Information FOMO: The unhealthy fear of missing out on information. A method for removing misleading data for healthier models

Ethan Pickering[1,*] and Themistoklis P. Sapsis[1,*]

[1]Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02138, USA

October 24, 2022

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 is either misleading or brings unnecessary complexity to the surrogate model of choice. Specifically, our method eliminates the phenomena of "double descent", where more data leads to worse performance. 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.

1 Introduction

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 is not always better. Data may present misleading information, or even worse, lead to overfitting. The overfitting case is linked to the concept of "deep double descent"[25, 5, 21], where more data does not result in better models.

The 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 phenomena is observed with respect to model complexity (i.e. number of layers or layer width), training epochs, or training samples [20], and has been both theoretically appreciated and empirically observed in numerous studies [25, 38, 11, 5, 1, 21, 12, 27]. Although all three manifestations of double descent can lead to significant errors, the first two, model complexity and training epochs, can be avoided through straight forward model parameter studies, such as "early stopping" strategies [44] in either training time [14] 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 literature surrounding sample-wise double descent is recognized in ridgeless regression [12, 22] or random features regression [19]. Improved generalization error due to sample-wise double descent has been observed through appropriate ℓ_2 regularization of the

---

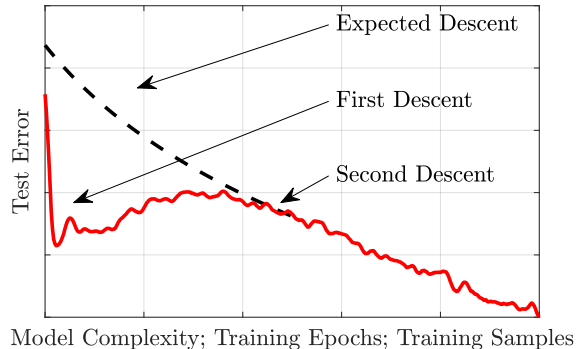*Co-corresponding authors: pickering@mit.edu, sapsis@mit.edu

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.

model in ridge regression [12, 22] or data-dependent regularizers in deep models [41, 42], however, the latter provides modest improvements of a few percent for most presented cases. Our work here follows in the vein of data-dependent regularizers, where the data itself is seen as a component to the level of complexity of the model.

While not applied to concepts of sample-wise double decent, there are several approaches for selecting subsets of training data to *reduce training cost* on large datasets that draw parallels to our approach. To reduce neural network training cost, random uniform sampling [36, 37], importance sampling [3, 24, 23, 15, 47], and adaptive importance sampling [10, 26, 2] are used to choose only a fraction of the data for training evaluations. Adaptive importance sampling is closest to our approach, using probability distributions to define sample importance and update the training set at each iteration [39]. 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.

Here we propose a method, inspired by Bayesian experimental design, for eliminating double descent through identifying, and ignoring, data that brings unnecessary complexity to the model. This is done 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 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, 1 out of every 20 samples, provide useful information. Together, the unnecessary need for splitting data intro 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.

To demonstrate the efficacy of our method we first present results for a simple, nonlinear $1D$ function approximated by Gaussian process (GP) regression that presents a visual example of sample-wise double descent. To our knowledge, double descent has yet to be observed for GP regression, underscoring a, later discussed, fundamental difference and challenge in sample-wise double descent. The approach is then extended to a larger and significantly more complex problem with 20,000 training samples with a Deep Neural Networks (DNNs) taking the role of surrogate model. We then discuss the generality and implications of the approach to limited and sparse datasets as well as any Bayesian or ensemble surrogate model.
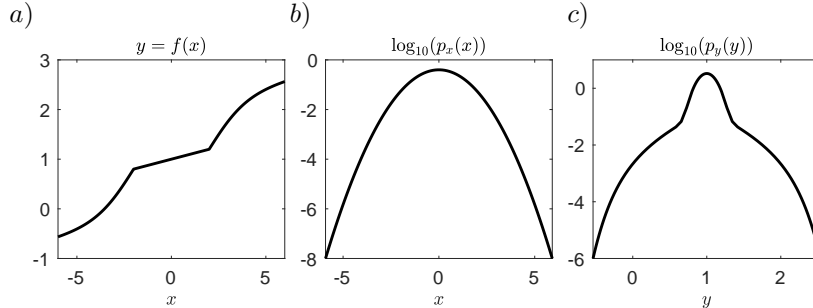
Figure 2: *a*) the true nonlinear solution $y = f(x)$, with respect to random variable x, *b*) the Gaussian PDF of x, and *c*) the non-Gaussian PDF of response variable y with heavy tails.

2 Results

The approach eliminates deep double descent in both mean-square error (MSE) and log-PDF error, GP and DNN surrogate models, and low and high dimensional problems, without the need for test or validation data.

2.1 $1D$ Example with Gaussian Process Regression

As with nearly all ML tasks, we wish to predict a state, y, from an observed input, x, by approximating the underlying map, $y = f(x)$. Approximating this map with a surrogate model may require substantial data depending on the complexity and dimension of the input space (e.g. x is multi-dimensional vector of inputs, $\mathbf{x}$). Figure 2 *a*) presents a test function we wish to approximate: a piece-wise nonlinear function (see § 4.2.1) with a linear core and nonlinear edges to emulate rare dynamical instabilities initiated at high magnitudes. 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 2 *b*). We may then calculate the PDF of y via standard weighted Gaussian kernel density estimators (KDE) as,

$$p_f(y) = \text{KDE}(\text{data} = y, \text{weights} = p_x(x)), \tag{1}$$

which displays a Gaussian core and heavy tails in figure 2 *c*), indicating rare and extreme events.

In figure 3, we test and observe training errors for 100 independent experiments for our two training approaches approximating 2 *a*) with a GP surrogate model, detailed in § 4.3.1, given n samples (sampled from a uniform distribution), ranging from 5-60, of observed input and output data, $\mathcal{D} = \{x_i, y_i\}_{i=1}^n$. Figures 3 *a*) and 3 *c*) present the normalized MSE and log-PDF test errors (see § 4.4 for error calculation details), respectively, against the number of samples in standard training, while 3 *b*) and 3 *d*) report the errors versus iteration using the "FOMO" algorithm, described in § 4.1. In each figure the median error is highlighted with shaded regions indicating the minimum and maximum errors.

Sample-wise double descent is observed for both metrics under standard training using GPs for the simple $1D$ problem, while the FOMO approach clearly eliminates this phenomenon. For this $1D$ case we will forego detailed early-stopping/training/testing/validation implications for the following example and focus on the 1D representations of double descent and do so by highlighting 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 3 *e*) 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.

Figure 3 *f*), presents another key feature of the FOMO algorithm, choosing only a fraction of the possible samples. The algorithm only adds samples that bring stable improvements to the model and ignores those that do not. We expand on this idea in the higher-dimensional case using DNNs next.
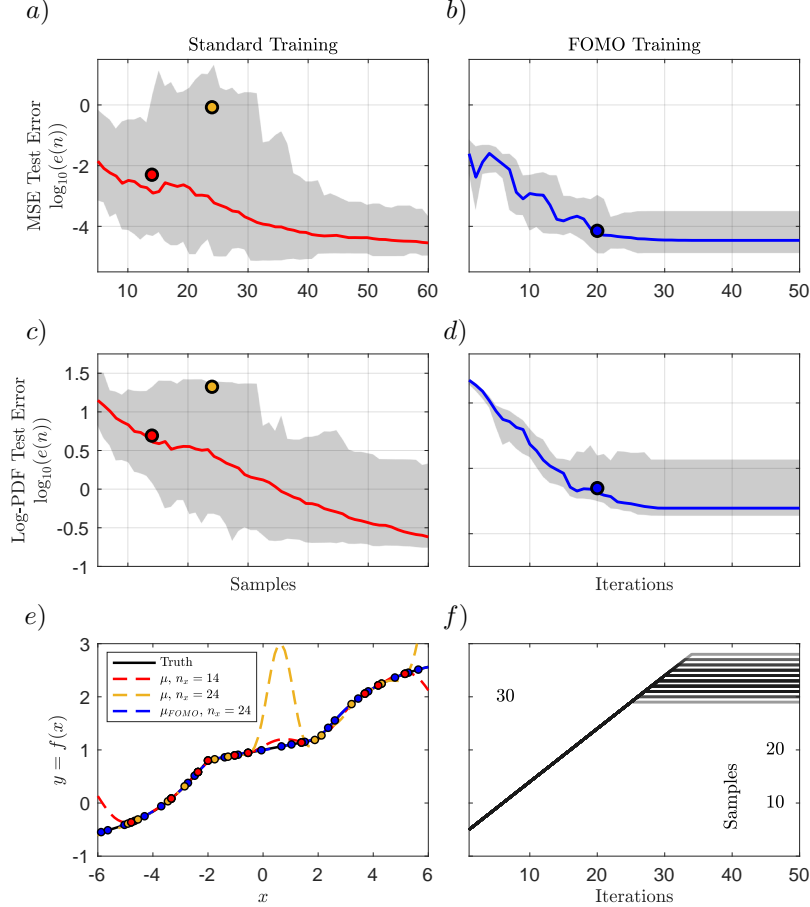
Figure 3: **GP FOMO model eliminates double descent, is superior to early stopping, and converges without testing or validation data.** The mean normalized MSE and log-PDF errors (with min and max values shaded) of 100 experiments with randomly chosen data samples (a) and c)), and with sequential selections (b) and d)). e) provides a comparison of the approximated and true solution for the errors denoted in $a) - d$) and f) is the number of chosen data samples by iteration for 100 independent sequential searches from b) and d).

2.2 Dispersive Nonlinear Wave Model with Deep Neural Networks

We now 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) [18] for the study of 1D wave turbulence. The initial conditions are chosen randomly from an $8D$ subspace: $u(x_L, t) = \mathbf{x}\mathbf{\Phi}(x_L)$, where x_L refers to the spatial variable, while x or $\mathbf{x}$ represents the random coefficients (an $8D$ vector), which are assumed to follow a known distribution. Our aim is to identify a maximum future wave height $|Re(u(x, t = \tau))|_\infty$ (see § 4.2.2 and § 4.2 for details on MMT and the wave height map). While GPs could be used as a surrogate, a companion study [27] showed DNNs are superior for the complexity of this problem and are used here, see § 4.3.2 for DNN details. Additionally, this example requires many training samples, at least at an order of magnitude of 1,000 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 experiment of both training approaches for approximating the underlying map with a DNN given n samples (sampled via Latin Hypercube sampling, LHS), ranging from 50-20,000, of observed input and output data, $\mathcal{D} = \{\boldsymbol{u}_i(x_L), y_i\}_{i=1}^n$. Figures 4 a) and 4 c) present the normalized MSE and log-PDF test errors, respectively, against the number of samples, while 4 b) and 4 d) report the errors versus iteration using the "FOMO" algorithm. In figures
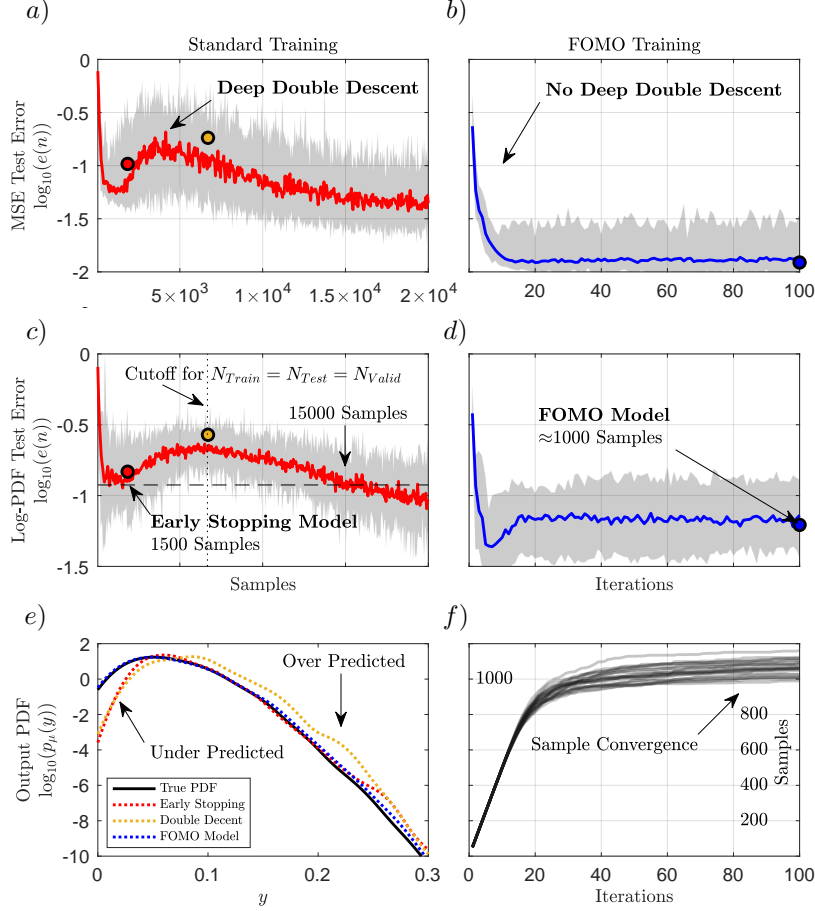
4

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) and c)), and with sequential selections (b) and d)). e) provides a comparison of the predicted and true output distributions for three errors denoted in $a) - d$) and f) the number of chosen data samples by iteration for 25 independent sequential searches from b) and d).

4 a) and 4 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 e) shows a drastic under prediction of the most probable states. More concerning, if an unconservative splitting of the 20,000 samples is taken, i.e. 6,700 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 e).

The "Information FOMO" results shown in figures 4 b) and 4 d), and found via algorithm 1, do not suffer from these difficulties. Instead, the FOMO approach provides the most accurate model (see figure 4 e)), does not undergo double descent, converges, 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 eliminates uncertainty in the model or provides essential information. Consequently, data that does not bear beneficial information to the model is deemed unnecessary and ignored.

Figure 4 f) 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_a samples (here $n_a = 50$) 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
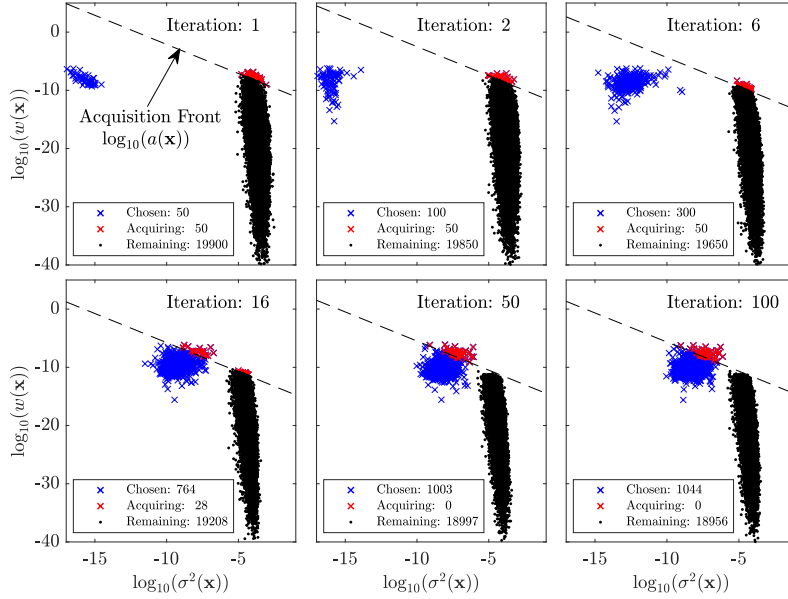
Figure 5: **Important and misleading/unnecessary datasets show clear separation.** A representative example of the iterative selection process, where data is sequentially acquired. The acquisition front indicates the acquisition score of the 50th optimal point at each iteration. Those above are acquired and those already chosen remain in the training set.

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 (figures *b*) and *d*)). Critically, this implies that the FOMO approach does not require data 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 [6],

$$w(\mathbf{x}) = \frac{p_{\mathbf{x}}(\mathbf{x})}{p_\mu(\mu)}, \tag{2}$$

while the $x-$axis presents the predictive variance, $\sigma^2(\mathbf{x})$, found amongst the ensemble of independently weight-initialized DNN predictions. The product of these two quantities is the acquisition function proposed by [6, 32], $a(\mathbf{x}) = w(\mathbf{x})\sigma^2(\mathbf{x})$. As shown in [34], the adopted acquisition function balances information with 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 [4, 7, 40, 13, 5].

Walking through the iterations of figure 5 we can see how the approach dynamically partitions the data into informative and unnecessary. Starting at iteration 1, only 50 samples, denoted in blue are used to train the model (See algorithm 1 for how these samples are chosen). Unsurprisingly, such little data allows 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_a samples, with respect to the acquisition function, retrain the model with the addition of the new samples, and repeat the process. Here we chose $n_a = 50$ and the 50th point sets what we define as the "acquisition front", denoting a line of constant acquisition value between samples the model wishes to acquire and those it does not. Iteration 2 and 6 show that as more data is acquired into the chosen/training set the uncertainty in the chosen set increases, while the uncertainty in the remaining set decreases. The latter observation
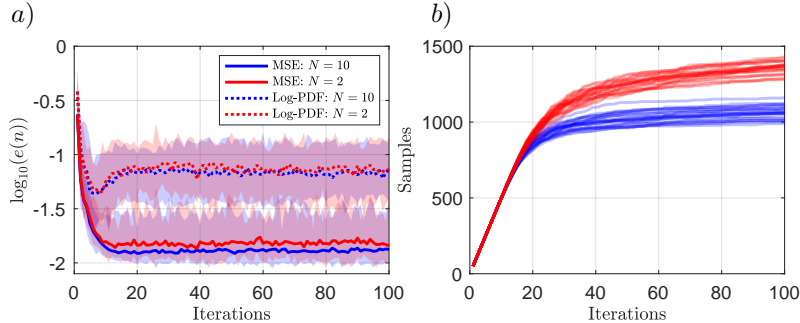
Figure 6: **Shallow ensembles perform well.** *a*) The median MSE and log-PDF errors (with min and max values shaded) of 25 experiments using DNN ensemble sizes of $N = 10$ and $N = 2$. *b*) the number of chosen data samples by iteration for 25 independent sequential searches for both $N = 10$ and $N = 2$.

is simply due to the increased training data that reduces 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 DNNs training errors increase. This behavior is critical to identifying when the DNN parameterization becomes stressed and susceptible to instabilities related to double descent. This is shown in the following iterations.

Iterations 16, 50, and 100, demonstrate the ability of the model to ignore data when 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 the 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 this 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 [16], are employed to calculate the predictive uncertainty. Although the validity of such approaches is hotly debated, [43] and also [28], 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 6 *a*) 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 6 *b*), where $N = 2$ chooses approximately 25% more samples. The surprisingly ability of shallow ensembles of $N = 2$ to perform well in active learning schemes was also observed in [27].

3   Discussion

Sample-wise 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 brings unwarranted instability to the chosen model that otherwise experiences an increase in test/generalization error with increased samples size, an instability which is often unrecoverable. As most datasets in real-world applications are fixed, 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 4-fold. An often unappreciated metric, the log-PDF error appropriately weights high-magnitude, rare phenomena that ML methods struggle to accurately predict [33].

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 DNN 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 inputs parameters, unlike robust 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, $p_\mu(\mu)$ 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 (i.e. 12,500 here), 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 misleading 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 2 samples, yet double descent is observed from 5-30 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 is not appropriately distributed in the input space, as is the case when randomly partitioned, than 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.

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 the surrogate solution for the entire dataset is located at the peak of double descent, the initial output PDF will emit a suboptimal FOMO initialization. Two approaches could help mitigate this: initialize with the early stopping solution or propose a physics-based assumption of the underlying output PDF.

2. The above underscores that although the approach will extract the most relevant samples, if no subset of the samples are sufficient to characterize the map, FOMO will not improve performance.

3. 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 would remove the instability.

4. Other regularization approaches for instabilities could be employed for DNNs, such as physical constraints or physics-based loss functions (e.g. PINNs [30]), but their efficacy for double descent is unknown. Additionally, the use of regularizations does not invalidate use of the FOMO method.

5. 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.

4 Methods

4.1 FOMO algorithm

The "FOMO" sequential selection method for fixed datasets is outlined in algorithm 1. The method is initialized by first choosing the n_a "optimal" samples of the dataset. These are determined by calculating the likelihood ratio, $w(\mathbf{x}) = p_{\mathbf{x}}(\mathbf{x})/p_\mu(\mu)$, or information metric, on all samples after approximating the output PDF, p_μ, through one surrogate model (e.g. DNN or GP) training on the complete input-output dataset. Once the top scoring information metric samples are chosen, the surrogate model is trained with these n_a samples and emit predictions of μ and σ^2 for all observed data pairs. The surrogate model is also 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 product of the information metric and predictive variance provide the acquisition values, $a(\mathbf{x})$ [6], and the best n_a samples, amongst the whole dataset, are augmented to the training set. Finally, because *all* samples are given acquisition scores, duplicates, i.e. a current training sample is chosen again, are removed. The process is then repeated until the algorithm ceases to acquire any unique samples.

Algorithm 1 Sequential selection algorithm for fixed data.

1: **Initialize:** Train surrogate model on **full** dataset of I-O pairs
2: Build weight function $w(\mathbf{x})$, choose n_{init} best samples from the dataset.
3: Retrain surrogate model on only n_{init} best samples.
4: **for** $n = 1$ **to** n_{iter} **do**
5:     From **full** dataset, select best point $\mathbf{x}_{n_a}$ for acquisition function $a(\mathbf{x})$
6:     Augment training dataset and **remove** duplicates.
7:     Retrain surrogate model on augmented dataset.
8: **end for**
9: **return** Final surrogate model *and* training dataset.

4.2 Map definition and data

4.2.1 Nonlinear $1D$ Function

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

$$y = \left(ax + \frac{L}{1 + e^{-k(x+2)}} - L/2\right)/b + 1 \qquad\qquad x < -2 \tag{3}$$

$$y = ax/b + 1 \qquad\qquad -2 \geq x \leq 2 \tag{4}$$

$$y = \left(ax + \frac{L}{1 + e^{-k(x-2)}} - L/2\right)/b + 1 \qquad\qquad x > 2, \tag{5}$$

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

4.2.2 Nonlinear Dispersive Wave Model

The underlying nonlinear system in our study is the Majda, McLaughlin, and Tabak (MMT) [18] model, a dispersive nonlinear model that also includes selective dissipation for high wavenumbers, used for studying 1D wave turbulence. It is described by

$$iu_t = |\partial_x|^\alpha u + \lambda |\partial_x|^{-\beta/4} \left(\left| |\partial_x|^{-\beta/4} u \right|^2 |\partial_x|^{-\beta/4} u \right) + iDu, \tag{6}$$

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 [18, 8] and presents a unique utility as a physical model for extreme ocean waves, or rogue waves [46, 45, 29]. We refer the reader to [27], where identical parameters were used and to [9] 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

$$u(x, t = 0) \approx \mathbf{x}\mathbf{\Phi}(x_L), \quad \forall \quad x_L \in [0, 1) \tag{7}$$

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 $\{\mathbf{\Lambda}, \mathbf{\Phi}(x)\}$ contains the first m eigenpairs via a Karhunen-Loeve expansion of the correlation matrix defined by the complex and periodic kernel,

$$k(x_L, x_L') = \sigma_u^2 e^{i(x_L - x_L')} e^{-\frac{(x_L - x_L')^2}{\ell_u}}, \tag{8}$$

with $\sigma_u^2 = 1$ 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, in that 6 standard deviations in each direction from the mean. Here we chose $m = 4$, or 8 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 [27] 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,

$$f(\mathbf{x}) = ||Re(u(x_L, t = T; \mathbf{x}))||_\infty, \tag{9}$$

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 consist 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.

4.3 Surrogate Models

4.3.1 Gaussian Process Regression

Gaussian process regression [31] 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

$$\mu(\mathbf{x}) = m_0 + k(\mathbf{x}, \mathbf{X})\mathbf{K}^{-1}(\mathbf{y} - m_0) \tag{10}$$

$$\sigma^2(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - k(\mathbf{x}, \mathbf{X})\mathbf{K}^{-1}k(\mathbf{X}, \mathbf{x}) \tag{11}$$

respectively, where $\mathbf{K} = k(\mathbf{X}, \mathbf{X}) + \sigma_\epsilon^2 \mathbf{I}$. Equation (10) can predict the value of the surrogate model at any point $\mathbf{x}$, and (11) to quantify uncertainty in prediction at that point [31]. Here, we chose the radial-basis-function (RBF) kernel with automatic relevance determination (ARD),

$$k\left(\mathbf{x}, \mathbf{x}'\right) = \sigma_f^2 \exp\left[-\left(\mathbf{x} - \mathbf{x}'\right)^\top \mathbf{L}^{-1}\left(\mathbf{x} - \mathbf{x}'\right)/2\right], \tag{12}$$

where $\boldsymbol{L}$ is a diagonal matrix containing the lengthscales for each dimension and the GP hyperparameters appearing in the covariance function (σ_f^2 and $\boldsymbol{L}$ in (11) are trained by maximum likelihood estimation). Additionally, training the GP in equation (11) 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 [31, 35]. The prohibitive cost of this for large datasets ($> \mathcal{O}(10^3)$) is one reason DNNs are used for the MMT case.

4.3.2 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 [17]. The network consists of 8 layers, 250 neurons, is trained for 1000 epochs, under a learning rate of $l_r = 0.001$, 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

$$\sigma^2(u) = \frac{1}{(N-1)} \sum_{n=1}^{N} (\tilde{G}_n(u) - \overline{\tilde{G}(u)})^2, \tag{13}$$

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 $u = \mathbf{x}\boldsymbol{\Phi}(x_L^1, ... x_L^m)$, 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:

$$\sigma^2(\mathbf{x}) = \frac{1}{(N-1)} \sum_{n=1}^{N} \left(\tilde{G}_n(\mathbf{x}\boldsymbol{\Phi}(x_L^1, ... x_L^m)) - \overline{\tilde{G}(\mathbf{x}\boldsymbol{\Phi}(x_L^1, ... x_L^m))} \right)^2. \tag{14}$$

4.4 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,

$$e_{\text{MSE}}(n) = \frac{\sum_{i=1}^{n} (y_{\text{pdf},i} - \mu_{\text{pdf},i})^2}{\sum_{i=1}^{n} y_{\text{pdf},i}^2} \tag{15}$$

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

$$e_{\log - \text{PDF}}(n) = \int |\log_{10} p_{\mu_n}(y) - \log_{10} p_f(y)| \mathrm{d}y, \tag{16}$$

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

$$p_f(y) = \text{KDE}(\text{data} = y_{\text{lhs}}, \text{weights} = p_\mathbf{x}(\mathbf{X}_{\text{lhs}})), \tag{17}$$

and

$$p_\mu(y) = \text{KDE}(\text{data} = \mu_{\text{lhs}}, \text{weights} = p_\mathbf{x}(\mathbf{X}_{\text{lhs}})). \tag{18}$$

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 insure that the approximation appropriately accounts for PDF tails, where rare, high magnitude events, i.e. extreme events, live.

11

Code and Data Availability

Data and code pertaining to the FOMO sequential search algorithm can be provided upon request and will be made public upon publication.

Acknowledgments

References

[1] M. S. Advani, A. M. Saxe, and H. Sompolinsky. High-dimensional dynamics of generalization error in neural networks. *Neural Networks*, 132:428–446, 2020.

[2] G. Alain, A. Lamb, C. Sankar, A. Courville, and Y. Bengio. Variance reduction in sgd by distributed importance sampling. *arXiv preprint arXiv:1511.06481*, 2015.

[3] Z. Allen-Zhu, Z. Qu, P. Richtárik, and Y. Yuan. Even faster accelerated coordinate descent using non-uniform sampling. In *International Conference on Machine Learning*, pages 1110–1119. PMLR, 2016.

[4] D. Angluin and C. H. Smith. Inductive inference: Theory and methods. *ACM Computing Surveys (CSUR)*, 15(3):237–269, 1983.

[5] M. Belkin, D. Hsu, S. Ma, and S. Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

[6] A. Blanchard and T. Sapsis. Output-weighted optimal sampling for bayesian experimental design and uncertainty quantification. *SIAM/ASA Journal on Uncertainty Quantification*, 9(2):564–592, 2021.

[7] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Occam's razor. *Information Processing Letters*, 24(6):377–380, 1987.

[8] D. Cai, A. J. Majda, D. W. McLaughlin, and E. G. Tabak. Spectral bifurcations in dispersive wave turbulence. *Proceedings of the National Academy of Sciences*, 96(25):14216–14221, 1999.

[9] W. Cousins and T. P. Sapsis. Quantification and prediction of extreme events in a one-dimensional nonlinear dispersive wave model. *Physica D: Nonlinear Phenomena*, 280:48–58, 2014.

[10] D. Csiba, Z. Qu, and P. Richtárik. Stochastic dual coordinate ascent with adaptive probabilities. In *International Conference on Machine Learning*, pages 674–683. PMLR, 2015.

[11] M. Geiger, S. Spigler, S. d'Ascoli, L. Sagun, M. Baity-Jesi, G. Biroli, and M. Wyart. Jamming transition as a paradigm to understand the loss landscape of deep neural networks. *Physical Review E*, 100(1):012115, 2019.

[12] T. Hastie, A. Montanari, S. Rosset, and R. J. Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *The Annals of Statistics*, 50(2):949–986, 2022.

[13] T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[14] R. Heckel and F. F. Yilmaz. Early stopping in deep networks: Double descent and how to eliminate it. *arXiv preprint arXiv:2007.10099*, 2020.

[15] A. Katharopoulos and F. Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *International conference on machine learning*, pages 2525–2534. PMLR, 2018.

[16] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, 30, 2017.

[17] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021.

[18] A. J. Majda, D. W. McLaughlin, and E. G. Tabak. A one-dimensional model for dispersive wave turbulence. *Journal of Nonlinear Science*, 7(1):9–44, 1997.

[19] S. Mei and A. Montanari. The generalization error of random features regression: Precise asymptotics and the double descent curve. *Communications on Pure and Applied Mathematics*, 75(4):667–766, 2022.

[20] P. Nakkiran. More data can hurt for linear regression: Sample-wise double descent. *arXiv preprint arXiv:1912.07242*, 2019.

[21] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003, 2021.

[22] P. Nakkiran, P. Venkat, S. Kakade, and T. Ma. Optimal regularization can mitigate double descent. *arXiv preprint arXiv:2003.01897*, 2020.

[23] D. Needell, R. Ward, and N. Srebro. Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm. *Advances in neural information processing systems*, 27, 2014.

[24] Y. Nesterov and S. U. Stich. Efficiency of the accelerated coordinate descent method on structured optimization problems. *SIAM Journal on Optimization*, 27(1):110–123, 2017.

[25] M. Opper. Statistical mechanics of learning: Generalization. *The handbook of brain theory and neural networks*, pages 922–925, 1995.

[26] D. Perekrestenko, V. Cevher, and M. Jaggi. Faster coordinate descent via adaptive importance sampling. In *Artificial Intelligence and Statistics*, pages 869–877. PMLR, 2017.

[27] E. Pickering, G. E. Karniadakis, and T. P. Sapsis. Discovering and forecasting extreme events via active learning in neural operators. *arXiv preprint arXiv:2204.02488*, 2022.

[28] E. Pickering and T. P. Sapsis. Structure and distribution metric for quantifying the quality of uncertainty: Assessing gaussian processes, deep neural nets, and deep neural operators for regression. *arXiv preprint arXiv:2203.04515*, 2022.

[29] A. Pushkarev and V. E. Zakharov. Quasibreathers in the MMT model. *Physica D: Nonlinear Phenomena*, 248:55–61, 2013.

[30] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

[31] C. E. Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.

[32] T. P. Sapsis. Output-weighted optimal sampling for Bayesian regression and rare event statistics using few samples. *Proceedings of the Royal Society A*, 476(2234):20190834, 2020.

[33] T. P. Sapsis. Statistics of extreme events in fluid flows and waves. *Annual Review of Fluid Mechanics*, 53:85–111, 2021.

[34] T. P. Sapsis and A. Blanchard. 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*, 380(2229):20210197, 2022.

[35] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

[36] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter. Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical programming*, 127(1):3–30, 2011.

[37] S. Shalev-Shwartz and T. Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *JMLR*, 14:567–599, 2013.

[38] S. Spigler, M. Geiger, S. d'Ascoli, L. Sagun, G. Biroli, and M. Wyart. A jamming transition from under-to over-parametrization affects loss landscape and generalization. *arXiv preprint arXiv:1810.09665*, 2018.

[39] S. U. Stich, A. Raj, and M. Jaggi. Safe adaptive importance sampling. *Advances in Neural Information Processing Systems*, 30, 2017.

[40] V. Vapnik. *The nature of statistical learning theory*. Springer Science & Business Media, 1999.

[41] C. Wei and T. Ma. Data-dependent sample complexity of deep neural networks via lipschitz augmentation. *Advances in Neural Information Processing Systems*, 32, 2019.

[42] C. Wei and T. Ma. Improved sample complexities for deep networks and robust classification via an all-layer margin. *arXiv preprint arXiv:1910.04284*, 2019.

[43] A. G. Wilson and P. Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. *arXiv preprint arXiv:2002.08791*, 2020.

[44] Y. Yao, L. Rosasco, and A. Caponnetto. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315, 2007.

[45] V. E. Zakharov, F. Dias, and A. Pushkarev. One-dimensional wave turbulence. *Physics Reports*, 398(1):1–65, 2004.

[46] V. E. Zakharov, P. Guyenne, A. N. Pushkarev, and F. Dias. Wave turbulence in one-dimensional models. *Physica D: Nonlinear Phenomena*, 152:573–619, 2001.

[47] P. Zhao and T. Zhang. Stochastic optimization with importance sampling for regularized loss minimization. In *international conference on machine learning*, pages 1–9. PMLR, 2015.