallow ensembles of $N = 2$ to perform well in active learning schemes was also observed in Pickering et al. [9].



(**a**) (**b**)

Figure 6. Shallow ensembles perform well. (**a**) The median MSE and log-PDF errors (with min and max values shaded) of 25 experiments using DNN ensemble sizes of $N = 10$ and $N = 2$. (**b**) the number of chosen data samples by iteration for 25 independent sequential searches for both $N = 10$ and $N = 2$.

4. Discussion

Slow sample-wise error convergence, or double descent, a traditionally unavoidable phenomenon for fixed datasets, is eliminated in both the MSE and log-PDF errors for both GP and DNN surrogate models. This Bayesian-inspired approach ignores data that bring unwarranted instability to the chosen model that otherwise results in an increase in test/generalization error with increased sample size, an instability that is often unrecoverable. For real-world datasets, especially where data collection has concluded or is expensive to collect, removing sample-wise double descent is critically important. This method not only substantially reduces the errors in normalized MSE from a median of 20%, to 1%, but also the log-PDF error by four-fold. An often unappreciated metric, the log-PDF error appropriately weights high-magnitude, rare phenomena that ML methods struggle to accurately predict [9,39].

The approach is model-agnostic and statistically driven, requiring a dynamic interaction between the model and the data. The statistical metrics used to select data means that regardless of the chosen surrogate model (e.g., DNN, Gaussian process regression, etc.) or parameters (e.g., layers, neurons, training epochs, activation functions, kernels, etc.), the model and the dataset have an opportunity to "discuss" the inherent deficiencies of the model and the short comings of the data. As only statistics are used to guide the iterative data selection process, the approach only requires that the surrogate model of choice emit a mean and variance prediction. Although rigorous predictions of the quantities, such as Gaussian process regression, are clear surrogate model candidates, our work here shows that simple, ensemble DNNs provide sufficient predictions for success.

Shallow DNN ensembles are simple in implementation, scalable, and computationally tractable. Ensemble DNNs are simple to implement. They only require that DNN weights are randomly initialized, the default setting in all DNN architectures. Despite this simple approach, we find that shallow ensembles of DNNs, even just two, provide sufficient predictive uncertainties to eliminate double descent. DNNs also provide ideal scaling in data size and input parameters, unlike more rigorous methods such as Gaussian processes. Finally, although the iterative process requires many new DNNs to be trained, the DNNs are trained on only a small subset of the data, 1/20th here, during the selection process. Together, ease of implementation, modest scaling, shallow ensembles, and reduced training data are compelling for real-world application.

Convergence of the approximated output PDF ensures convergence of training samples and removes the need for testing and validation sets. The surrogate model output PDF, p_μ directly, and dynamically, informs the information metric (i.e., likelihood ratio) for acquiring new samples. Once the surrogate output PDF converges, the information metric does as well. Consequently, the acquisition front becomes set, only the predictive variance provides an avenue for acquiring more samples, and the remaining samples above this threshold are acquired, ending the selection. While it is traditional to consider the point-wise MSE in validation and testing, consisting of 2/3 of the data, the surrogate output PDF is approximated through 10^6, or more, test samples. The surrogate output PDF provides a global measure of the model generalization error, rather than a sparse and limited pointwise comparison. Thus, we argue that the cost of losing information to a local comparison tool, by setting aside 2/3 or more of available data, compared to a global metric is a far greater and unnecessary risk.

An optimal sparse representation of the underlying map is preferred to a plentiful, but arbitrary representation, opening the door to numerous datasets with various deficiencies, whether that be from unnecessary or limited data. Although the likelihood ratio is likely appropriate for most physical systems, other choices may be optimal for different applications. Further work is also necessary to extend the method's performance in pathological cases and sufficiently noisy data.

Sample-wise double descent in the nonlinear context appears to have a fundamentally different meaning compared to traditional discussions of double descent. While we do not provide any theoretical findings here, the observation of sample-wise double descent for Gaussian Process regression does not follow current reasoning. Double descent is typically associated with a crossover from under- to over-parameterization of DNNs. However, a GP in $1D$ has only two tunable parameters (three if noise is considered), meaning the under- to over-parameterization occurs over two samples, yet double descent is observed from 5 to 30 samples. From our observations of both GPs and DNNs, we believe double descent is due to the underlying map and the distribution of sample data. If the map is sufficiently complex, i.e., nonlinear, and data are not appropriately distributed in the input space, as is the case when randomly partitioned, then the surrogate model may possess instabilities. Of course, in the $1D$ case, it is trivial to propose a uniform distribution, but in higher-dimensional cases, such as the MMT case here, a sufficient uniform distribution of samples is infeasible. The FOMO method circumvents this by finding the distribution of fixed samples that best recover the global, output PDF, which includes both the Gaussian core and heavy tails where nonlinearities lie.

Instabilities of the surrogate model are induced by the underlying map complexity. Specifically, the complexity refers to the degree to which extremes and heavy-tailed statistics are present in the underlying map. By varying a tunable nonlinear coefficient, we show that sample double descent is not present for a non-extreme Gaussian map while double descent becomes increasingly apparent for maps with nonlinearities, extremes, and heavy tails. As more nonlinearities or extremes are present, the surrogate models become more susceptible to instabilities. Using our approach, these data-induced instabilities are avoided.

Drawbacks to this approach. While the FOMO approach improves the ability to eliminate double descent, double descent may still be unavoidable without enough samples. We list a few limitations below:

1. If no subset of the samples is sufficient to characterize the map, FOMO will not improve performance.
2. While the GP case provides a compelling $1D$ representation of double descent, the instability could be removed by introducing a nonzero noise term. Based on our knowledge that the problem is noiseless, this would be an incorrect application of noise, but, it would remove the instability.
3. Other regularization approaches for instabilities could be employed for DNNs, such as physical constraints or physics-based loss functions (e.g., PINNs [40]), but their efficacy for faster error convergence or sample-wise double descent is unknown. Additionally, the use of regularization approaches does not invalidate the use of the FOMO method.
4. The method assumes a known input distribution $p_{\mathbf{x}}$. This distribution could be approximated or rationalized from the input data. If incorrect, the performance would suffer. Future work is necessary to determine the sensitivity of the method with respect to errors in the input distribution.

In summary, we believe the FOMO approach is compelling for many practical problems. It mitigates sample-wise error convergence in both MSE and log-PDF, eliminates instabilities introduced by data, removes traditional train/test/validation by converging towards global statistics, and is simple and scalable via flexible off-the-shelf DNNs and shallow ensembles. Future work should continue to explore this performance in additionally challenging problems, on noisy data, and other surrogate models where sample-wise double descent is observed. While we generally discuss small datasets here, the FOMO approach could also be implemented to compress data and information for intractably large datasets. This is a tremendous challenge for domains such as climate where individual climate simulations can be on the order of terabytes to petabytes.

Author Contributions: Conceptualization, E.P. and T.P.S.; methodology, E.P.; validation, E.P.; formal analysis, E.P.; investigation, E.P.; resources, T.P.S.; data curation, E.P.; writing—original draft preparation, E.P.; writing—review and editing, E.P. and T.P.S.; visualization, E.P.; project administration, T.P.S.; funding acquisition, T.P.S. All authors have read and agreed to the published version of the manuscript.

Funding: The authors acknowledge support from AFOSR, Grant No. FA9550-23-1-0517, as well as MURI grant FA9550-21-1-0058, awarded to MIT.

Data Availability Statement: Data and code pertaining to the FOMO sequential search algorithm will be made public upon publication. Currently, the code for the GP problem is provided and executable via this Google Colaboratory link: GP FOMO Google Colab (accessed on 30 September 2024), while the DNN problem is provided through this Dropbox link: DNN FOMO Dropbox (accessed on 30 September 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Appendix A.1. Map Definition and Data

Appendix A.1.1. Nonlinear 1D Function

The piece-wise 1D equation is given by a linear function with logistic functions at each end,

$$y = \left(ax + \frac{C}{1 + e^{-k(x+2)}} - C/2 \right) / b + 1 \qquad x < -2 \tag{A1}$$

$$y = ax/b + 1 \qquad -2 \geq x \leq 2 \tag{A2}$$

$$y = \left(ax + \frac{C}{1 + e^{-k(x-2)}} - C/2 \right) / b + 1 \qquad x > 2, \tag{A3}$$

where $C = 20$, $k = 1$, $a = 1$, and $b = 10$.

Appendix A.1.2. Nonlinear Dispersive Wave Model

The underlying nonlinear system in our study is the Majda, McLaughlin, and Tabak (MMT) [28] model, a dispersive nonlinear model that also includes selective dissipation for high wavenumbers, used for studying 1D wave turbulence. It is described by

$$iu_t = |\partial_x|^{\alpha} u + \lambda |\partial_x|^{-\beta/4} \left(\left| |\partial_x|^{-\beta/4} u \right|^2 |\partial_x|^{-\beta/4} u \right) + iDu, \tag{A4}$$

where u is a complex scalar, exponents α and β are chosen model parameters, and D is a selective Laplacian. Under appropriate choice of parameters its response is characterized by extreme events which occur intermittently. Specifically, the model contains a rich set of physical phenomena, from four-wave resonant interactions that produce both direct and inverse cascades [28,41] and presents a unique utility as a physical model for extreme ocean waves, or rogue waves [42–44]. We refer the reader to [9], where identical parameters were used and to [45] for the numerical calculation technique.

In defining an input-output problem, we are not interested in quantifying the entire MMT behavior, but a specific observation that does not possess any closed-form solution. We define the input as

$$u(x, t = 0) \approx \mathbf{x}\boldsymbol{\Phi}(x_L), \qquad \forall \qquad x_L \in [0, 1) \tag{A5}$$

where $\mathbf{x} \in \mathbb{C}^m$ is a vector of complex coefficients and both the real and imaginary components of each coefficient are normally distributed with zero mean and diagonal covariance matrix Λ, and $\{\boldsymbol{\Lambda}, \boldsymbol{\Phi}(x)\}$ contains the first m eigenpairs via a Karhunen–Loeve expansion of the correlation matrix defined by the complex and periodic kernel,

$$k(x_L, x_L') = \sigma_u e^{i2\sin^2(\pi|x_L - x_L'|/P)} e^{-2\sin^2(\pi|x_L - x_L'|/P)/\ell_u^2}, \tag{A6}$$

with $\sigma_u^2 = 1$, $P = 1$, and $\ell_u = 0.35$. This gives the dimension of the parameter space as $2m$ due to the complex nature of the coefficients. For all cases, the random variable $\mathbf{x}_i$ is restricted to a domain ranging from -6 to 6, equivalent to six standard deviations in each direction from the mean. Here, we chose $m = 4$, or eight total independent and real variables. This value could easily be chosen to be higher; however, doing so requires an intractable amount of data to recover from deep double descent (see [9] for a 20D double descent example), further underscoring the threat sample-wise double descent brings to ML.

Finally, the output map is then defined as,

$$f(\mathbf{x}) = ||Re(u(x_L, t = T; \mathbf{x}))||_{\infty}, \tag{A7}$$

where $T = 20$. This map serves the purpose as a predictor for the largest incoming wave and, if trained appropriately, informs of a future dangerous rogue wave.

The training data for each of the 25 independent experiments consists of 20,000 samples distributed throughout the $8D$ parameter space with Latin Hypercube sampling. Both the traditional training and FOMO algorithm share the same 25 independent datasets of 20,000 samples.

Appendix A.2. Surrogate Models

Appendix A.2.1. Gaussian Process Regression

Gaussian process regression [46] is the "gold standard" for Bayesian design. A Gaussian process, $\bar{f}(\mathbf{x})$, is completely specified by its mean function $m(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}')$. For a dataset $\mathcal{D}$ of input–output pairs ($\{\mathbf{X}, \mathbf{y}\}$) and a Gaussian process with constant mean $m_{\mathbf{0}}$, the random process $\bar{f}(\mathbf{x})$ conditioned on $\mathcal{D}$ follows a normal distribution with posterior mean and variance

$$\mu(\mathbf{x}) = m_0 + k(\mathbf{x}, \mathbf{X})\mathbf{K}^{-1}(\mathbf{y} - m_0) \tag{A8}$$

$$\sigma^2(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - k(\mathbf{x}, \mathbf{X})\mathbf{K}^{-1}k(\mathbf{X}, \mathbf{x}) \tag{A9}$$

respectively, where $\mathbf{K} = k(\mathbf{X}, \mathbf{X}) + \sigma_\epsilon^2 \mathbf{I}$. Equation (A8) can predict the value of the surrogate model at any point $\mathbf{x}$, and (A9) to quantify uncertainty in prediction at that point [46]. Here, we chose the radial-basis-function (RBF) kernel with automatic relevance determination (ARD),

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left[-(\mathbf{x} - \mathbf{x}')^\top \mathbf{D}^{-1}(\mathbf{x} - \mathbf{x}')/2\right], \tag{A10}$$

where $\mathbf{D}$ is a diagonal matrix containing the lengthscales for each dimension and the GP hyperparameters appearing in the covariance function (σ_f^2 and $\mathbf{D}$ in (A9) are trained by maximum likelihood estimation). Additionally, training the GP in equation (A9) requires the inversion of the matrix $\mathbf{K}$. Typically performed by Cholesky decomposition, the inversion cost scales as $O(n^3)$, with n being the number of samples [46,47]. The prohibitive cost of this for large datasets (>$\mathcal{O}(10^3)$) is one reason DNNs are used for the MMT case.

Appendix A.2.2. Ensemble Deep Neural Network

The DNN implemented here is a simple feed-forward neural network, built with TENSORFLOW and using packages including DEEPXDE and DEEPONET [27]. The network consists of eight layers, 250 neurons, and is trained for 1000 epochs, under a learning rate of 0.001, ReLu activation functions, and a mean square error loss function.

For each set of training data, an ensemble of N Glorot normal randomly weight-initialized DNN models, each denoted as $\tilde{G}_n$, approximate the associated solution field y for feature inputs u. This allows us to determine the point-wise variance of the models as

$$\sigma^2(u) = \frac{1}{(N-1)} \sum_{n=1}^{N} (\tilde{G}_n(u) - \overline{\tilde{G}(u)})^2, \tag{A11}$$

where $\overline{\tilde{G}(u)}$ is the mean solution of the model ensemble. Finally, we must adjust the above representation to match the description for Bayesian design. In the case of traditional Bayesian design and GPs, the input parameters, $\mathbf{x}$, are random coefficients to a set of functions that represent a decomposition of a random function $u = \mathbf{x}\boldsymbol{\Phi}(x_L^1, \ldots x_L^m)$, where x_L^1 and x_L^m represent the spatial boundaries of the function. Thus, the DNN description for UQ may be recast as:

$$\sigma^2(\mathbf{x}) = \\ \frac{1}{(N-1)} \sum_{n=1}^{N} \left(\tilde{G}_n(\mathbf{x}\boldsymbol{\Phi}(x_L^1, ... x_L^m)) - \overline{\tilde{G}(\mathbf{x}\boldsymbol{\Phi}(x_L^1, ... x_L^m))} \right)^2. \tag{A12}$$

Appendix A.3. Test Error Computation

To compute the MSE and log-PDF test errors, we select two random sets of test samples, 10^2 for the GP example and 10^5 for the DNN example, $\mathbf{X}_{\text{pdf}}$, pulled from the input PDF $p_{\mathbf{x}}$, and $\mathbf{X}_{\text{lhs}}$, from Latin Hypercube sampling (random uniform for the GP case, $\mathbf{X}_{\text{uni}}$), where $\mathbf{X}_{\text{GP}} \in \mathbb{R}^{d \times 10^2}$ and $\mathbf{X}_{\text{DNN}} \in \mathbb{R}^{d \times 10^5}$. The normalized MSE is computed using $\mathbf{X}_{\text{pdf}}$ as input and $y_{\text{pdf}} = f(\mathbf{X}_{\text{pdf}})$ as output. The normalized MSE is then calculated as,

$$e_{\text{MSE}}(n) = \frac{\sum_{i=1}^{n} (y_{\text{pdf},i} - \mu_{\text{pdf},i})^2}{\sum_{i=1}^{n} y_{\text{pdf},i}^2} \tag{A13}$$

where $n = 10^5$. The log-PDF error uses the $y_{\text{lhs}} = f(\mathbf{X}_{\text{lhs}})$ input-output pairs and is calculated as

$$e_{\text{log}-\text{PDF}}(n) = \int |\log_{10} p_{\mu_n}(y) - \log_{10} p_f(y)| \text{d}y, \tag{A14}$$

where both the true PDF, $p_f(y)$, and approximated PDF, $p_{\mu_n}(y)$, are found via a kernel density estimator as,

$$p_f(y) = \text{KDE}(\text{data} = y_{\text{lhs}}, \text{weights} = p_{\mathbf{x}}(\mathbf{X}_{\text{lhs}})), \tag{A15}$$

and

$$p_\mu(y) = \text{KDE}(\text{data} = \mu_{\text{lhs}}, \text{weights} = p_{\mathbf{x}}(\mathbf{X}_{\text{lhs}})). \tag{A16}$$

With the exception that the GP example uses the uniform sampling, $y_{\text{uni}}, \mu_{\text{uni}}, \mathbf{X}_{\text{uni}}$. We reiterate that the purpose of the log-PDF error is to ensure that the approximation appropriately accounts for PDF tails, where rare, high-magnitude events, i.e., extreme events, live.

References

1. Opper, M. Statistical mechanics of learning: Generalization. In *The Handbook of Brain Theory and Neural Networks*; MIT Press: Cambridge, MA, USA, 1995; pp. 922–925.
2. Belkin, M.; Hsu, D.; Ma, S.; Mandal, S. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proc. Natl. Acad. Sci. USA* **2019**, *116*, 15849–15854.
3. Nakkiran, P.; Kaplun, G.; Bansal, Y.; Yang, T.; Barak, B.; Sutskever, I. Deep double descent: Where bigger models and more data hurt. *J. Stat. Mech. Theory Exp.* **2021**, *2021*, 124003.
4. Nakkiran, P. More data can hurt for linear regression: Sample-wise double descent. *arXiv* **2019**, arXiv:1912.07242.
5. Spigler, S.; Geiger, M.; d'Ascoli, S.; Sagun, L.; Biroli, G.; Wyart, M. A jamming transition from under-to over-parametrization affects generalization in deep learning. *J. Phys. A Math. Theor.* **2019**, *52*, 474001.
6. Geiger, M.; Spigler, S.; d'Ascoli, S.; Sagun, L.; Baity-Jesi, M.; Biroli, G.; Wyart, M. Jamming transition as a paradigm to understand the loss landscape of deep neural networks. *Phys. Rev. E* **2019**, *100*, 012115.
7. Advani, M.S.; Saxe, A.M.; Sompolinsky, H. High-dimensional dynamics of generalization error in neural networks. *Neural Netw.* **2020**, *132*, 428–446.
8. Hastie, T.; Montanari, A.; Rosset, S.; Tibshirani, R.J. Surprises in high-dimensional ridgeless least squares interpolation. *Ann. Stat.* **2022**, *50*, 949–986.
9. Pickering, E.; Guth, S.; Karniadakis, G.E.; Sapsis, T.P. Discovering and forecasting extreme events via active learning in neural operators. *Nat. Comput. Sci.* **2022**, *2*, 823–833.
10. Yao, Y.; Rosasco, L.; Caponnetto, A. On early stopping in gradient descent learning. *Constr. Approx.* **2007**, *26*, 289–315.
11. Heckel, R.; Yilmaz, F.F. Early Stopping in Deep Networks: Double Descent and How to Eliminate it. In Proceedings of the International Conference on Learning Representations, Vienna, Austria, 4 May 2021.
12. Nakkiran, P.; Venkat, P.; Kakade, S.M.; Ma, T. Optimal Regularization can Mitigate Double Descent. In Proceedings of the International Conference on Learning Representations, Vienna, Austria, 4 May 2021.
13. Mei, S.; Montanari, A. The generalization error of random features regression: Precise asymptotics and the double descent curve. *Commun. Pure Appl. Math.* **2022**, *75*, 667–766.
14. Wei, C.; Ma, T. Data-dependent sample complexity of deep neural networks via lipschitz augmentation. *Adv. Neural Inf. Process. Syst.* **2019**, *32* .
15. Wei, C.; Ma, T. Improved sample complexities for deep networks and robust classification via an all-layer margin. *arXiv* **2019**, arXiv:1910.04284.
16. Shalev-Shwartz, S.; Singer, Y.; Srebro, N.; Cotter, A. Pegasos: Primal estimated sub-gradient solver for svm. *Math. Program.* **2011**, *127*, 3–30.

17. Shalev-Shwartz, S.; Zhang, T. Stochastic dual coordinate ascent methods for regularized loss minimization. *JMLR* **2013**, *14*, 567–599.
18. Allen-Zhu, Z.; Qu, Z.; Richtárik, P.; Yuan, Y. Even faster accelerated coordinate descent using non-uniform sampling. In Proceedings of the International Conference on Machine Learning, PMLR, New York, NY, USA, 20–22 June 2016; pp. 1110–1119.
19. Nesterov, Y.; Stich, S.U. Efficiency of the accelerated coordinate descent method on structured optimization problems. *SIAM J. Optim.* **2017**, *27*, 110–123.
20. Needell, D.; Ward, R.; Srebro, N. Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm. *Adv. Neural Inf. Process. Syst.* **2014**, *27* .
21. Katharopoulos, A.; Fleuret, F. Not all samples are created equal: Deep learning with importance sampling. In Proceedings of the International Conference on Machine Learning, PMLR, Stockholm, Sweden, 10–15 July 2018; pp. 2525–2534.
22. Zhao, P.; Zhang, T. Stochastic optimization with importance sampling for regularized loss minimization. In Proceedings of the International Conference on Machine Learning, PMLR, Lille, France, 6–11 July 2015; pp. 1–9.
23. Csiba, D.; Qu, Z.; Richtárik, P. Stochastic dual coordinate ascent with adaptive probabilities. In Proceedings of the International Conference on Machine Learning, PMLR, Lille, France, 6–11 July 2015; pp. 674–683.
24. Perekrestenko, D.; Cevher, V.; Jaggi, M. Faster coordinate descent via adaptive importance sampling. In Proceedings of the Artificial Intelligence and Statistics, PMLR, Fort Lauderdale, FL, USA, 20–22 April 2017; pp. 869–877.
25. Alain, G.; Lamb, A.; Sankar, C.; Courville, A.; Bengio, Y. Variance reduction in sgd by distributed importance sampling. *arXiv* **2016**, arXiv:1511.06481.
26. Stich, S.U.; Raj, A.; Jaggi, M. Safe adaptive importance sampling. *Adv. Neural Inf. Process. Syst.* **2017**, *30* .
27. Lu, L.; Jin, P.; Pang, G.; Zhang, Z.; Karniadakis, G.E. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nat. Mach. Intell.* **2021**, *3*, 218–229.
28. Majda, A.J.; McLaughlin, D.W.; Tabak, E.G. A one-dimensional model for dispersive wave turbulence. *J. Nonlinear Sci.* **1997**, *7*, 9–44.
29. Blanchard, A.; Sapsis, T. Output-weighted optimal sampling for Bayesian experimental design and uncertainty quantification. *SIAM/ASA J. Uncertain. Quantif.* **2021**, *9*, 564–592.
30. Sapsis, T.P. Output-weighted optimal sampling for Bayesian regression and rare event statistics using few samples. *Proc. R. Soc. A* **2020**, *476*, 20190834.
31. Sapsis, T.P.; Blanchard, A. Optimal criteria and their asymptotic form for data selection in data-driven reduced-order modelling with Gaussian process regression. *Philos. Trans. R. Soc. A* **2022**, *380*, 20210197.
32. Angluin, D.; Smith, C.H. Inductive inference: Theory and methods. *ACM Comput. Surv. (CSUR)* **1983**, *15*, 237–269.
33. Blumer, A.; Ehrenfeucht, A.; Haussler, D.; Warmuth, M.K. Occam's razor. *Inf. Process. Lett.* **1987**, *24*, 377–380.
34. Vapnik, V. *The Nature of Statistical Learning Theory*; Springer Science & Business Media: New York, NY, USA, 1999.
35. Hastie, T.; Tibshirani, R.; Friedman, J.H. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*; Springer: New York, NY, USA, 2009; Volume 2.
36. Lakshminarayanan, B.; Pritzel, A.; Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *Adv. Neural Inf. Process. Syst.* **2017**, *30* .
37. Wilson, A.G.; Izmailov, P. Bayesian Deep Learning and a Probabilistic Perspective of Generalization. In Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS'20, Vancouver, BC, Canada, 6–12 December 2020.
38. Pickering, E.; Sapsis, T.P. Structure and Distribution Metric for Quantifying the Quality of Uncertainty: Assessing Gaussian Processes, Deep Neural Nets, and Deep Neural Operators for Regression. *arXiv* **2022**, arXiv:2203.04515.
39. Sapsis, T.P. Statistics of extreme events in fluid flows and waves. *Annu. Rev. Fluid Mech.* **2021**, *53*, 85–111.
40. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707.
41. Cai, D.; Majda, A.J.; McLaughlin, D.W.; Tabak, E.G. Spectral bifurcations in dispersive wave turbulence. *Proc. Natl. Acad. Sci. USA* **1999**, *96*, 14216–14221.
42. Zakharov, V.E.; Guyenne, P.; Pushkarev, A.N.; Dias, F. Wave turbulence in one-dimensional models. *Phys. D Nonlinear Phenom.* **2001**, *152*, 573–619.
43. Zakharov, V.E.; Dias, F.; Pushkarev, A. One-dimensional wave turbulence. *Phys. Rep.* **2004**, *398*, 1–65.
44. Pushkarev, A.; Zakharov, V.E. Quasibreathers in the MMT model. *Phys. D Nonlinear Phenom.* **2013**, *248*, 55–61.
45. Cousins, W.; Sapsis, T.P. Quantification and prediction of extreme events in a one-dimensional nonlinear dispersive wave model. *Phys. D Nonlinear Phenom.* **2014**, *280*, 48–58.
46. Rasmussen, C.E. Gaussian processes in machine learning. In *Summer School on Machine Learning*; Springer: New York, NY, USA, 2003; pp. 63–71.
47. Shahriari, B.; Swersky, K.; Wang, Z.; Adams, R.P.; De Freitas, N. Taking the human out of the loop: A review of Bayesian optimization. *Proc. IEEE* **2015**, *104*, 148–175.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.