



Quantifying uncertainty for deep learning based forecasting and flow-reconstruction using neural architecture search ensembles

Romit Maulik^{a,*}, Romain Egele^b, Krishnan Raghavan^c, Prasanna Balaprakash^d

^a Information Science and Technology & Institute for Computational and Data Sciences, Pennsylvania State University, University Park, PA 16802, USA

^b Université Paris-Saclay, 1190 Gif-sur-Yvette, France

^c Argonne National Laboratory, Lemont, IL 60439, USA

^d Oak Ridge National Laboratory, 1 Bethel Valley Road, Oak Ridge, TN 37831, USA

ARTICLE INFO

Article history:

Received 19 February 2023

Received in revised form 20 June 2023

Accepted 11 July 2023

Available online 4 August 2023

Dataset link: <https://drive.google.com/drive/folders/1pVW4epkeHkT2WHZB7Dym5IURcfOP4cXu?usp=sharing>, <https://drive.google.com/drive/folders/1pVW4epkeHkT2WHZB7Dym5IURcfOP4cXu?usp=sharing>

Keywords:

Deep ensembles

Scientific machine learning

Neural architecture and hyperparameter search

ABSTRACT

Classical problems in computational physics such as data-driven forecasting and signal reconstruction from sparse sensors have recently seen an explosion in deep neural network (DNN) based algorithmic approaches. However, most DNN models do not provide uncertainty estimates, which are crucial for establishing the trustworthiness of these techniques in downstream decision making tasks and scenarios. In recent years, ensemble-based methods have achieved significant success for the uncertainty quantification in DNNs on a number of benchmark problems. However, their performance on real-world applications remains under-explored. In this work, we present an automated approach to DNN discovery and demonstrate how this may also be utilized for ensemble-based uncertainty quantification. Specifically, we propose the use of a scalable neural and hyperparameter architecture search for discovering an ensemble of DNN models for complex dynamical systems. We highlight how the proposed method not only discovers high-performing neural network ensembles for our tasks, but also quantifies uncertainty seamlessly. This is achieved by using genetic algorithms and Bayesian optimization for sampling the search space of neural network architectures and hyperparameters. Subsequently, a model selection approach is used to identify candidate models for an ensemble set construction. Afterwards, a variance decomposition approach is used to estimate the uncertainty of the predictions from the ensemble. We demonstrate the feasibility of this framework for two tasks – forecasting from historical data and flow reconstruction from sparse sensors for the sea-surface temperature. We demonstrate superior performance from the ensemble in contrast with individual high-performing models and other benchmarks.

© 2023 Elsevier B.V. All rights reserved.

1. Introduction

1.1. Motivation

Data-driven surrogate modeling research has shown great promise in improving the predictability and efficiency of computational physics applications [1,2]. Among various algorithms, deep learning-based models have been observed to show significant gains in accuracy and time-to-solution over classical numerical-methods based techniques. However, the widespread adoption of deep learning models is still limited by its black-box nature [3]. To that end, uncertainty quantification methods have been developed to overcome the challenges associated with the black-box nature of the deep-learning models and establish

trustworthiness by providing uncertainty estimates along with the predictions [4]. The data or aleatoric uncertainty is attributed to the noise in the data, for example, low resolution sensors and sparse measurements; the model or epistemic uncertainty is attributed to the lack of training data. The former is inherent to the data and cannot be reduced by collecting more data; the latter is used to characterize the model's predictive capability with respect to the training data and it can be reduced by collecting appropriate data. In many computational physics applications, it is crucial to effectively quantify both data- and model-form uncertainties in predictions from deep learning models. Quantification of such uncertainties alleviates the risks associated with the deployment of such black-box data-driven models for real-world tasks.

Deep ensembles is a promising approach for uncertainty quantification [5]. In this approach, an ensemble of neural networks (NNs) are trained independently but they differ in the way in

* Corresponding author.

E-mail address: rmaulik@anl.gov (R. Maulik).

which they are trained. Consequently, the weights of the neural network parameters will be different. The prediction from these models are then used to improve prediction and estimate uncertainty. Despite its simplicity, deep-ensembles-based uncertainty estimation has achieved superior performance over more sophisticated uncertainty quantification methods on a number of benchmarks.

Crucial to the effectiveness of uncertainty quantification in the deep-ensemble-based methods is the diversity of the high-performing models. Specifically, if the models are significantly different from each other and also equally high performing, then the prediction and the uncertainty estimates become accurate [6]. However, this poses additional challenges due to the large overhead associated with the manual design of models. Despite one time cost, the development overhead significantly increases the offline design and training costs of the ensemble model development and increase amortization time (i.e., the time required to offset offline costs). To that end, we explore an integrated automated deep ensemble approach that not only discovers high-performing models but also makes ensemble predictions with quantified uncertainty in a scalable manner. Specifically, the highlights of this article are as follows:

- We demonstrate a unified strategy for discovering high-performing deep learning models for dynamical systems forecasting and signal recovery with epistemic and aleatoric uncertainty quantification that leverages distributed computing.
- For uncertainty quantification, we use the law of decomposition of variance from members of an ensemble to separately estimate both the aleatoric and epistemic uncertainty.
- We validate our proposed approach for forecasting flow-fields as well as instantaneous state reconstructions for a real-world, high-dimensional scientific machine learning problem with complex dynamics, given by the NOAA Optimum Interpolation Data Set.

1.2. Related work

In recent times, deep learning models have been popular for complex predictive modeling tasks. In this section, we review past work in surrogate modeling of dynamical systems as well as attempts to quantify the uncertainty of these data-driven models. Data-driven surrogate models are primarily designed to reduce the computational costs associated with expensive forward models in many-query tasks [7–15]. Therefore these find extensive application in control [16–21], optimization [22,23], uncertainty quantification [24–26] and data-assimilation [27–30] among others. Typically, for complex dynamical systems possessing large degrees of freedom, a first step in model construction is the identification of a reduced-basis for evolving dynamics [31–33]. After identifying this basis, a cost-effective strategy is necessary for evolving the dynamics in this transformed space [7,34–39]. A popular approach to this latent-space dynamics evolution has been the use of neural networks with inductive biases for sequential data. For example, long short-term memory (LSTM) neural networks [40] have recently become very popular in modeling these dynamics [41–51]. Other methods include Gaussian processes [52–54], transformer models [55,56] which use temporal attention to learn patterns and make forecasts as well as neural ordinary differential equations (commonly known as the neural ODE), where neural networks are assumed to parameterize the right-hand side of an ODE [43,57]. We note that this review is not exhaustive and that several research studies have utilized variants of LSTMs with greater complexity for complex forecasting tasks.

Data-driven methods have also seen extensive use in flow-reconstruction from sparse sensors. This is due to a desire to build better technologies to reconstruct fluid-flow fields, that may have a large number of frequencies and degrees of freedom from sparse fixed sensors or moving tracer particles (such as those used in experimental imaging with particle image velocimetry). Typically, these techniques rely on linear stochastic [58] or gappy proper orthogonal decomposition [59] estimation. The former reconstructs the flow-field by building a correlation matrix between the sensor inputs and the full flow-fields while the latter solves a linear least-squares problem in an affine subspace spanned by the truncated proper orthogonal decomposition basis vectors. Neural network architectures have been seen to outperform these classical methods for various applications [60–62]. Variational inference-based neural architectures, potentially augmented with physics-informed biases, have also been used to reconstruct flow fields from sparse sensors with quantified uncertainty [63,64]. However, it is well-known that Bayesian approaches add significant computational cost, mainly due to their sampling process (e.g. Markov Chain Monte Carlo for characterizing the posterior) and are outperformed by ensemble-based approaches for uncertainty quantification [6].

From the perspective of uncertainty quantification for machine learning applications of computational science, several recent articles have attempted to develop algorithms that provide both predictions as well as confidence interval estimates. Here we draw a distinction between using machine learning to accelerate parametric uncertainty quantification, for example in Bayesian inversion with surrogate models (examples include Sherifdeen et al. [65], Constantine et al. [66], Goh et al. [26], Liu et al. [67], Lye et al. [68]), and performing uncertainty quantification of the machine learning algorithm predictions themselves. In the context of the latter, recently Maulik et al. [69] demonstrate how a probabilistic neural network may be used to capture the aleatoric uncertainty associated with surrogate model or flow-reconstruction predictions. In Morimoto et al. [70], the epistemic uncertainty of deep learning frameworks was captured using a weight sampling procedure during convergence, however, with strong assumptions of approximate convexity of the loss surface. In practical problems, particularly with physics-based constraints which promote non-smoothness [71], such an approach is limited. In Zhu et al. [72], a probabilistic surrogate model, based on a generative neural network architecture (the normalizing flow) was able to obtain a probability density function of the output quantity of interest, conditioned by the input parameters. While this approach avoids the Gaussian assumption for aleatoric uncertainty in Maulik et al. [69], the weights of the surrogate were deterministic and therefore did not quantify NN epistemic uncertainty. In Pawar et al. [73], both aleatoric and epistemic uncertainty were accounted for using the deep ensembles approach but all members of the ensemble were identical neural network architectures initialized differently and therefore were not optimized for the function approximation task.

In this study, we jointly address the issue of optimal architecture and hyperparameter selection, as well as deep ensembles based uncertainty quantification. It is well-known that selecting the best neural network architecture for a function approximation is a difficult task – to that end we have recently demonstrated how scalable neural architecture search may be used for surrogate model discovery [74]. In this article, we further expand on our algorithm by demonstrating how the architecture search may be vertically integrated for ensemble forecasting and signal recovery while providing predictions with confidence intervals.

2. Methods and data

2.1. AutoDEUQ: Uncertainty quantification through deep neural network ensembles

Automated deep ensemble with uncertainty quantification (AutoDEUQ) is a recently proposed deep-ensemble-based UQ method [6]. AutoDEUQ estimates aleatoric and epistemic uncertainties by automatically generating a catalog of NN models through joint neural architecture and hyperparameter search. The neural architecture and hyperparameters are optimized jointly, and the catalog contains all the evaluated configurations of neural networks (configuration = architecture + hyperparameters) completed during the search. Individual members of the catalog can have different types of residual connections, number of layers/units, learning rates, etc. Each catalog member is trained to minimize the negative log-likelihood to capture aleatoric uncertainty (data noise/uncertainty), and selecting a set of similarly performing models from the catalog to construct the ensembles and model epistemic uncertainty (i.e., function approximation uncertainty). The overall schematic of our neural architecture search based deep ensembles algorithm for uncertainty quantification of scientific machine learning applications is shown in Fig. 1. We explain the search process in further detail below.

We assume a supervised learning scenario, with a dataset \mathcal{D} composed of i.i.d points $(\mathbf{x}_i \in \mathcal{X}, y_i \in \mathcal{Y})$, where \mathbf{x}_i and $y_i = f(\mathbf{x}_i)$ are the input and the corresponding output of the i th point, respectively, and $\mathcal{X} \subset \mathbb{R}^N$ and $\mathcal{Y} \subset \mathbb{R}^M$ are the input and output spaces of N and M dimensions, respectively. In the context of regression problems, a focus of our study, the outputs are given by a scalar or vector of real values. Given \mathcal{D} , we seek to characterize the predictive distribution $p(y|\mathbf{x})$ using a parameterized distribution $p_\theta(y|\mathbf{x})$, which estimates aleatoric uncertainty through individual trained NNs and then estimates the epistemic uncertainty with an ensemble of high-performing NNs $p_\mathcal{E}(y|\mathbf{x})$. We define Θ to be the sample space for θ .

The aleatoric uncertainty can be characterized by using the quantiles of p_θ . Following previous work Lakshminarayanan et al. [5], we make the usual assumption of a Gaussian distribution for $p_\theta \sim \mathcal{N}(\mu_\theta, \sigma_\theta^2)$ and use variance as a measure of the aleatoric uncertainty. We explicitly partition θ into $(\theta_a, \theta_h, \theta_w)$ such that Θ is decomposed into $(\Theta_a, \Theta_h, \Theta_w)$, where $\theta_a \in \Theta_a$ represents the values of the neural architecture decision variables (network topology parameters), $\theta_h \in \Theta_h$ represents NN training hyperparameters (e.g., learning rate, batch size), and $\theta_w \in \Theta_w$ represents the NN weights. The NN is trained to output mean μ_θ and variance σ_θ^2 . For a given choice of architecture decision variables θ_a and training hyperparameters θ_h , to obtain θ_w^* , we seek to minimize the negative log-likelihood given the real data \mathcal{D} . Specifically, we can learn the aleatoric uncertainty using the negative log-likelihood loss (as opposed to the usual mean squared error) in the training: [5]:

$$\ell(\mathbf{x}, y; \theta) = -\log p_\theta = \frac{\log \sigma_\theta^2(\mathbf{x})}{2} + \frac{(y - \mu_\theta(\mathbf{x}))^2}{2\sigma_\theta^2(\mathbf{x})} + \text{cst}, \quad (1)$$

where cst is a constant. The NN training problem is then

$$\theta_w^* = \arg \max_{\theta_w \in \Theta_w} \ell(\mathbf{x}, y; \theta_a, \theta_h, \theta_w). \quad (2)$$

To model epistemic uncertainty, we use deep ensembles (an ensemble composed of NNs) [5]. In our approach, we generate a catalog of NN models $\mathcal{C} = \{\theta_i, i = 1, 2, \dots, c\}$ (where $\theta \in \Theta$ is a tuple of architecture, optimization hyperparameters, and weights) and repeatedly sample K models to form the ensemble $\mathcal{E}|\mathcal{E} = \{\theta_i, i = 1, 2, \dots, K\}$. Let p_θ describe the probability that θ is a member of the ensemble $\forall \theta \in \mathcal{C}$. Let $p_\mathcal{E}$ – the probability

density function of the ensemble – be obtained as a mixture distribution where the mixture is given as $p_\mathcal{E} = \mathbb{E}p_\theta$. Define μ_θ and σ_θ^2 as the mean and variance of each element in the ensemble, respectively. Then, the mean of the mixture is $\mu_\mathcal{E} := \mathbb{E}[\mu_\theta]$, and the variance [75] is

$$\sigma_\mathcal{E}^2 := \mathbb{V}[p_\mathcal{E}] = \underbrace{\mathbb{E}[\sigma_\theta^2]}_{\text{Aleatoric Uncertainty}} + \underbrace{\mathbb{V}[\mu_\theta]}_{\text{Epistemic Uncertainty}}, \quad (3)$$

where \mathbb{E} refers to the expected value and \mathbb{V} refers to the variance. Eq. (3) formally provides the decomposition of the overall uncertainty of the ensemble into its individual components such that $\mathbb{E}[\sigma_\theta^2]$ marginalizes the effect of θ and captures the aleatoric uncertainty and $\mathbb{V}[\mu_\theta]$ captures the spread of the prediction across different models and neglects the noise of the data, therefore capturing the epistemic uncertainty.

The overall estimate of the mean and the variance may then be written as

$$\begin{aligned} \mu_\mathcal{E} &= \frac{1}{K} \sum_{\theta \in \mathcal{E}} \mu_\theta \\ \sigma_\mathcal{E}^2 &= \underbrace{\frac{1}{K} \sum_{\theta \in \mathcal{E}} \sigma_\theta^2}_{\text{Aleatoric Uncertainty}} + \underbrace{\frac{1}{K-1} \sum_{\theta \in \mathcal{E}} (\mu_\theta - \mu_\mathcal{E})^2}_{\text{Epistemic Uncertainty}}, \end{aligned} \quad (4)$$

where K is the size of the ensemble. The total uncertainty here quantified by $\sigma_\mathcal{E}^2$ is a combination of aleatoric and epistemic uncertainty.

Let us assume that our total data \mathcal{D} is partitioned in the usual train-validation-test split manner as $\mathcal{D} = \mathcal{D}^{\text{train}} \cup \mathcal{D}^{\text{valid}} \cup \mathcal{D}^{\text{test}}$. A neural architecture configuration θ_a is a vector from the neural architecture search space Θ_a , defined by a set of neural architecture decision variables. A hyperparameter configuration θ_h is a vector from the training hyperparameter search space Θ_h defined by a set of hyperparameters used for training (e.g., learning rate, batch size). The problem of joint neural architecture and hyperparameter search can be formulated as the following bilevel optimization problem:

$$\begin{aligned} \theta_a^*, \theta_h^* &= \arg \max_{\theta_a, \theta_h} \frac{1}{N^{\text{valid}}} \sum_{\mathbf{x}, y \in \mathcal{D}^{\text{valid}}} \ell(\mathbf{x}, y; \theta_a, \theta_h, \theta_w^*) \\ \text{s.t. } \theta_w^* &= \arg \max_{\theta_w} \frac{1}{N^{\text{train}}} \sum_{\mathbf{x}, y \in \mathcal{D}^{\text{train}}} \ell(\mathbf{x}, y; \theta_a, \theta_h, \theta_w), \end{aligned} \quad (5)$$

where the best architecture decision variables θ_a^* and training hyperparameters values θ_h^* are selected based on $\mathcal{D} = \mathcal{D}^{\text{valid}}$ and the corresponding weights θ_w are selected based on $\mathcal{D} = \mathcal{D}^{\text{train}}$.

The pseudo-code of the AutoDEUQ is shown in Algorithm 1. To perform a joint neural architecture and hyperparameter search, we leverage aging evolution with asynchronous Bayesian optimization (AgEBO) [76].

Aging evolution (AgE) [77] is a parallel neural architecture search (NAS) method for searching over the architecture space. The AgEBO method follows the manager-worker paradigm, wherein a manager node runs a search method to generate multiple NNs and W workers (compute nodes) train them simultaneously. The AgEBO method constructs the initial population by sampling W architecture and W hyperparameter configurations and concatenating them (lines 1–7). The NNs obtained by using these concatenated configurations are sent for simultaneous evaluation on W workers (line 6). The iterative part (lines 8–26) of the method checks whether any of the workers finish their evaluation (line 9), collects validation metric values from the finished workers, and uses them to generate the next set of architecture and hyperparameter configurations for simultaneous evaluation to fill up the free workers that finished their

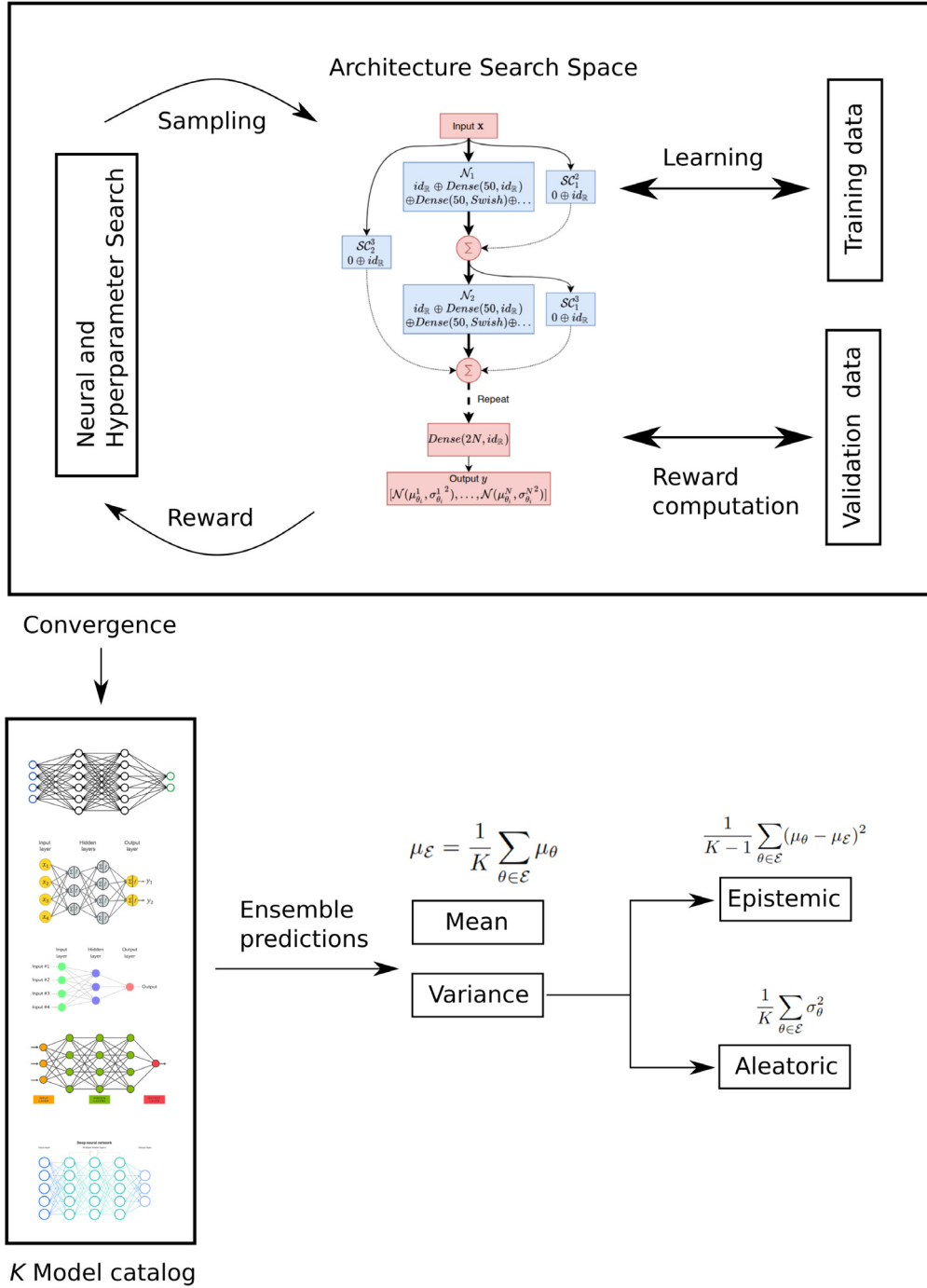


Fig. 1. Overall schematic of deep ensembles based uncertainty quantification using AutoDEUQ [6]. An outer-loop search algorithm samples neural network architectures from a search space which are optimized on training data and ranked in terms of performance on validation data (denoted the reward). Through an iterative process, the highest-ranked architectures are used for an ensemble-based function approximation with quantified uncertainty.

evaluations (lines 11–25). At a given iteration, in order to generate a NN, architecture, and hyperparameter configurations are generated in the following way. From the incumbent population, S NNs are sampled (line 17). A random mutation is applied to the best of S NNs to generate a child architecture configuration (line 18). This mutation is obtained by first randomly selecting an architecture decision variable from the selected NN and replacing its value with another randomly selected value excluding the current value. The new child replaces the oldest member of the population. The AgEBO optimizes the hyperparameters (θ_h) by marginalizing the architecture decision variables (θ_a). At a given iteration, to generate a hyperparameter configuration, the

AgEBO uses a (supervised learning) model M to predict a point estimate (mean value) $\mu(\theta_h^i)$ and standard deviation $\sigma(\theta_h^i)$ for a large number of unseen hyperparameter configurations. The best configuration is selected by ranking all sampled hyperparameter configurations using the upper confidence bound acquisition function, which is parameterized by $\kappa \geq 0$ that controls the trade-off between exploration and exploitation. To generate multiple hyperparameter configurations at the same time, the AgEBO leverages a multipoint acquisition function based on a constant liar strategy [78].

The catalog \mathcal{C} of NN models is obtained by running AgEBO and storing all the models from the runs. To build the ensemble \mathcal{E} of

Algorithm 1: AutoDEUQ: Neural architecture and hyperparameter search for ensemble construction.

```

inputs: P: population size, S: sample size, W: workers
output:  $\mathcal{E}$ : ensemble of models
/* Initialization for AgEBO */
1 population  $\leftarrow$  create_queue(P) // Alloc empty Q of size P
2 BO  $\leftarrow$  Bayesian_Optimizer()
3 for  $i \leftarrow 1$  to W do
4   config. $\theta_a \leftarrow$  random_sample( $\Theta_a$ )
5   config. $\theta_h \leftarrow$  random_sample( $\Theta_h$ )
6   submit_for_training(config) // Nonblocking
7 So
8 end
/* Optimization loop for AgEBO */
9 while stopping criterion not met do
  // Query results
10 results  $\leftarrow$  check_finished_training()
11  $\mathcal{C} \leftarrow \mathcal{C} \cup$  results // Add to catalogue population
12 if |results| > 0 then
13   population.push(results) // Aging population
14   // Generate hyperparameter configs
15   next  $\leftarrow$  BO.ask(|results|) // Generate architecture configs
16   for  $i \leftarrow 1$  to |results| do
17     if |population| = P then
18       parent.config  $\leftarrow$  select_parent(population, S)
19       child.config. $\theta_a \leftarrow$  mutate(parent. $\theta_a$ )
20     else
21       child.config. $\theta_a \leftarrow$  random_sample( $\Theta_a$ )
22     end
23     child.config. $\theta_h \leftarrow$  next[i]. $\theta_h$ 
24     submit_for_training(child.config) // Nonblocking
25   end
26 end
/* Ensemble selection */
27  $\mathcal{E} \leftarrow \{\}$ 
28 while | $\mathcal{E}$ .unique()|  $\leq K$  do
29    $\theta^* \leftarrow \operatorname{argmin}_{\theta \in \mathcal{C}} \ell(\theta, X, y)$  // Find lowest validation loss
30    $\mathcal{E} \leftarrow \mathcal{E} \cup \{\theta^*\}$  // Append model to ensemble
31 end
32 return  $\mathcal{E}$ 

```

models from \mathcal{C} , we adopt a top-K strategy (lines 27–32). Here, the idea is to select the K highest-performing models for the validation data and construct the ensemble from its individual predictions. This approach is appropriate when the validation data is representative of the generalization task (i.e., big enough, diverse enough, with good coverage) [79].

The architecture search space is modeled by using a directed acyclic graph, which starts and ends with input and output nodes, respectively. They represent the input and output layers of NN, respectively. Between the two are intermediate nodes defined by a series of variable \mathcal{N} and skip connection \mathcal{SC} nodes. Both types of nodes correspond to categorical decision variables. The variable nodes model dense layers with a list of different layer configurations. The skip connection node creates a skip connection between the variable nodes. This second type of node can take two values: disable or create the skip connection. For a given pair of consecutive variable nodes $\mathcal{N}_k, \mathcal{N}_{k+1}$, three skip connection nodes $\mathcal{SC}_{k-3}^{k+1}, \mathcal{SC}_{k-2}^{k+1}, \mathcal{SC}_{k-1}^{k+1}$ are created. These nodes allow for connection to the previous nonconsecutive variable nodes $\mathcal{N}_{k-3}, \mathcal{N}_{k-2}, \mathcal{N}_{k-1}$, respectively.

For the hyperparameter search space, we use a learning rate in the continuous range $[10^{-4}, 10^{-1}]$ with a log-uniform prior; a batch size in the discrete range $[32, \dots, b_{\max}]$ (where $b_{\max} = 256$) with a log-uniform prior; an optimizer in {sgd, rmsprop, adagrad, adam}; a patience value of 15 for the reduction of the learning rate, and value of 20 early stopping of training. The individual definitions of the architecture search space are experiment specific and are explored in the respective problem definitions below.

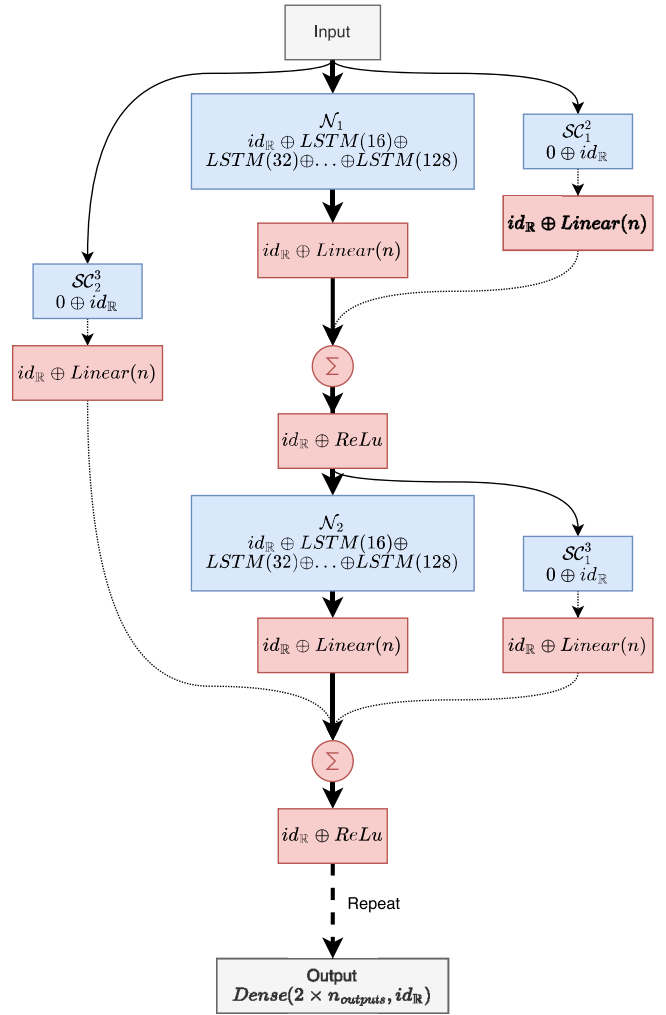


Fig. 2. A description of the neural architecture search space constructed for the flow-forecasting task.

Models are checkpointed during their evaluation based on the minimum validation loss achieved. We note that AutoDEUQ is deployed in the DeepHyper package, designed for scalable automatic machine learning on leadership class high-performance computers [80].

2.2. Dataset

In this study we use the open-source National Oceanic and Atmospheric Administration (NOAA) Optimum Interpolation sea surface temperature V2 data set (henceforth NOAA-SST).¹ This data set has a strong periodic structure due to seasonal fluctuations in addition to rich fine-scaled phenomena due to complex ocean dynamics. Weekly-averaged NOAA-SST snapshots are available on a quarter-degree grid which is sub-sampled to a one-degree grid for the purpose of demonstrating our proposed methodology in a computationally efficient manner. This data set, at the 1-degree resolution, has previously been used in several data-driven analysis tasks (for instance, see Kutz et al. [81], Callahan et al. [82] for specific examples), particularly from the point of view of extracting seasonal and long-term trends as well as for flow-field recovery [69]. Each “snapshot” of data

¹ Available at <https://www.esrl.noaa.gov/psd/>.

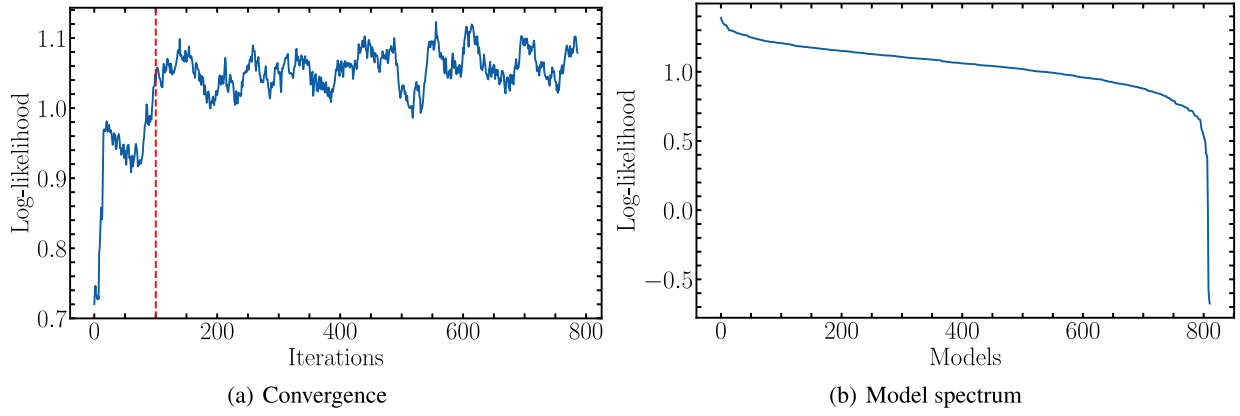


Fig. 3. Convergence of neural architecture search shown through moving averages (left) and spectrum of model performance (right) for the forecasting task. The red dashed line indicates the start of intelligent search space exploration.

originally corresponds to an array of size 360×180 (i.e., arranged according to the longitudes and latitudes of a one-degree resolution). However, for effective utilization in forecasting and flow-reconstruction, a mask is used to remove missing locations in the array that corresponds to the land area. Furthermore, it should be noted that regressions are performed for those coordinates which correspond to oceanic regions alone, and inland bodies of water are ignored. The non-zero data points then are subsequently flattened to obtain a column vector for each snapshot of our training and test data. We note that this data is available from October 22, 1981, to June 30, 2018 (i.e., 1,914 snapshots for the weekly averaged temperature).

2.3. Task 1: Forecasting

For our forecasting task, we utilize the period of October 22, 1981, to December 31, 1989. The rest (i.e., 1990 to 2018) is used for testing. Our final number of snapshots for training amounts to 427, and for testing amounts to 1487. This train-test split of the data set is a common configuration for data-driven studies [82] and the 8-year training period captures several short and long-term trends in the global sea surface temperature. Individual training samples are constructed by selecting a window of inputs (from the past) and a corresponding window of outputs (for the forecast task in the future) from the set of 427 training snapshots from the NOAA dataset. From the perspective of notation, if \mathbf{y}_t is a snapshot of training data, $t = 1, 2, \dots, 427$, a forecasting technique may be devised by learning to predict $\mathbf{y}_{t+1}, \dots, \mathbf{y}_{t+\tau}$ given $\mathbf{y}_t, \mathbf{y}_{t-1}, \dots, \mathbf{y}_{t-\tau}$. We note that this forecast is performed non-autoregressively – that is, the data-driven method is *not* utilized for predictions beyond the desired window size τ . Therefore, it is always assumed that the *true* $\mathbf{y}_t, \mathbf{y}_{t-1}, \dots, \mathbf{y}_{t-\tau}$ is available prior to making predictions. This means that given a window of true inputs (for example obtained via an observation of the state of the system given re-analysis data), a forecast is made for a series of outputs (corresponding to the window length) before a metric of accuracy is computed for optimization. This is in contrast to a situation where simply one step of a prediction is utilized for computing a fitness metric which adversely affects the ability of the predictive method for longer forecasts into the future (due to amplifying errors with each forecast step). The window size for the set of experiments on this data set is fixed at 8 weeks. The window-in and window-out construction of the training data leads to a final training data set of size 411 samples. Since this data set is produced by combining local and satellite temperature observations, it represents an attractive forecasting task for *non-intrusive* data-driven methods without requiring the physical modeling of underlying processes.

Subsequently, proper orthogonal decomposition (POD) provides a systematic method to project dynamics of a high-dimensional system onto a lower-dimensional subspace. We suppose that a single snapshot of the full system is a vector in \mathbb{R}^N , where N could be the number of grid points at which the field is resolved. Observing the system across a number of time points gives us the snapshots $\mathbf{y}_1, \dots, \mathbf{y}_T$, with mean subtracted by convention. The aim of POD is to find a small set of orthonormal basis vectors $\mathbf{v}_1, \dots, \mathbf{v}_M$, with $M \ll N$, which approximates the spatial snapshots, $\mathbf{y}_t \approx \sum_{j=1}^M a_j(t) \mathbf{v}_j$, $t = 1, \dots, T$, and so allows us to approximate the evolution of the full N dimensional system by considering only the evolution of the M coefficients $a_j(t)$. POD chooses the basis, \mathbf{v}_j , to minimize the residual with respect to the L_2 norm, $R = \sum_{t=1}^T \|\mathbf{y}_t - \sum_{j=1}^M a_j(t) \mathbf{v}_j\|^2$. Defining the snapshot matrix, $\mathbf{S} = [\mathbf{y}_1 | \dots | \mathbf{y}_T]$, the optimal basis is given by the M eigenvectors of $\mathbf{S}\mathbf{S}^T$, with largest eigenvalues, after which, the coefficients are found by orthogonal projection, $\mathbf{a}(t) = \langle \mathbf{y}_t, \mathbf{v} \rangle$ [83]. The coefficients $\mathbf{a}(t)$ correspond to a time-series with an M -dimensional state vector which is the focus of our temporal forecasting.

For our learning, we take only the training data snapshots, say $\mathbf{D}_1, \dots, \mathbf{D}_T$, from which we calculate the mean $\bar{\mathbf{D}} = (1/T) \sum_t \mathbf{D}_t$, hence defining the mean subtracted snapshots $\mathbf{y}_t = \mathbf{D}_t - \bar{\mathbf{D}}$. We then create the snapshot matrix, \mathbf{S} , and find numerically the M eigenvectors of $\mathbf{S}\mathbf{S}^T$ with largest eigenvalues. From this, we train models to forecast the coefficients making predictions of future coefficients given previous ones. Note that our model is also capable of making predictions for the aleatoric uncertainty, i.e., we can predict the variance of the output data given a Gaussian likelihood assumption for the loss function. This implies that we also predict variances in addition to the deterministic mean, i.e.,

$$f^\dagger(\mathbf{a}(t), \mathbf{a}(t-1), \dots, \mathbf{a}(t-\tau)) = ((\hat{\mathbf{a}}(t+1), \sigma_{\hat{\mathbf{a}}}^2(t+1)),$$

$$(\hat{\mathbf{a}}(t+2), \sigma_{\hat{\mathbf{a}}}^2(t+2)), \dots, (\hat{\mathbf{a}}(t+\tau), \sigma_{\hat{\mathbf{a}}}^2(t+\tau))) \quad (6)$$

$$(\mathbf{a}(t+1), \mathbf{a}(t+2), \dots, \mathbf{a}(t+\tau)) \approx (\hat{\mathbf{a}}(t+1),$$

$$\hat{\mathbf{a}}(t+2), \dots, \hat{\mathbf{a}}(t+\tau)) \quad (7)$$

Here, τ corresponds to the time-delay that is embedded into the input feature space for the purpose of forecasting. In this article, τ also stands for the forecast length obtained from the fit model although in the general case, these quantities can be chosen to be different. We can now test for predictions on unseen data, $\mathbf{E}_1, \dots, \mathbf{E}_Q$, where \mathbf{E}_t is an unseen snapshot of data at time t and Q is the total number of test snapshots. Note that these test snapshots are obtained for a time-interval that has not been utilized for constructing the POD basis vectors. We

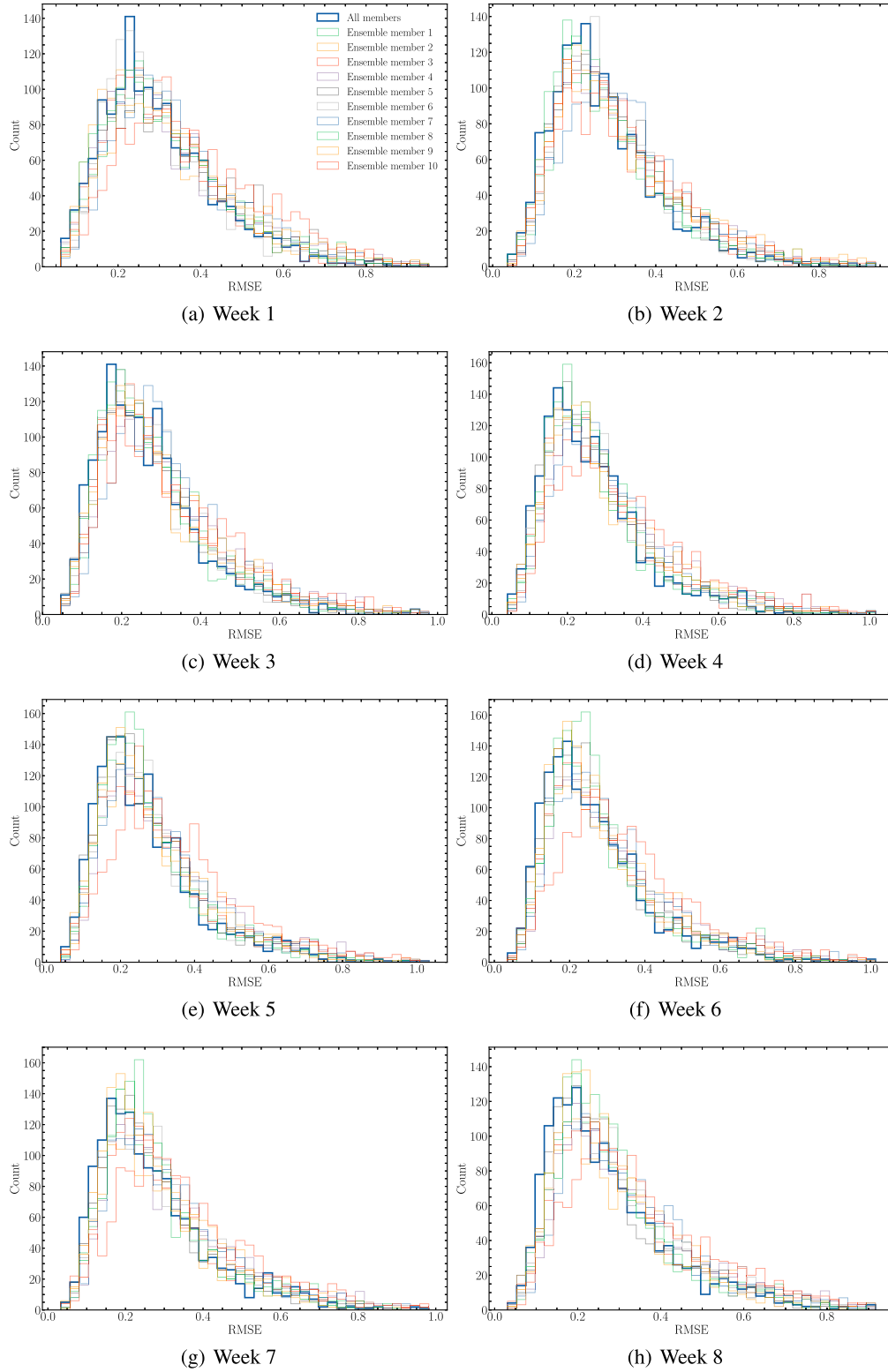


Fig. 4. Pointwise RMSE histograms for various forecast weeks showing individual member bin counts as well as ensemble mean performance. Most points on the grid show a RMSE of around 0.2.

proceed by taking the mean $\bar{\mathbf{D}}$, and vectors \mathbf{v}_j calculated from the training data to get test coefficients at an instant in time $\mathbf{a}_j(t) = (\mathbf{E}_t - \bar{\mathbf{D}}, \mathbf{v}_j)$, $j = 1, \dots, M$, which will be used with the model f^\dagger to make future predictions in the unseen testing interval. The prediction for the coefficients $\hat{\mathbf{a}}$, can be converted into predictions in the physical space by taking $\bar{\mathbf{D}} + \sum_j \hat{\mathbf{a}}_j \mathbf{v}_j$.

This procedure only makes use of testing data to pass into the model, not to train the model in any way. Crucially, to make a forecast of $\mathbf{E}_{t+1}, \mathbf{E}_{t+2}, \dots, \mathbf{E}_{t+\tau}$, only previous measurements $\mathbf{E}_t, \mathbf{E}_{t-1}, \dots, \mathbf{E}_{t-\tau}$ are needed. The connection to AgeBO comes in through how one may obtain several models like f^\dagger which may subsequently be used for forecasting.

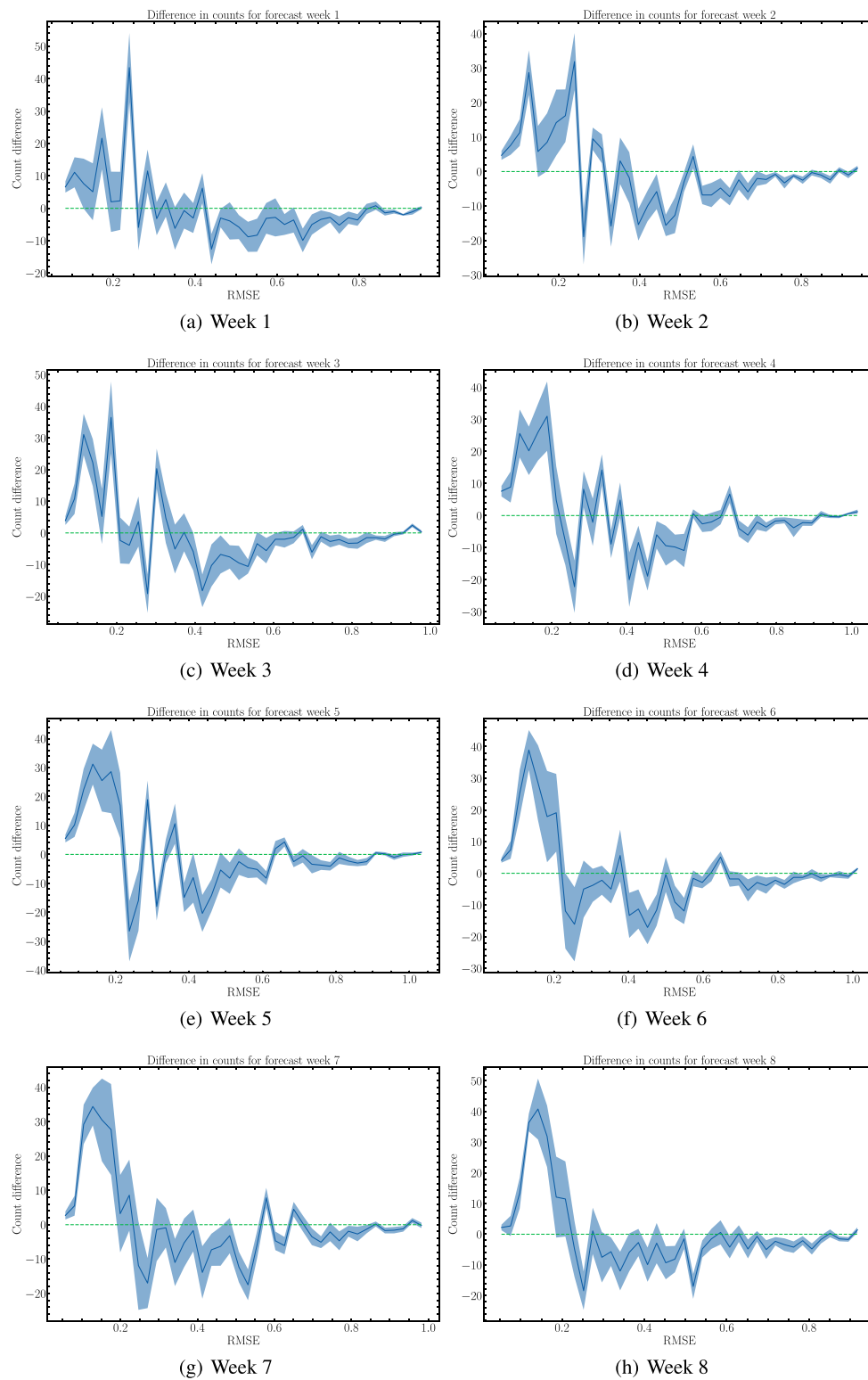


Fig. 5. Showing count differences between the ensemble RMSE and individual member RMSE histograms for the SST forecasting task at different lead-times. Here count values larger than zero indicate a greater number of ensemble predictions at a particular RMSE level. It is clear that large positive values of this metric are encountered before large negative values for all weeks – indicating that the ensemble is superior to individual members.

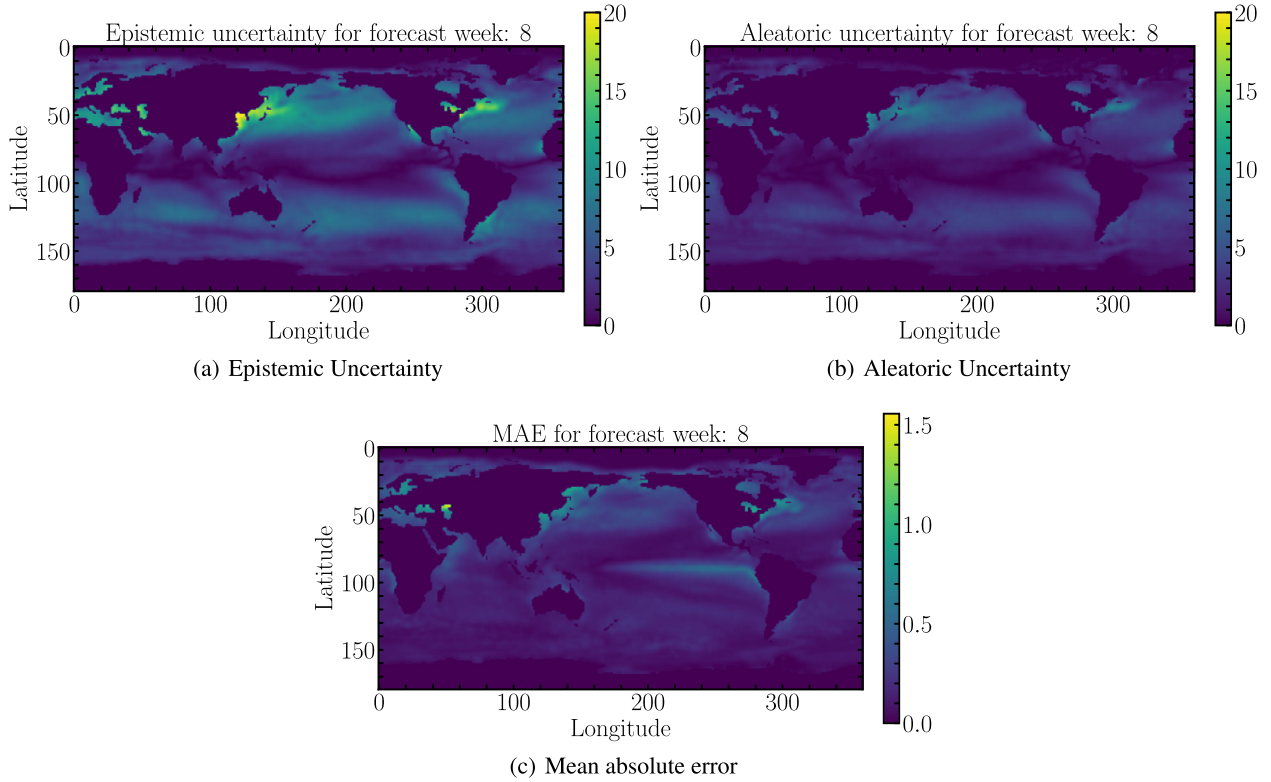


Fig. 6. Epistemic uncertainty (top-left), aleatoric uncertainty (top-right) and mean absolute errors (bottom) for the forecasting problem (showing forecasts on week 8). Uncertainty is represented by the absolute value of the standard deviation of modal coefficients projected back into the physical space.

2.4. Task 2: Field reconstruction from sparse sensors

In this section, we consider the problem of reconstructing the sea-surface temperature data given sparse observations of the flow-field. We use 20 years of data (1040 snapshots spanning 1981 to 2001) as the training data set, while the test data set is prepared from 874 snapshots spanning from year 2001 to 2018. Thus the regression problem is given by attempting to find a map

$$\mathbf{y}_t \approx \mathbf{g}^\dagger(H[\mathbf{y}_t]) \quad (8)$$

where H is the observation operator mapping the full state to sparse observations. We note that unlike the previous forecasting problem, there is no requirement of a normalization procedure on the data for constructing our data-driven models. This test setting represents extrapolation in time but not physics, since the data set has the influence of seasonal periodicity. The aforementioned problem setting follows the work of Callahan et al. [84] who attempted to reconstruct fluid-flow fields from local sensors using sparse representations. As in their article, the input sensors for the baseline model are chosen randomly from the region of 50° S to 50° N. We utilize the same configuration as our previous articles [85] that have also used this benchmark problem for algorithmic development for the purpose of consistency. We also note that our experiments indicated little sensitivity to the number of random sensor locations beyond the number of chosen here upto an order of 200 sensors, indicating that the search space of architectures and training data are an expressive combination to reconstruct the flow-field accurately. However, we do note that in complex flow-reconstruction tasks, the number of sensors required is closely connected to the prevailing length-scales in the flow-field.

3. Results

In the following section, we detail our results for the forecasting and reconstruction experiments.

3.1. Forecasting

We first assess the capability of DeepHyper NAS for discovering high-performing models that forecasting the NOAA sea-surface temperature. Here, forecasts are performed for 8-week output windows, given 8-week input information. The inputs and outputs of our model are given by sets of POD coefficients. We devise a search-space of stacked long short-term memory (LSTM) neural network models as shown in Fig. 2. In this search space, we utilize five fully-connected (i.e., stacked) LSTM cells that are then allowed to have interactions between them using skip connections. The output of this neural architecture is a sequence of coefficients (with corresponding mean and aleatoric variance) in the latent space of the dynamics.

In Fig. 3, we show results from the NAS of stacked LSTM architectures. We first show that the search for neural architectures demonstrates improving performance with increasing number of iterations of the outer-loop differential evolution algorithm with a sharp increase in the log-likelihood over the first few iterations followed by a gradual convergence. Here, we use a moving average of 25 iterations to track the convergence. We note however, that convergence is not a prerequisite for the utilization of the best performing architectures for ensemble function approximation. In addition, we also show the performance of various models obtained at the final iteration (denoted model spectrum in Fig. 3(b)), where a majority of the models have superior performance in comparison to a few that are abnormally inaccurate. A behavior as seen here also indicates a converged search.

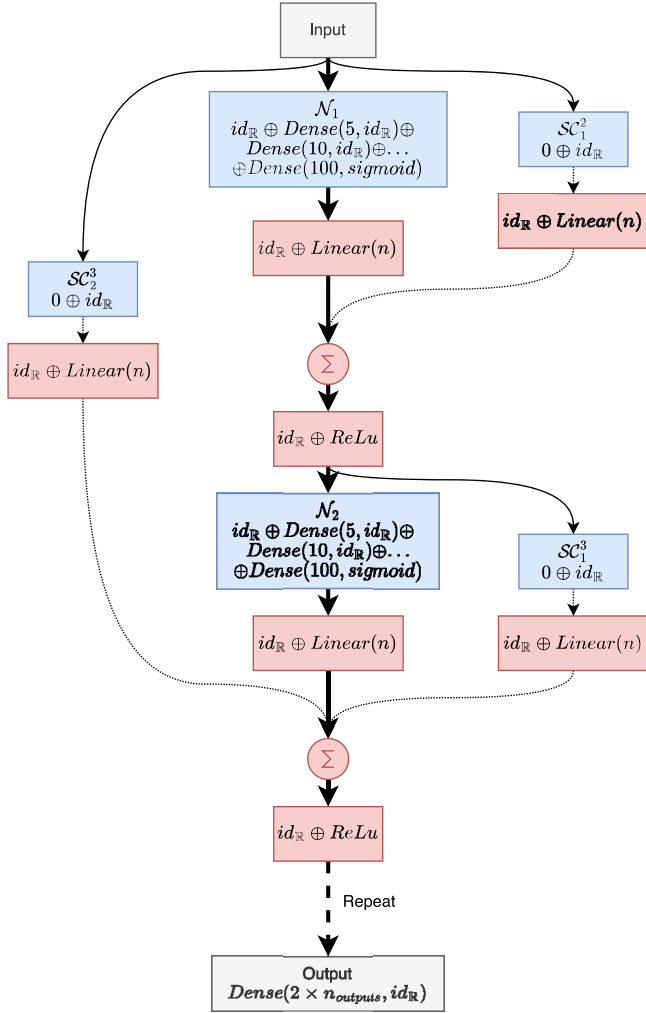


Fig. 7. A description of the neural architecture search space constructed for the flow-reconstruction task.

We extract 10 neural architectures from the entire search population and perform ensemble analyses with them. Note that the choice of the number of architectures is a user-defined parameter. We first compute the root-mean-squared-error of each neural architecture with respect to the truth for the testing range of the data set. We remind the reader that the output of a neural architecture member of the ensemble is a mean value for a Gaussian as well as a variance that represents the aleatoric uncertainty. We compute error statistics for individual ensemble members solely with the mean value predicted at the output layer. For computing the prediction of the entire ensemble, we compute a mean of the predictions from each member of the ensemble and then compute corresponding error statistics. We plot histograms of the RMSE for various weeks in Fig. 4 which demonstrate how ensemble statistics reduce the potential heavy-tailed errors of individual ensemble members (see Fig. 5).

Finally, we use the ensemble predictions from our 10 selected high-performing architectures to construct estimates of epistemic and aleatoric uncertainty from our identified surrogate models. Fig. 6 shows these plots for the final week of forecasts. A first takeaway from these plots is that Epistemic and Aleatoric uncertainty is characterized in coherent regions of the flow-field.

These coherent regions are distinctly indicative of geographical trends. In terms of accuracy, the performance of the ensemble-based predictions (over here represented by the ensemble mean) improves on the previous search model reported in Maulik et al. [74] significantly as shown in Table 1. We also mention comparisons with two physics-based models given by the Community Earth System Model (CESM) [86]² and Global Hybrid Coordinate Ocean Model (HYCOM)³ that rely on the solution of complex partial differential equations on distributed computing resources. However, we highlight the fact that the slightly larger biases in the CESM and HYCOM data may be an artifact of interpolation from grids of different spatiotemporal resolution.

From the perspective of computational cost, the search for the LSTM surrogate models in this section required 1 node-hour (8 GPU-hours) of wall-time on Theta GPU and evaluated 812 architectures sampled from the search-space. Here, it must be noted that for larger training requirements (i.e., when data sets are extremely large), smaller randomized subsets of the training data can be used to accelerate the convergence of architecture discovery before the full-data is exposed to the high-performing models. We leave such considerations for a future study. Additionally, our search procedure inherently priorities architectures that have fewer parameters since these members convergence more rapidly. In practice, this property of neural architecture search may be leveraged to find surrogates that lead to greater computational gains without sacrificing accuracy. Individual surrogate models identified are very cheap to deploy with each forward model evaluation obtained on the order of milliseconds.

3.2. Reconstruction

In this subsection, we evaluate our proposed methodology for the previously introduced benchmark flow-reconstruction problem. Our neural architecture search space builds a fully-connected network architecture from prespecified sensor input locations that measure the state, to the output which is the entire flow-field of dimension 360 times 180. The search space for sampling high-performing neural ensembles is shown in Fig. 7. In this search space, we have added 5 layers of fully-connected layers that are then allowed to interact within each other using discoverable skip connections. The training and testing data are segregated according to Maulik et al. [69], and specifically, the former is further split to obtain a validation data set that is used for directing the search for members of the ensemble.

Fig. 8(a) shows the convergence of the neural architecture search for the flow-reconstruction task. We observe an initial period of exploration which leads to several underperforming architectures but the search is soon seen to converge to a high value of the log-likelihood. Fig. 8(b) shows the range of model performance for all the sampled architectures during the search. A small proportion of the total models is seen to have poor performance, representing the initial stage of exploration by the differential evolution algorithm. As in the previous example, we retain the 10 best models obtained at the end of this search and compute statistics for RMSE for individual ensemble members as well as the prediction by the ensemble. This is shown through histogram counts in Fig. 9(a), where one can qualitatively observe that large counts in the tails in some members of the ensemble are reduced. The improvement on individual members of the ensemble can be observed in Fig. 9(b) where larger values of the difference in counts imply that the ensemble made more

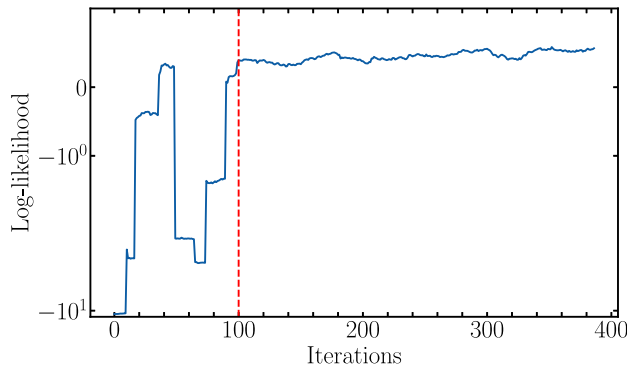
² <http://www.cesm.ucar.edu/projects/community-projects/LENS/datasets.html>

³ <https://www.ncdc.noaa.gov/data-access/model-data/model-datasets/navocean-hycom-glb>

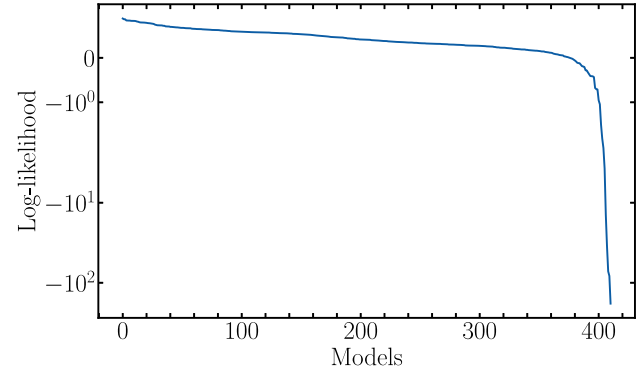
Table 1

RMSE breakdown (in Celsius) for different forecast techniques compared against the NAS-POD-LSTM forecasts between April 5, 2015, and June 24, 2018, in the Eastern Pacific region (between -10 to $+10$ degrees latitude and 200 to 250 degrees longitude). Ensemble based forecasts outperform the 'best-model' obtained via neural architecture search as reported in previous literature [87].

	RMSE ($^{\circ}\text{C}$)							
	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8
This study	0.33	0.34	0.34	0.35	0.37	0.38	0.38	0.39
Maulik et al. [74]	0.62	0.63	0.64	0.66	0.63	0.66	0.69	0.65
CESM	1.88	1.87	1.83	1.85	1.86	1.87	1.86	1.83
HYCOM	0.99	0.99	1.03	1.04	1.02	1.05	1.03	1.05

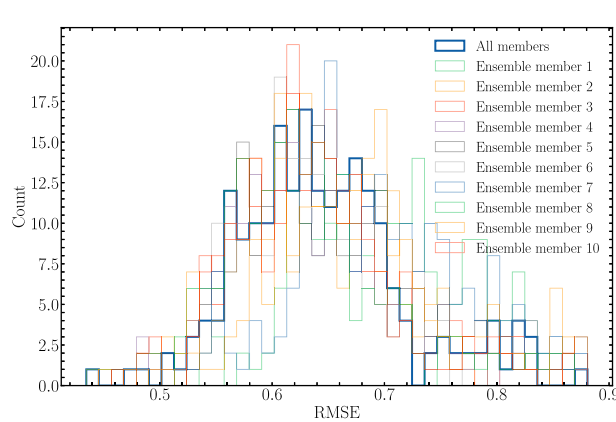


(a) Convergence

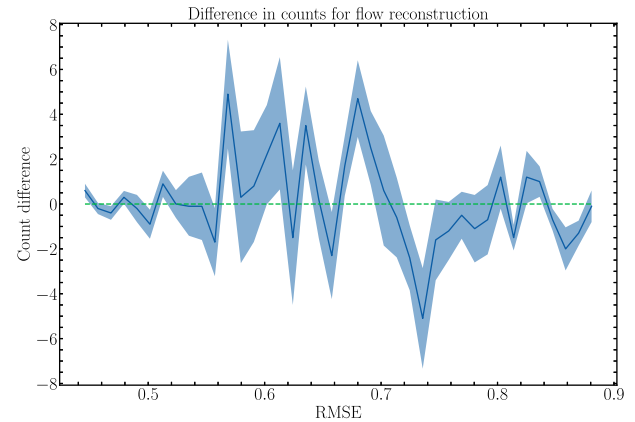


(b) Model spectrum

Fig. 8. Convergence of neural architecture search (left) and spectrum of model performance (right) for the flow-reconstruction task. The vertical dashed line at 100 iterations indicates the start of intelligent exploration of the search space.



(a) RMSE histogram



(b) RMSE count differences

Fig. 9. RMSE histogram for flow-reconstruction on test data set showing ensemble predictions as well as individual member statistics (left) and count differences between the ensemble RMSE and individual member RMSE for the SST flow reconstruction task. Here count values larger than zero indicate a greater number of ensemble predictions, i.e., there were more predictions from the ensemble with that value of an RMSE. It is clear that large positive values of this metric are encountered before large negative values – indicating that the ensemble is superior to individual members (right).

predictions with that value of the error. The desirable behavior of have large positive values at low RMSE and larger negative values at high RMSE is observed here as well. Finally averaged estimates of uncertainty are shown in Fig. 10 where it can be seen that epistemic uncertainty is mainly driven by coastal areas in the northern hemisphere whereas aleatoric uncertainty arises from the typical ENSO fluctuations in the Pacific Ocean. The metric used for assessing the performance of the ensemble is given by the relative L_2 -norm error of 0.0428 which is comparable to the model obtained in Maulik et al. [69] but with added estimates of epistemic and aleatoric uncertainty represented clearly.

For an intuitive interpretation of epistemic and aleatoric uncertainty, a low value of the former implies (as in Fig. 10) that making the neural architecture search space more complex will

not lead to greater benefits for epistemic uncertainty reduction. To elaborate, the interpretation and accuracy of uncertainty is usually limited by the quality of the ensemble, in particular, the diversity of the architectures in the ensemble. If different models in the ensemble are not diverse enough, then the uncertainty estimation does not capture the true shift in the underlying distribution. The more complex the neural architecture search space, the more diverse the models and the greater the benefit in reducing epistemic uncertainty. For aleatoric uncertainty, it is known from literature Domingos [88], that the mean squared error (MSE) exceeds the aleatoric variance by the model bias. In reality, we only estimate the aleatoric variance through our ensemble process and regions where it dramatically exceeds the MSE are considered approximate biases in the modeling strategy.

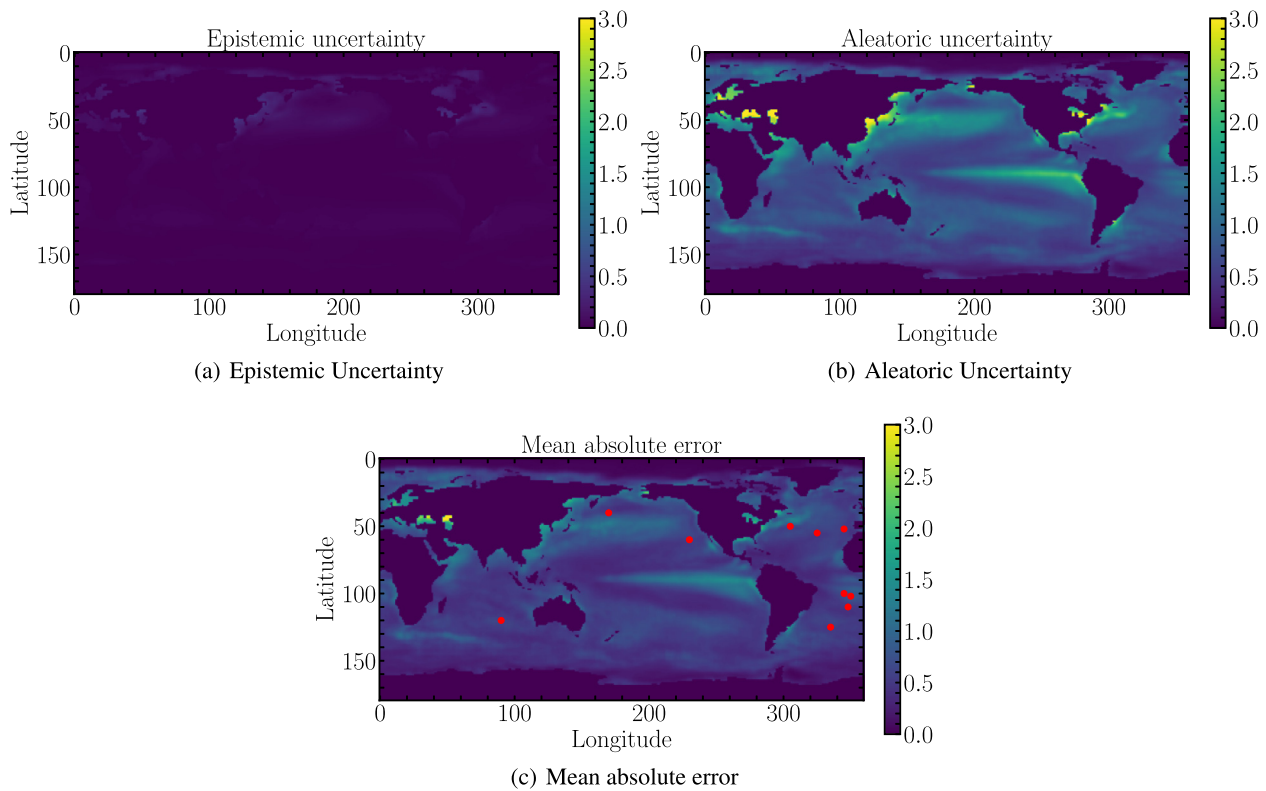


Fig. 10. Epistemic uncertainty (top-left), aleatoric uncertainty (top-right) and mean absolute errors (bottom) for the flow-reconstruction problem. The sensor locations are shown in red. One can observe that the mean-absolute errors are highly correlated with the aleatoric uncertainty.

On the other hand, regions where the MSE are much smaller than the aleatoric variance can be assumed to be regions of overfit. This is the strength of uncertainty quantification in a model regression workflow. From a computational perspective, the neural architecture search was performed using 8 GPU hours, with the total evaluation of 401 models and the model evaluation post-training was performed virtually instantaneously. A significant correlation between the error and the uncertainty is also clearly observed.

4. Conclusion

In this paper, we have demonstrated a computational workflow that may be used both for constructing deep learning models for forecasting dynamical systems and for reconstructing flow-field information from sparse sensors. Our experiments demonstrate that a joint neural architecture and hyperparameter search not only discovers effective models for target function approximation but also doubles as an effective tool to quantify uncertainty in our predictions. This is accomplished by using a number of the high-performing models discovered by the deep neural network search in an ensemble-based uncertainty quantification approach. By endowing our search space with an output layer that may use a maximum-likelihood based estimation, i.e., if the outputs are both the mean as well as variance which goes into a Gaussian likelihood maximization formulation, we are able to quantify the aleatoric uncertainty of the predictions associated with each model. By leveraging the law of decomposition of the total uncertainty of an ensemble, we are also able to quantify the aleatoric and epistemic uncertainty in predictions of the entire ensemble of members. Our data set for experiments comes from the NOAA optimum interpolation sea-surface temperature data set, thereby representing a representative data set for complex geoscience data set. Our conclusions suggest that scalable distributed neural architecture and hyperparameter search may

represent a vertically integrated workflow for high-performing surrogate model construction as well as efficient uncertainty quantification.

Code availability

Sample codes for neural architecture search as well as ensemble based predictions will be made available after peer-review (or on request). Tutorials and examples for deploying the neural architecture and hyperparameter searches for arbitrary modeling tasks can be found at <https://github.com/deephyper/deephyper>.

CRediT authorship contribution statement

Romit Maulik: Conceptualization, Methodology, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Romain Egele:** Conceptualization, Methodology, Software, Formal analysis, Writing – original draft, Writing – review & editing. **Krishnan Raghavan:** Conceptualization, Methodology, Formal analysis, Writing – original draft, Writing – review & editing. **Prasanna Balaprakash:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing, Project administration, Funding acquisition, Supervision.

Declaration of competing interest

The authors declare no conflict of interest.

Data availability

Training data used in the present study are available publicly on Google Drive: Example 1 (NOAA sea surface temperature - forecasting):

<https://drive.google.com/drive/folders/1pVW4epkeHkT2WHZB7Dym5IURcfOP4cXu?usp=sharing>,

Example 2 (NOAA sea surface temperature - reconstruction):
<https://drive.google.com/drive/folders/1pVW4epkeHkT2WHZB7Dym5IURcfOP4cXu?usp=sharing>,

Acknowledgments

This work was supported by the U.S. Department of Energy (DOE), Office of Science, Office of Advanced Scientific Computing Research (ASCR), under Contract DE-AC02-06CH11357. We acknowledge funding support from ASCR for DOE-FOA-2493 “Data-intensive scientific machine learning” and the DOE Early Career Research Program award. This research was funded in part and used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357.

References

- [1] Rick Stevens, Valerie Taylor, Jeff Nichols, Arthur Barney Maccabe, Katherine Yelick, David Brown, AI for Science: Report on the department of energy (DOE) town halls on artificial intelligence (AI) for science, Technical report, Argonne National Lab.(ANL), Argonne, IL (United States), 2020.
- [2] Ricardo Vinuesa, Steven L. Brunton, The potential of machine learning to enhance computational fluid dynamics, 2021, pp. 1–13, arXiv preprint arXiv:2110.02085.
- [3] Cynthia Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, *Nature Mach. Intell.* 1 (5) (2019) 206–215.
- [4] Apostolos F Psaros, Xuhui Meng, Zongren Zou, Ling Guo, George Em Karniadakis, Uncertainty quantification in scientific machine learning: Methods, metrics, and comparisons, *J. Comput. Phys.* (2023) 111902.
- [5] Balaji Lakshminarayanan, Alexander Pritzel, Charles Blundell, Simple and scalable predictive uncertainty estimation using deep ensembles, 2016, arXiv preprint arXiv:1612.01474.
- [6] Romain Egele, Romit Maulik, Krishnan Raghavan, Bethany Lusch, Isabelle Guyon, Prasanna Balaprakash, Autodeuq: Automated deep ensemble with uncertainty quantification, in: 2022 26th International Conference on Pattern Recognition (ICPR), IEEE, 2022, pp. 1908–1914.
- [7] Kevin Carlberg, Charbel Bou-Mosleh, Charbel Farhat, Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations, *Internat. J. Numer. Methods Engrg.* 86 (2) (2011) 155–181.
- [8] Z. Wang, I. Akhtar, J. Borggaard, T. Iliescu, Proper orthogonal decomposition closure models for turbulent flows: a numerical comparison, *Comput. Meth. Appl. M.* 237 (2012) 10–26.
- [9] O. San, J. Borggaard, Principal interval decomposition framework for POD reduced-order modeling of convective Boussinesq flows, *Int. J. Numer. Meth. Fl.* 78 (1) (2015) 37–62.
- [10] Francesco Ballarin, Andrea Manzoni, Alfio Quarteroni, Gianluigi Rozza, Supremizer stabilization of POD–Galerkin approximation of parametrized steady incompressible Navier–Stokes equations, *Internat. J. Numer. Methods Engrg.* 102 (5) (2015) 1136–1161.
- [11] Omer San, Romit Maulik, Extreme learning machine for reduced order modeling of turbulent geophysical flows, *Phys. Rev. E* 97 (4) (2018) 042322.
- [12] Q. Wang, J.S. Hesthaven, D. Ray, Non-intrusive reduced order modeling of unsteady flows using artificial neural networks with application to a combustion problem, *J. Comput. Phys.* 384 (2019) 289–307.
- [13] Youngsoo Choi, Kevin Carlberg, Space-time least-squares Petrov–Galerkin projection for nonlinear model reduction, *SIAM J. Sci. Comput.* 41 (1) (2019) A26–A58.
- [14] S. Ashwin Renganathan, Romit Maulik, Vishwas Rao, Machine learning for nonintrusive model order reduction of the parametric inviscid transonic flow past an airfoil, *Phys. Fluids* 32 (4) (2020) 047110.
- [15] Jiahao Ren, Haiou Wang, Jiangkuan Xing, Kun Luo, Jianren Fan, A lower-dimensional approximation model of turbulent flame stretch and its related quantities with machine learning approaches, *Phys. Fluids* 32 (11) (2020) 115113.
- [16] J.L. Proctor, S.L. Brunton, J.N. Kutz, Dynamic mode decomposition with control, *SIAM J. Appl. Dyn. Syst.* 15 (1) (2016) 142–161.
- [17] Sebastian Peitz, Sina Ober-Blobaum, Michael Dellnitz, Multiobjective optimal control methods for the navier-stokes equations using reduced order modeling, *Acta Appl. Math.* 161 (1) (2019) 171–199.
- [18] Bernd R. Noack, Marek Morzynski, Gilead Tadmor, *Reduced-Order Modelling for Flow Control*, vol. 528, Springer Science & Business Media, 2011.
- [19] C.W. Rowley, S.T.M. Dawson, Model reduction for flow analysis and control, *Annu. Rev. Fluid Mech.* 49 (2017) 387–417.
- [20] Cedric Raibaud, Peng Zhong, Bernd R Noack, Robert John Martinuzzi, Machine learning strategies applied to the control of a fluidic pinball, *Phys. Fluids* 32 (1) (2020) 015108.
- [21] Feng Ren, Hai-bao Hu, Hui Tang, Active flow control using machine learning: A brief review, *J. Hydrodyn.* 32 (2) (2020) 247–253.
- [22] Benjamin Peherstorfer, Karen Willcox, Max Gunzburger, Optimal model management for multifidelity Monte Carlo estimation, *SIAM J. Sci. Comput.* 38 (5) (2016) A3163–A3194.
- [23] Dewei Fan, Bingfu Zhang, Yu Zhou, Bernd R. Noack, Optimization and sensitivity analysis of active drag reduction of a square-back airfoil using machine learning control, *Phys. Fluids* 32 (12) (2020) 125117.
- [24] Themistoklis P. Sapsis, Andrew J. Majda, Statistically accurate low-order models for uncertainty quantification in turbulent dynamical systems, *Proc. Natl. Acad. Sci. USA* 110 (34) (2013) 13705–13710.
- [25] Matthew J. Zahr, Kevin T. Carlberg, Drew P. Kouri, An efficient, globally convergent method for optimization under uncertainty using adaptive model reduction and sparse grids, 2018, arXiv preprint arXiv:1811.00177.
- [26] Hwan Goh, Sheroze Sherifdeen, Jonathan Wittmer, Tan Bui-Thanh, Solving Bayesian inverse problems via variational autoencoders, 2019, arXiv preprint arXiv:1912.04212.
- [27] Rossella Arcucci, Laetitia Mottet, Christopher Pain, Yi-Ke Guo, Optimal reduced space for variational data assimilation, *J. Comput. Phys.* 379 (2019) 51–69.
- [28] Meng Tang, Yimin Liu, Louis J. Durlofsky, A deep-learning-based surrogate model for data assimilation in dynamic subsurface flow problems, *J. Comput. Phys.* (2020) 109456.
- [29] César Quilodrán Casas, Rossella Arcucci, Pin Wu, Christopher Pain, Yi-Ke Guo, A reduced order deep data assimilation model, *Physica D* 412 (2020) 132615.
- [30] Romit Maulik, Vishwas Rao, Jiali Wang, Gianmarco Mengaldo, Emil Constantinescu, Bethany Lusch, Prasanna Balaprakash, Ian Foster, Rao Kotamarthi, Efficient high-dimensional variational data assimilation with machine-learned reduced-order models, *Geosci. Model Dev.* 15 (8) (2022) 3433–3445.
- [31] Omer San, Jeff Borggaard, Basis selection and closure for POD models of convection dominated Boussinesq flows, in: 21st International Symposium on Mathematical Theory of Networks and Systems, vol. 5 no. 3, 2014.
- [32] Milan Korda, Mihai Putinar, Igor Mezić, Data-driven spectral analysis of the Koopman operator, *Appl. Comput. Harmon. Anal.* (2018).
- [33] Virginia L. Kalb, Anil E. Deane, An intrinsic stabilization scheme for proper orthogonal decomposition based low-dimensional models, *Phys. Fluids* 19 (5) (2007) 054106.
- [34] I. Kalashnikova, M.F. Barone, On the stability and convergence of a Galerkin reduced order model (ROM) of compressible flow with solid wall and far-field boundary treatment, *Internat. J. Numer. Methods Engrg.* 83 (10) (2010) 1345–1375.
- [35] Muhammad Mohebujjaman, Leo G. Rebholz, Traian Iliescu, Physically constrained data-driven correction for reduced-order modeling of fluid flows, *Int. J. Numer. Meth. Fl.* 89 (3) (2019) 103–122.
- [36] D. Xiao, F. Fang, J. Du, C.C. Pain, I.M. Navon, A.G. Buchan, Ahmed H. Elsheikh, G. Hu, Non-linear Petrov–Galerkin methods for reduced order modelling of the Navier–Stokes equations using a mixed finite element pair, *Comput. Methods Appl. Mech.* 255 (2013) 147–157.
- [37] Fangxin Fang, Christopher C Pain, I.M. Navon, Ahmed H. Elsheikh, Juan Du, D. Xiao, Non-linear Petrov–Galerkin methods for reduced order hyperbolic equations and discontinuous finite element methods, *J. Comput. Phys.* 234 (2013) 540–559.
- [38] Kevin Carlberg, Matthew Barone, Harbir Antil, Galerkin v. least-squares Petrov–Galerkin projection in nonlinear model reduction, *J. Comput. Phys.* 330 (2017) 693–734.
- [39] Zhongzhan Huang, Senwei Liang, Hong Zhang, Haizhao Yang, Liang Lin, Accelerating numerical solvers for large-scale simulation of dynamical system via NeurVec, 2022, arXiv preprint arXiv:2208.03680.
- [40] Sepp Hochreiter, Jürgen Schmidhuber, Long short-term memory, *Neural Comput.* 9 (8) (1997) 1735–1780.
- [41] P.R. Vlachas, W. Byeon, Z.Y. Wan, T.P. Sapsis, P. Koumoutsakos, Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks, *Proc. R. Soc. A Math. Phys. Eng. Sci.* 474 (2213) (2018) 20170844.
- [42] S.E. Ahmed, O. San, A. Rasheed, T. Iliescu, A long short-term memory embedding for hybrid uplifted reduced order models, 2019, arXiv preprint, arXiv:1912.06756.
- [43] Romit Maulik, Arvind Mohan, Bethany Lusch, Sandeep Madireddy, Prasanna Balaprakash, Time-series learning of latent-space dynamics for reduced-order model closure, 2019, arXiv preprint arXiv:1906.07815.

- [44] A.T. Mohan, D. Daniel, M. Chertkov, D. Livescu, Compressed convolutional LSTM: An efficient deep learning framework to model high fidelity 3D turbulence, 2019, arXiv preprint, [arXiv:1903.00033](https://arxiv.org/abs/1903.00033).
- [45] Arvind T. Mohan, Datta V. Gaitonde, A deep learning based approach to reduced order modeling for turbulent flow control using LSTM neural networks, 2018, arXiv preprint [arXiv:1804.09269](https://arxiv.org/abs/1804.09269).
- [46] F.J. Gonzalez, M. Balajewicz, Learning low-dimensional feature dynamics using deep convolutional recurrent autoencoders, 2018, arXiv preprint, [arXiv:1808.01346](https://arxiv.org/abs/1808.01346).
- [47] Kazuto Hasegawa, Kai Fukami, Takaaki Murata, Koji Fukagata, CNN-LSTM based reduced order modeling of two-dimensional unsteady flows around a circular cylinder at different Reynolds numbers, *Fluid Dyn. Res.* 52 (6) (2020) 065501.
- [48] Qian Wang, Nicolò Ripamonti, Jan S. Hesthaven, Recurrent neural network closure of parametric POD-Galerkin reduced-order models based on the Mori-Zwanzig formalism, *J. Comput. Phys.* 410 (2020) 109402.
- [49] Kazuto Hasegawa, Kai Fukami, Takaaki Murata, Koji Fukagata, Machine-learning-based reduced-order modeling for unsteady flows around bluff bodies of various shapes, *Theor. Comput. Fluid Dyn.* 34 (2020) 367–383.
- [50] Ashesh Chattopadhyay, Adam Subel, Pedram Hassanzadeh, Data-Driven Super-Parameterization Using Deep Learning: Experimentation With Multiscale Lorenz 96 Systems and Transfer Learning, *J. Adv. Modelling Earth Syst.* 12 (11) (2020) e2020MS002084.
- [51] Romit Maulik, Bethany Lusch, Prasanna Balaprakash, Non-autoregressive time-series methods for stable parametric reduced-order models, *Phys. Fluids* 32 (8) (2020) 087115.
- [52] S. Ashwin Renganathan, Romit Maulik, Jai Ahuja, Enhanced data efficiency using deep neural networks and Gaussian processes for aerodynamic design optimization, *Aerosp. Sci. Technol.* 111 (2021) 106522.
- [53] Zhan Ma, Wenxiao Pan, Data-driven nonintrusive reduced order modeling for dynamical systems with moving boundaries using Gaussian process regression, *Comput. Methods Appl. Mech. Engrg.* 373 (2021) 113495.
- [54] Mengwu Guo, Jan S. Hesthaven, Reduced order modeling for nonlinear structural analysis using Gaussian process regression, *Comput. Methods Appl. Mech. Engrg.* 341 (2018) 807–826.
- [55] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, Illia Polosukhin, Attention is all you need, *Adv Neural Inform Process Syst* 30 (2017).
- [56] Nicholas Geneva, Nicholas Zabarar, Transformers for modeling physical systems, *Neural Netw.* 146 (2022) 272–289.
- [57] Alec J Linot, Joshua W Burby, Qi Tang, Prasanna Balaprakash, Michael D Graham, Romit Maulik, Stabilized neural ordinary differential equations for long-time forecasting of dynamical systems, *J. Comput. Phys.* 474 (2023) 111838.
- [58] R.J. Adrian, P. Moin, Stochastic estimation of organized turbulent structure: homogeneous shear flow, *J. Fluid Mech.* 190 (1988) 531–559.
- [59] Richard Everson, Lawrence Sirovich, Karhunen–Loève procedure for gappy data, *J. Opt. Soc. Amer. A* 12 (8) (1995) 1657–1664.
- [60] Kai Fukami, Koji Fukagata, Kunihiro Taira, Assessment of supervised machine learning methods for fluid flows, *Theor. Comput. Fluid Dyn.* 34 (2020) 497–519.
- [61] Douglas W Carter, Francis De Voogt, Renan Soares, Bharathram Ganapathisubramani, Data-driven sparse reconstruction of flow over a stalled aerofoil using experimental data, *Data-Centric Eng.* 2 (2021) e5.
- [62] N Benjamin Erichson, Lionel Mathelin, Zhewei Yao, Steven L Brunton, Michael W Mahoney, J Nathan Kutz, Shallow neural networks for fluid flow reconstruction with limited sensors, *Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.* 476 (2238) (2020) 20200097.
- [63] Luning Sun, Jian-Xun Wang, Physics-constrained bayesian neural network for fluid flow reconstruction with sparse and noisy data, *Theor. Appl. Mech. Lett.* 10 (3) (2020) 161–169.
- [64] Pierre Dubois, Thomas Gomez, Laurent Planckaert, Laurent Perret, Machine learning for fluid flow reconstruction from limited measurements, *J. Comput. Phys.* 448 (2022) 110733.
- [65] Sheroze Sherifdeen, Jean C Ragusa, Jim E Morel, Marvin L Adams, Tan Bui-Thanh, Accelerating PDE-constrained inverse solutions with deep learning and reduced order models, 2019, arXiv preprint [arXiv:1912.08864](https://arxiv.org/abs/1912.08864).
- [66] Paul G. Constantine, Carson Kent, Tan Bui-Thanh, Accelerating Markov chain Monte Carlo with active subspaces, *SIAM J. Sci. Comput.* 38 (5) (2016) A2779–A2805.
- [67] Yang Liu, Dewei Wang, Xiaodong Sun, Nam Dinh, Rui Hu, Uncertainty quantification for multiphase-CFD simulations of bubbly flows: a machine learning-based Bayesian approach supported by high-resolution experiments, *Reliab. Eng. Syst. Saf.* 212 (2021) 107636.
- [68] Kjetil O. Lye, Siddhartha Mishra, Deep Ray, Deep learning observables in computational fluid dynamics, *J. Comput. Phys.* 410 (2020) 109339.
- [69] Romit Maulik, Kai Fukami, Nesar Ramachandra, Koji Fukagata, Kunihiro Taira, Probabilistic neural networks for fluid flow surrogate modeling and data recovery, *Phys. Rev. Fluids* 5 (10) (2020) 104401.
- [70] Masaki Morimoto, Kai Fukami, Romit Maulik, Ricardo Vinuesa, Koji Fukagata, Assessments of epistemic uncertainty using Gaussian stochastic weight averaging for fluid-flow regression, *Physica D* 440 (2022) 133454.
- [71] Aditi Krishnapriyan, Amir Gholami, Shandian Zhe, Robert Kirby, Michael W Mahoney, Characterizing possible failure modes in physics-informed neural networks, *Adv. Neural Inf. Process. Syst.* 34 (2021) 26548–26560.
- [72] Yin hao Zhu, Nicholas Zabarar, Phaedon-Stelios Koutsourelakis, Paris Perdikaris, Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data, *J. Comput. Phys.* 394 (2019) 56–81.
- [73] Suraj Pawar, Omer San, Prakash Vedula, Adil Rasheed, Trond Kvamsdal, Multi-fidelity information fusion with concatenated neural networks, *Sci. Rep.* 12 (1) (2022) 5900.
- [74] Romit Maulik, Romain Egele, Bethany Lusch, Prasanna Balaprakash, Recurrent neural network architecture search for geophysical emulation, in: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC '20*, IEEE Press, 2020.
- [75] Matthew R. Rudary, On Predictive Linear Gaussian Models, University of Michigan, 2009, pp. 385–386.
- [76] Romain Egele, Prasanna Balaprakash, Venkatram Vishwanath, Isabelle Guyon, Zhengying Liu, AgEBO-Tabular: joint neural architecture and hyperparameter search with autotuned data-parallel training for tabular data, in: *SC21: International Conference for High Performance Computing, Networking, Storage and Analysis*, 2021, pp. 1–14, <http://dx.doi.org/10.1145/3458817.3476203>.
- [77] Esteban Real, Alok Aggarwal, Yanping Huang, Quoc V. Le, Regularized evolution for image classifier architecture search, in: *Proceedings of the Aaai Conference on Artificial Intelligence*, vol. 33 no. 01, 2019, pp. 4780–4789.
- [78] David Ginsbourger, Rodolphe Le Riche, Laurent Carraro, Kriging is well-suited to parallelize optimization, in: Yoel Tenne, Chi-Keong Goh (Eds.), *Computational Intelligence in Expensive Optimization Problems*, vol. 2, Adaptation Learning and Optimization, Springer Berlin Heidelberg, 2010, pp. 131–162, http://dx.doi.org/10.1007/978-3-642-10701-6_6, URL http://link.springer.com/10.1007/978-3-642-10701-6_6Series Title.
- [79] Rich Caruana, Alexandru Niculescu-Mizil, Geoff Crew, Alex Skikes, Ensemble selection from libraries of models, in: *Twenty-First International Conference on Machine Learning - ICML '04*, ACM Press, 2004, p. 18, <http://dx.doi.org/10.1145/1015330.1015432>, URL <http://portal.acm.org/citation.cfm?doid=1015330.1015432>.
- [80] P. Balaprakash, M. Salim, T. Uram, V. Vishwanath, S. Wild, DeepHyper: Asynchronous hyperparameter search for deep neural networks, in: *2018 IEEE 25th International Conference on High Performance Computing (HiPC)*, IEEE, 2018, pp. 42–51.
- [81] J. Nathan Kutz, Xing Fu, Steven L. Brunton, Multiresolution dynamic mode decomposition, *SIAM J. Appl. Dyn. Syst.* 15 (2) (2016) 713–735.
- [82] Jared L. Callahan, Kazuki Maeda, Steven L. Brunton, Robust flow reconstruction from limited measurements via sparse representation, *Phys. Rev. Fluids* 4 (10) (2019) 103907.
- [83] Kunihiro Taira, Maziar S. Hemati, Steven L. Brunton, Yiyang Sun, Karthik Duraisamy, Shervin Bagheri, Scott T.M. Dawson, Chi-An Yeh, Modal analysis of fluid flows: Applications and outlook, *AIAA J.* 58 (3) (2020) 998–1022, <http://dx.doi.org/10.2514/1.J058462>.
- [84] J.L. Callahan, K. Maeda, S.L. Brunton, Robust flow reconstruction from limited measurements via sparse representation, *Phys. Rev. Fluids* 4 (2019) 103907.
- [85] Romit Maulik, Nathan A Garland, Joshua W Burby, Xian-Zhu Tang, Prasanna Balaprakash, Neural network representability of fully ionized plasma fluid model closures, *Phys. Plasmas* 27 (7) (2020) 072106.
- [86] Jennifer E Kay, Clara Deser, A Phillips, A Mai, Cecile Hannay, Gary Strand, Julie Michelle Arblaster, SC Bates, Gokhan Danabasoglu, J Edwards, et al., The Community Earth System Model (CESM) large ensemble project: A community resource for studying climate change in the presence of internal climate variability, *Bull. Am. Meteorol. Soc.* 96 (8) (2015) 1333–1349.
- [87] Romit Maulik, Themistoklis Botsas, Nesar Ramachandra, Lachlan Robert Mason, Indranil Pan, Latent-space time evolution of non-intrusive reduced-order models using Gaussian process emulation, *Physica D* (2020) 132797.
- [88] Pedro Domingos, A unified bias-variance decomposition, in: *Proceedings of 17th International Conference on Machine Learning*, Morgan Kaufmann Stanford, 2000, pp. 231–238.