Uncertainty-aware predictions of molecular x-ray absorption spectra using neural network ensembles

Animesh Ghose,[1] Mikhail Segal,[1] Fanchen Meng,[2] Zhu Liang,[2] Mark S. Hybertsen,[2] Xiaohui Qu,[2] Eli Stavitski,[3] Shinjae Yoo,[1] Deyu Lu,[2,*] and Matthew R. Carbone[1,†]

[1]*Computational Science Initiative, Brookhaven National Laboratory, Upton, New York 11973, USA*
[2]*Center for Functional Nanomaterials, Brookhaven National Laboratory, Upton, New York 11973, USA*
[3]*National Synchrotron Light Source II, Brookhaven National Laboratory, Upton, New York 11973, USA*

(Received 28 October 2022; accepted 23 January 2023; published 15 March 2023)

As machine learning (ML) methods continue to be applied to a broad scope of problems in the physical sciences, uncertainty quantification is becoming correspondingly more important for their robust application. Uncertainty-aware machine learning methods have been used in select applications, but largely for scalar properties. In this work, we showcase an exemplary study in which neural network ensembles are used to predict the x-ray absorption spectra of small molecules, as well as their pointwise uncertainty, from local atomic environments. The performance of the resulting surrogate clearly demonstrates quantitative correlation between errors relative to ground truth and the predicted uncertainty estimates. Significantly, the model provides an upper bound on the expected error. Specifically, an important quality of this uncertainty-aware model is that it can indicate when the model is predicting on out-of-sample data. This allows for its integration with large-scale sampling of structures together with active learning or other techniques for structure refinement. Additionally, our models can be generalized to larger molecules than those used for training, and also successfully track uncertainty due to random distortions in test molecules. While we demonstrate this workflow on a specific example, ensemble learning is completely general. We believe it could have significant impact on ML-enabled forward modeling of a broad array of molecular and materials properties.

DOI: 10.1103/PhysRevResearch.5.013180

I. INTRODUCTION

Recent years have witnessed the emergence of a thriving research enterprise directed towards the application of data-driven science to condensed matter physics, chemistry, and materials science [1,2]. In particular, machine learning (ML) models such as artificial neural networks, which are universal approximators that in principle can fit any function, have been widely used to model complex relationship among physical quantities. The intersection of ML tools, emerging high-performance computing platforms and a growing number of large open source datasets has made a transformative impact on research in the physical sciences.

In the context of first-principles simulations, it has been demonstrated that ML can be used to predict molecular or materials properties from atomistic structure at comparable accuracy to the quantum mechanical theories used to produce their training data, but at only a tiny fraction of their computational cost [1,3–5]. As a result, ML has the potential to tremendously accelerate computational studies, bridge first-principles simulations to a larger time and length scales, and enable efficient materials discovery pipelines [6]. Similarly, ML surrogate models can also be used to bypass the numerical solution [7,8] and to explore the quantum states [9,10] of model Hamiltonians.

While a trained ML model provides a prediction, it usually does not provide a measure of its confidence, despite the crucial importance of model uncertainty for the researchers who apply them. In particular, ML models are designed to make accurate predictions on inputs sampled from the same distribution as the training set. However, they often fail completely when tasked with predicting on data sampled from a different distribution. Importantly, it is not always obvious (or detectable via some heuristic) when a model is performing inference on an out-of-sample input. In order to detect when this happens, one needs methodologies that incorporate uncertainty quantification (UQ). These are broadly classified as predictive methodologies that include accurate estimates of different types of statistical uncertainty. In the domain of ML and surrogate modeling, this often refers to the ability of trained models to provide some measure of confidence in the accuracy of their predictions [11]. In research scenarios where out-of-sample data are likely to be frequently encountered, the ability of ML models to perform UQ becomes crucial.

Understanding model confidence is a key piece of the ML pipeline. Most research work simply evaluates model performance on a testing set, treating that as a proxy for

*dlu@bnl.gov
†mcarbone@bnl.gov

understanding on-the-fly model performance. UQ takes this one step further. For example, UQ is a critical component of Gaussian process-based [12] and neural network ensemble (NNE)-based Bayesian optimization, which has been employed for the autonomous design of experiments in many different domains, from the design of nanopaticles via flow reactors [13–15] to the optimization of mechanical properties of materials [16,17]. Neural network potentials (NNPs) [3,18–23] often utilize UQ to predict where their models are failing, and where they require retraining [24,25] (known as active learning [26]). Other works apply UQ in active learning for data-efficient prediction of molecular properties, such as the enthalpy, atomization energy, polarizability, and HOMO/LUMO energy levels [27]. However, existing uncertainty-aware (UA) models are mostly limited to several specific topics and not broadly applied in ML applications.

In this study, we present a NNE method for quantifying the uncertainty of predicted *vector* targets. Specifically, we use local atomic environment information to predict the x-ray absorption spectra (XAS) of small molecules. From existing literature on UQ implementations in ML models, very little can be discerned about their performance on spectral functions and vector targets in general, as it is unclear how the standard aforementioned applications would generalize from predicting scalars to a much higher dimensional space. We consider this XAS problem as a case study, but note that our approach is completely general. It can be applied in the broader context of *any* molecular or materials property. We will show that our NNEs are not only capable of making quantitatively accurate predictions of the XAS spectra, but also of making accurate estimates of the pointwise uncertainty of said predictions.

XAS is a widely used element-specific materials characterization technique that is sensitive to the local chemical environment of the absorbing sites [28–33]. However, interpreting XAS data is nontrivial. While some important, physically motivated heuristics are well known, full understanding of the relationship between spectra and underlying atomic-scale structure is mediated by electronic states. In particular, first-principles XAS simulations are playing an essential role in XAS analysis, allowing the interpretation of precise structure-property relationships otherwise much more challenging to resolve experimentally. In an XAS simulation, the spectrum is calculated from the atomic arrangement of the system. Depending on the complexity of the theory and system, the spectral simulation can be prohibitively time consuming. This limits its use for fast structure screening/refinement or spectral feature assignment. As a result, there is a growing interest to develop surrogate models that can predict XAS spectra, and other types of spectral targets, from atomistic structures [34].

We focus on the near-edge portion of the x-ray absorption spectrum, known as x-ray absorption near edge structure (XANES). Specifically we consider K-edge XANES, corresponding to excitation from a 1s core orbital electron to empty orbitals or the continuum. We simulate the K-edge XANES spectra of C, N, and O atoms in small molecular systems for a large database of density functional theory (DFT)-relaxed molecular structures: QM9 [35]. Taking a divide-and-conquer approach, the NNEs are trained only on the local environments of individual absorbing atoms, and molecular spectra are constructed by averaging the predictions of the individual absorption sites. Errors are interpreted as standard deviations and propagated accordingly. Figure 1 demonstrates the overall workflow. This allows the NNEs to make predictions on molecules larger than what the models were trained on (similar to how neural network potentials can generalize to larger systems if correlations are sufficiently short-range). As such, this approach could have broad implications in the fields of inverse design and generalizable surrogate modeling.

Key to any UQ methodology is understanding how trained models perform when data are pushed out-of-sample during inference. The addition of UQ to a surrogate model supports its generalized use. In the context of molecular systems specifically, we enumerate four physically motivated classes of generalization in which a UQ methodology can flag adverse effects of the changing local environment on model performance:

(1) *chemical*, e.g., including larger molecules relative to the training set,

(2) *configurational*, e.g., including structural distortions from equilibrium geometry (such as due to thermal effects),

(3) *electronic*, e.g., introducing new chemical motifs (such as aromaticity) that are not necessarily a function of molecular size,

(4) *environmental*, e.g., introducing molecule-solvant interaction due to solvation.

In principle, UQ methods should be able to detect when out-of-sample data due to any of the above situations occurs. In this work, we directly study how new distributions of testing data due to chemical and configurational changes affect model and UQ performance. Studies of electronic and environmental effects are beyond the scope of the current work.

As noted, to date, incorporation of UQ into ML applications for materials science and chemistry has been limited in scope, particularly focusing on scalar target quantities (we highlight a notable exception, in which various UQ-enabled ML methods were used to predict the x-ray *emission* spectra of transition metal complexes [36]). In the present work, with our focus on spectroscopy, we explore the challenges associated with incorporating UQ into models where the target space is of a substantially higher dimension, corresponding to the vector of spectroscopic intensity versus x-ray photon energy. In particular, this is the first time UQ methods and the requisite analysis have been demonstrated for XAS prediction.

The manuscript is organized as follows. In Sec. II, we outline the procedures used for constructing our databases, and describe the featurization and forward modeling. This includes a brief discussion on our choice of local feature embedding and various forms of UQ. Next, in Sec. III, we outline our procedures for analyzing the database using unsupervised and general data-analysis techniques, as well as final data preparations for machine learning. Section IV contains the main results of this work, and demonstrates the ability of our trained models to make accurate predictions and perform UQ on different subsets of small molecules. Section IV B specifically highlights the power of our models to generalize to molecules with more atoms than those used during training. In Section IV C, we also discuss the NNE's ability to accurately

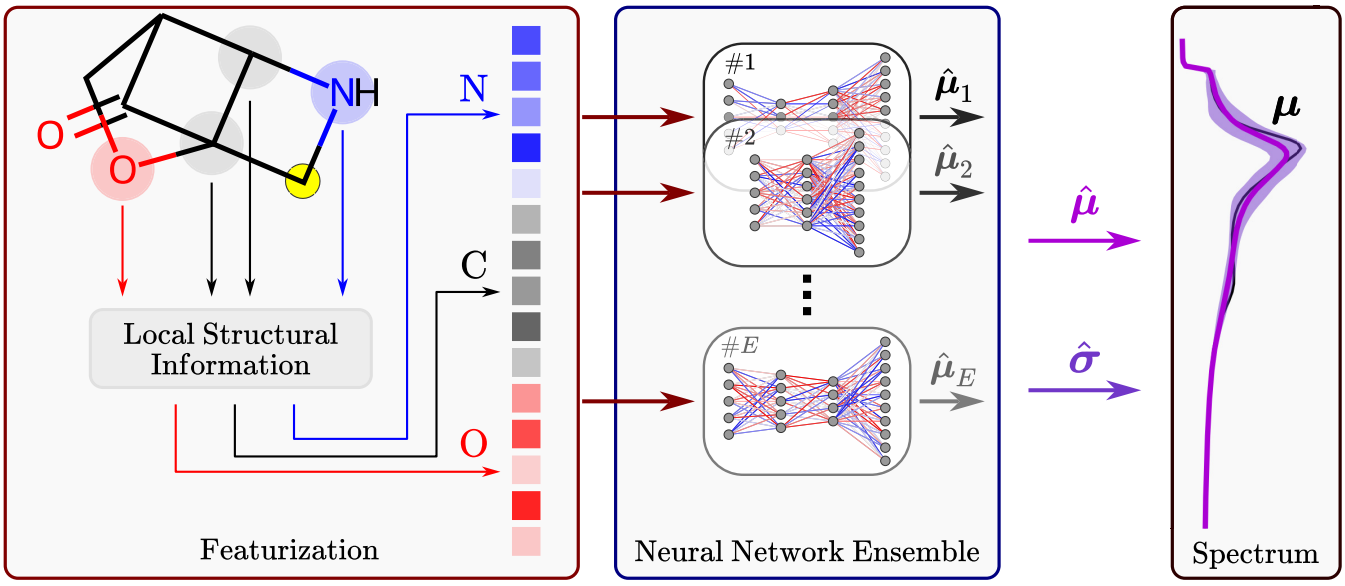

FIG. 1. A cartoon of our workflow showcasing the three required steps for inference using the NNE method on a molecule in its optimized geometry. *Featurization*: the atom type-resolved local structural information (red, gray and blue highlights) about the absorbing atom (yellow highlight) is encoded into a fixed-length vector representation. Fluorine and hydrogen are also encoded during featurization (for visual clarity, these are not shown in this example). *Neural Network Ensemble*: each of the estimators receives the identical input vector, and each outputs a spectrum. The results are averaged over the NNE, and the spread determined. Each estimator is depicted above as having 5 inputs and 10 outputs. However, in this work, input vectors have $M = 155$ elements, and the output spectra have $M' = 200$ elements. *Spectrum*: the predicted spectra $\hat{\boldsymbol{\mu}}$ and the estimate for its uncertainty $\hat{\boldsymbol{\sigma}}$ are shown, with the ground truth $\boldsymbol{\mu}$ for reference.

quantify uncertainty when tested on structures which are not in their relaxed geometries, a different type of generalization from the training database. Finally, we conclude and discuss the outlook and future plans of this work in Sec. V.

II. THEORY

The general theory of ensemble learning and UQ has a rich history in the ML literature. In this section, we summarize previous work and highlight the key principles and theory behind ensemble learning and UQ. Specifically, we provide an overview of supervised learning (and feed-forward neural networks), UQ, and ensembling in Secs. II A–II C, respectively.

A. Supervised learning and feed-forward neural networks

In supervised learning tasks, a model $f_{\boldsymbol{\theta}}$ is tasked with learning a mapping $f_{\boldsymbol{\theta}} : \mathbb{R}^M \mapsto \mathbb{R}^{M'}$, where M is the number of features (or the length of the input vector) and M' is the number of targets (or the length of the target vector). This mapping can take many forms, including polynomials, random forests, nonparametric models such as Gaussian processes, or deep learning architectures, such as neural networks. For the purposes of this discussion, we focus on parametric models, where a finite number of parameters $\boldsymbol{\theta}$ determine the form of the approximating function $f_{\boldsymbol{\theta}}$. During the training (or fitting) process, parameters $\boldsymbol{\theta}$ are optimized so as to minimize the loss function, which is a measure of difference between the ground truth target values $\{\mathbf{y}^{(i)}\}$ and the model predictions $f_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) = \hat{\mathbf{y}}^{(i)}$. The data on which the model is fit is referred to as the training set. Models are fine-tuned on the cross-validation set, and the final model evaluation is performed on data the model has yet to see, usually called the testing set. For an in-depth tutorial on ML techniques and proper use, we refer the reader to Refs. [37,38].

We use feed-forward neural networks (FFNN) to perform the supervised learning task in all results to follow. The details of FFNNs are explained in many other works (and especially in the context of spectroscopy prediction and analysis [8,34,39–43]), and are thus not explained here. However, one key property of FFNNs is that they are *universal approximators*, meaning they can, in principle, model any function provided the model has enough trainable parameters and is trained on enough data. This is relevant for constructing ensembles of neural networks, described in Sec. II C. The details of our features and targets are explained in Sec. III D.

B. Uncertainty quantification

Generally, ML models are tasked with modeling inputs to outputs, as described in the previous subsection. However, there are currently significant ongoing efforts to leverage statistical principles to model, or quantify, the uncertainty in these predictions. For example, Gaussian processes (GPs) [12] are nonparametric generalizations of the multivariate normal distribution to the continuum, and are finding widespread use due to their ability to rigorously quantify statistical uncertainty. This uncertainty is derived from assumptions about correlation lengths embedded in a covariance kernel, allowing one to draw samples from the GP consistent with the parameters of the embedded length scales. For the purposes of this work, we henceforth outline ways to quantify uncertainty in *parametric* models such as FFNNs. We reserve the discussion of ensembling specifically to Sec. II C.

Uncertainty-aware models incorporate UQ in order to address two different types of uncertainty: aleatoric and epistemic [11,44]. Aleatoric uncertainty is also called "irreducible," as it is due to natural physical processes (such as randomness in nature) or inherent instrument error. Intrinsic broadening processes in spectroscopy or noise in an image are two examples of aleatoric uncertainty. We highlight that error bars corresponding to uncertainty in a physical measurement are also examples of aleatoric uncertainty. Epistemic uncertainty is due to an insufficient model. It is often large when training data in some region of the input space is not adequately sampled. Unlike aleatoric uncertainty, epistemic uncertainty can be improved by using some combination of a more sophisticated model, incorporating more training data or prior information. While it is nontrivial, recent work has demonstrated that both classes of uncertainties can be predicted using ML techniques [45].

One method for modeling aleatoric uncertainty in particular is the mean-variance estimation (MVE) [46]. A MVE model will attempt not only to predict the target value, but also an estimation for the uncertainty in that prediction as another output. Concretely, given an input feature dataset $X \in \mathbb{R}^{N\times M}$ (where N is the number of training examples) and a target dataset $Y \in \mathbb{R}^{N\times M'}$, a MVE model will attempt to learn a mapping $f : \mathbb{R}^{M} \mapsto \mathbb{R}^{2M'}$. The output space is doubled in size since for every output prediction, an uncertainty estimate is also predicted. If we consider the scalar output case $M' = 1$, we have a single prediction $\hat{y}$ and a single prediction for the estimate of the uncertainty $\hat{\sigma}$. The MVE model is trained to minimize a negative Gaussian log-likelihood (NLL) loss function,

$$L(y, \hat{y}, \hat{\sigma}) = \frac{1}{2}\ln 2\pi + \frac{1}{2}\ln \hat{\sigma}^2 + \frac{(y-\hat{y})^2}{2\hat{\sigma}^2}. \tag{1}$$

The principle is simple: if the model makes an accurate prediction [i.e., $(y-\hat{y})^2$ is small], it can "afford" to also predict a low uncertainty σ. However, if the prediction error is large and cannot be improved during training (perhaps due to a significantly noisy observation) the model will compensate by increasing $\hat{\sigma}$, despite paying the penalty of the second term in Eq. (1). The fact that a MVE model trained using the NLL loss function has the flexibility to make this tradeoff allows it to estimate the uncertainty in its own predictions, offering considerable utility when modeling irreducible noise.

In our work, the simulation of a XANES spectrum from molecular coordinates is deterministic. Hence, we will only be concerned with modeling epistemic uncertainty. There are several uncertainty quantification techniques that can model epistemic uncertainty, such as Bayesian neural networks (BNN) [47], Monte Carlo dropout (MCD) ensembles [48], and neural network ensembles [49,50]. Instead of directly predicting an estimate of uncertainty, BNN's treat all of their parameters as random variables, allowing one to sample from this distribution during inference, leading to a distribution of predictions. While proven to be extremely effective in certain cases [51–53], BNN's are expensive to train, and it has been argued that they are consistent with simpler, ensemble-based approaches [50]. MCD ensembles work under a similar principle, by randomly disabling neurons during inference, allowing for a sampling over many "effective models" and thus allowing ensemblelike predictions to be obtained. Statistical bootstrapping can also be used to generate model diversity, and has been shown to be superior to MCD in similar applications [36]. In this study, we choose to use NNEs for the reasons explained in detail below. Similar to Ref. [36], we also downsample our training set size in a way similar to bootstrapping, and combine this with ensembling to create even more model diversity than either would produce on its own (and also found that MCD is less effective than our combined downsampling and ensembling method).

C. Neural network ensembles

NNEs are sets of individual models (or estimators, in the case of this work these are FFNNs) which are usually trained independently but used together during inference (see Fig. 1). The working principle of NNEs is that of the wisdom of the crowd (or "query-by-committee" [26]): each individual's prediction is less robust than the aggregate opinion. More rigorously, any two models that produce an accurate result will *by definition* produce predictions that are similar. However, due to the vast training weight-space of deep learning models, when any two models both fail, they will usually fail in different ways. The training weight-space refers to the complete set of trainable parameters of a neural network. The values of these weights are randomly initialized between different estimators, and since this space can be incredibly high-dimensional (easily >1 million and often many orders of magnitude larger), there are a vast number of possible sets of weights corresponding to local minima in the landscape of the loss function. The ensemble learning approach turns one of the neural network's greatest weaknesses into a strength. Each estimator (in its own local minimum) will produce roughly the same correct prediction in a well trained neural network and is partly what allows neural networks to be so flexible. However, the differences between these sets is also what leads to different estimator predictions in failure scenarios.

Empirically, the average over the entire ensemble not only produces better predictive accuracy, but also allows for UQ by interpreting the spread in the predictions of the individual estimators. Theoretically, an ensemble-averaged prediction can be thought of as an averaging over the space of "reasonable" possible functions mapping inputs to outputs given a fixed training set. This space is infinite, of course, and any finite sampling of models is not sufficient to rigorously cover this space. However, it is sufficient to provide useful uncertainty measures. Choosing how many estimators to use remains a highly problem- and model-dependent open research question [26]. This is in stark contrast to a GP, which not only provides an analytic form for the mean and spread of the GP averaged over an infinite number of estimators, the spread itself is rigorously the standard deviation of a Gaussian distribution. A NNE may not adequately approximate the space and the spread is not strictly interpretable as a proper standard deviation (though it can be used in a similar way [21]). We thus highlight a critical pitfall: like almost all ML techniques, especially in deep learning, UQ is hyperparameter-dependent. These methods must be rigorously tested in order to ensure they are of the appropriate quality given the problem at hand.

To our knowledge, there is not yet any formal theory for interpreting the distribution of predictions of NNEs.

In this study, we choose NNEs for their balance of relative simplicity, predictive power and overall performance. We also highlight that they operate on essentially the same paradigm that any individual estimator does, which makes them straightforward to train, debug and deploy.

III. METHODOLOGY

In this work, we showcase the utility of the NNE method for UQ on molecular structure-XAS pairings. Molecular structures are taken from the QM9 database [35], which is a subset of the GDB-17 chemical universe [54]. QM9 contains roughly 134k DFT-geometry optimized small molecules, each with at most 9 heavy atoms (C, N, O, F). Molecular spectra are computed using the multiple scattering code FEFF9 [55]. We focus on C, N and O K-edge XANES spectra from individual absorbing sites, and as such we partition our database into $\mathcal{D}_A$ for $A = \{\mathrm{C}, \mathrm{N}, \mathrm{O}\}$. For example, $\mathcal{D}_\mathrm{O}$ is a database containing all oxygen site-XANES pairs. In the following subsections, we explain how our features and targets are constructed (Secs. III A and III B, respectively) and analyzed (Sec. III C). Finally in Secs. III D and III E, we describe the procedure for setting up the training and testing sets used in the remainder of the work, and implementing our NNE approach.

A. Feature construction

XANES is sensitive to the local chemical environment of absorbing atoms. We therefore choose a structural descriptor that is local to each absorbing site: atom-centered symmetry functions (ACSFs) [3,18]. We also considered other descriptors (particularly those from the DSCRIBE library [56]) including the smooth overlap of atomic positions [57] and many-body tensor representations [58], but ultimately decided to use ACSF.[1] The ACSF feature encodings have been very successful in modeling total energy partitioned into local atomic contributions. ACSFs were first proposed by Behler in 2011 in the context of developing neural network potentials [3,18] (NNPs). The development of NNPs is summarized in a recent review by Behler [22], along with the utility and flexibility offered by ACSFs.

ACSF feature vectors are further described at length in quite a few recent works, including Refs. [34,60], and as such will not be repeated here. In brief, the ACSF feature vectors are atom-resolved representations of the local radial and angular atomic environments of a central atom, which in this work is the absorbing site.

[1]We also considered the weighted-ACSF (wACSF) feature encoding [59], which unlike the traditional ACSF, do not scale in size with the number of unique atom types in the considered data. While this type of encoding is particularly useful when there are a significant number of unique atom types (such as when dealing with the vast spaces of materials or materials complexes [36,39]), it is not necessary for our problem, as we only consider five unique atom types (H, C, N, O, and F).

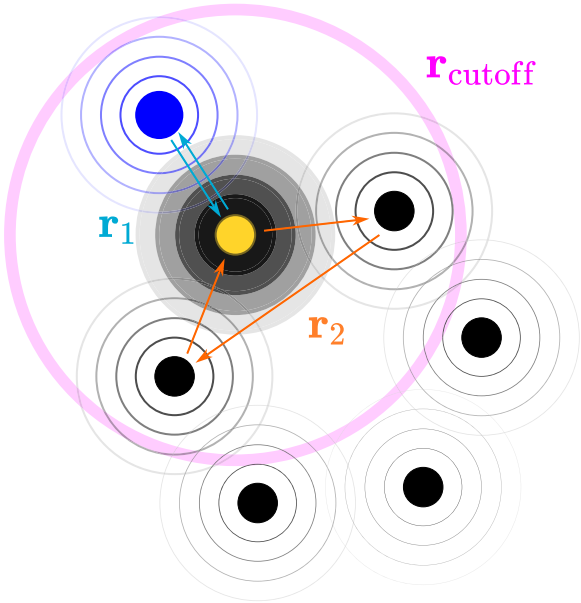

FIG. 2. A cartoon depicting two scattering paths from a central absorbing atom (marked in yellow; other C atoms are black) in an aniline molecule (with hydrogen atoms implicit, and N is blue). The first path is C→ N→ C, and the second C→ C→ C → C. The cutoff region around the central atom marks the geometric information included in the ACSF feature vectors. Information outside of this cutoff (which corresponds to relatively long scattering path lengths) is excluded.

We aim to leverage the locality of XANES in the same way local atomic energy contributions are in constructing NNPs. Physically, the argument that XANES is a local probe can be understood through multiple scattering theory, where the absorption coefficient at a given incident photon energy is determined by the interference between the outgoing wave and the back scattering waves from neighboring atoms. In the multiple scattering path expansion, the absorption coefficient decays exponentially with path length [61,62]. Overall, the longer the scattering path length, the smaller the overall contribution to the XANES spectrum. Typical paths that contribute are illustrated in Fig. 2, where $\mathbf{r}_1$ and $\mathbf{r}_2$ represent two- and three-atom scattering paths, respectively. Of course, a much larger number of paths contribute in principle, but longer paths, mostly those outside of the range $\mathbf{r}_\mathrm{cutoff}$, do not contribute significantly.

In preparing our ACSF features, for every absorbing atom site, we use a radial cutoff of 6 Å, as well as similar parameters to those used in Ref. [60]. H, C, N, O, and F neighbors were considered, and for each absorbing atom site, a feature vector of 155 entries was constructed. The details of our featurization process can be found in Appendix A 1.

We explored feature reduction/importance ranking, similar to that done in Ref. [34], in order to reduce the 155-dimensional input vector to a smaller dimension. We found that training was somewhat stabilized (i.e., loss functions decreased monotonically more consistently), but accuracy overall was not noticeably different from training performed using the full ACSF vector. Thus we choose to use the ACSF feature vectors as-is for inputs to our models.

B. Target construction

As previously mentioned, FEFF9 [55] is used to compute the XANES spectra using multiple scattering theory. Each molecule's FEFF spectrum is computed individually (atom-by-atom), and the details of these calculations are presented in Appendix A 2. In brief, we use a cutoff of 7 Å for self-consistent potential calculations and 9 Å for full multiple scattering calculations, which is commensurate with the geometric cutoff of 6 Å used for the ACSF descriptor construction. The targets are the XANES spectra interpolated using cubic splines onto a common grid, which was chosen to be 200 dimensional, corresponding to a resolution of 0.27 eV.

In brief, the spectral target can be represented as a vector

$$\boldsymbol{\mu}^{(i)} = \left[\mu_1^{(i)}, \mu_2^{(i)}, \ldots, \mu_M^{(i)}\right] \tag{2}$$

for training example i, where $M = 200$ is the number of target values. We consider a spectral range of 50 eV and scale the intensity to unity at the high energy tail, conforming to the standard XANES normalization procedure.

C. Principal component analysis

Prior to performing ML modeling, it is always prudent to explore the data to ensure sensible correlations or patterns exist between features and targets. It also provides the baseline intuition of what to expect from the ML models. Linear dimensionality reduction techniques, such as principal component analysis (PCA), cannot capture the nonlinear relations that a neural network will, but they are still quite useful in identifying overall trends and are largely parameter-independent. More sophisticated nonlinear techniques, such as t-distributed stochastic neighbor embedding (t-SNE) [63], can extract more complicated trends, but are usually highly parameter dependent and thus less robust [64]. We therefore apply PCA on both the ACSF features and spectra targets in order to resolve their relations in a tightly controlled manner.

PCA extracts the "directions of principal variance" in a dataset. The PCA decomposition diagonalizes the $M \times M$ covariance matrix of a dataset $X \in \mathbb{R}^{N\times M}$ (N examples each with M features). The most significant eigenvectors and eigenvalues (the eigenvectors which correspond to directions of maximal variance are indicated by the largest eigenvalues) of the covariance matrix are used to project the data into a lower-dimensional space. Formally, the (scaled) eigenvalues $\mathbf{w}_j$ are the (relative) captured variance ω_j along the direction defined by that eigenvector. For the ith example in the database $\mathbf{X}_i$, and for the jth eigenvector $\mathbf{w}_j$,

$$z_{ij} = \mathbf{X}_i \cdot \mathbf{w}_j, \tag{3}$$

where $\cdot$ is the dot product. For example, z_{i1} captures the first principal component (in the direction of maximal variance) of example i. We also highlight that given a value $\mathbf{z}_i = [z_{i1}, z_{i2}, \ldots, z_{id}]$ with $d < M$, an approximate reconstruction of $\mathbf{X}_i$ can be obtained via

$$\mathbf{X}_i \approx \sum_{j=1}^{d} z_{ij}\mathbf{w}_j. \tag{4}$$

Using the scikit-learn library [65], we apply PCA to the feature (the ACSF vectors) and target (spectra) spaces, independently. Specifically, we perform the dimensionality reduction on the ACSF feature data as $\mathbb{R}^{N\times 155} \to \mathbb{R}^{N\times 2}$ and the spectra target data as $\mathbb{R}^{N\times 200} \to \mathbb{R}^{N\times 2}$. The values of z_{i1} and z_{i2} for the ACSF decomposition are plotted in Fig. 3 on the x and y axes, respectively. The color value of the points represents the value of z_{i1} of the *spectra* decomposition. Note that scales are not shown here, as they not important for the qualitative analysis to follow. Table I tabulates the relative captured variance of each dimension for both the features and targets, in the first two principal directions.

TABLE I. The relative variance captured by the first two principal components of both the ACSF and spectral (μ) spaces.

Absorber	ω_1(ACSF)	ω_2(ACSF)	$\omega_1(\mu)$	$\omega_2(\mu)$
C	0.61	0.16	0.34	0.29
N	0.74	0.09	0.41	0.26
O	0.77	0.11	0.60	0.22

Analysis of Fig. 3 shows clear spatial correlations between the principal values of the ACSF features and spectral targets. Areas of high color density indicate a spectral feature which is strongly correlated to a common structural motif. These regions are indicated by the appropriate labels. For example, while the C atom clustering is the most poorly resolved, cluster (a) appears to correspond to aliphatic carbon chains. For N atoms, (b) clearly corresponds to azides and (c) to primary ketimines. For O atoms, (d), (e), and (f) correspond to esters, alcohols, and ethers, respectively. While the clustering patterns are not definitive on their own, they indicate a high degree of correlation between the XANES spectra and functional group, a result observed in previous work [40]. Not only does this further substantiate the locality of XANES, it also hints that ML techniques will be able to efficiently capture a more complicated nonlinear relationship between XANES spectra and local atomistic geometry.

D. Data splits and preparation

We test two hypotheses using the QM9 dataset, each necessitating different partitionings of the datasets $\mathcal{D}_A$. First, the ACSF feature vectors capture sufficient local structural information about absorbing atoms for accurate prediction of sitewise XANES spectra and uncertainty estimations. Testing our first hypothesis involves evaluating the overall effectiveness of a NNE trained using the usual *random* train/validation/test split. It corresponds to a use case in which the trained NNE is expected to perform on a randomly selected example in the QM9 database. Hence, this first partitioning is referred to as the "random partitioning" ($\mathcal{D}_A^{\mathrm{R}}$). Such a performance measure is perfectly valid if the distribution of molecules in the test set is chemically similar to those found in the QM9 training set.

The second hypothesis is that the XANES spectra are sufficiently local such that an ensemble can be trained on data containing molecules with fewer than n heavy atoms, but still perform on molecules with more than n heavy atoms. Furthermore, the individual local signals can then be averaged, with NNE error propagated, to estimate the molecular XANES, and its error. This is indeed a significant challenge.

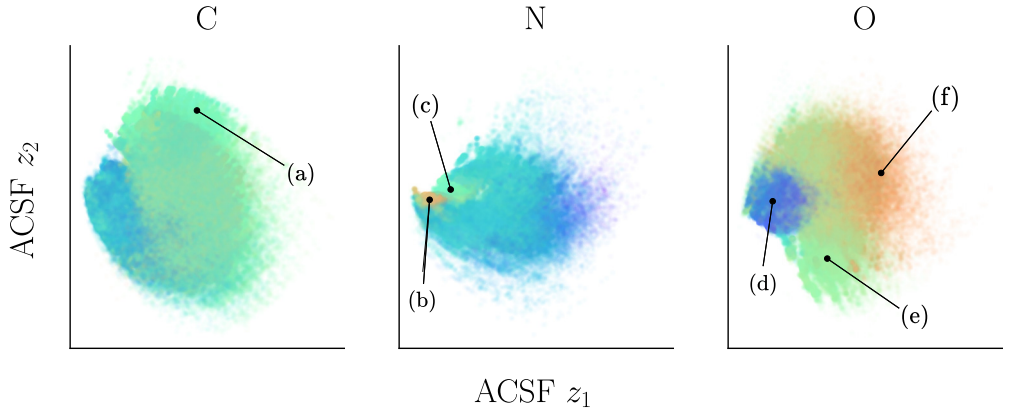

FIG. 3. Principal component analysis of the feature(structure)-target (spectrum) relationship for each of the three datasets of C, N, and O absorbing atoms. z_1 and z_2 are weights along the first and second principal directions ($\mathbf{w}_1$, $\mathbf{w}_2$) as constructed on the ACSF feature data. The color of the markers correspond to the first-principals value of the spectral target data. Regions of significant color density are indicated by labels (a)–(f).

It is known that neural network potentials, which often also use ACSF input vectors with similar parameters, can struggle in systems with long-range correlations. This hypothesis therefore provides a stringent test on both the locality of the XANES spectra and NNE. If nontrivial long-range correlation effects exist in the XANES spectra, our models will suffer due to the intrinsic locality constraint. Similarly, if the chemical environments captured in QM9 are significantly different than those contained in a dataset with larger molecules, the NNE will fail to generalize. As this partitioning will test the ability of the models to generalize, it is henceforth referred to as the "generalization test" ($\mathcal{D}_A^{\mathrm{G}}$).[2]

For each $\mathcal{D}_A^{\mathrm{R}}$, a simple random split is employed, where 90% of data are chosen for training and cross-validation, with the rest held-out as the testing set. For $\mathcal{D}_A^{\mathrm{G}}$, multiple splits are made. The testing sets always contain all of the QM9 molecules with a total of 9 heavy atoms. Training sets are constructed by choosing molecules with anywhere from 5–8 heavy atoms. For example, in one training set instance for nitrogen absorbing atoms, we include sites from all molecules containing at least one N atom, but less than, e.g., total 6 heavy atoms in the training (and cross-validation) splits. Evaluation is then consistently performed on atoms originating from molecules with 9 heavy atoms.

Significantly, the data partition $\mathcal{D}_A^{\mathrm{G}}$ has a strong impact on the size of the training set. Binned by the total number of heavy atoms per molecule, the total number of molecules increases exponentially in the QM9 database.

In Fig. 4, we show the total number of molecules in $\mathcal{D}_A$ as a function of the number of heavy atoms/molecule. As one can see, the total amount of data for 9 heavy atoms/molecule is roughly an order of magnitude greater than for 8. If the generalization test succeeds, an exponential increase in training data could be avoided.

[2]Note that the same data/examples are contained in $\mathcal{D}_A$, $\mathcal{D}_A^{\mathrm{R}}$, and $\mathcal{D}_A^{\mathrm{G}}$. We distinguish between them to highlight that the train/validation/test splits *are* different.

E. Machine learning

We train $|\mathcal{E}| = 30$ independent estimators for all experiments in this work. The details of the training procedures are given in Appendix B, and we highlight the following important points. First, each estimator was always trained on a random 90% sampling (without replacement) of the training set [66,67], meaning some data was purposely excluded during training (the dependence on the sampling proportion is analyzed in Appendix D). Second, each estimator used a randomly initialized neural network architecture. Both of these procedures were employed to maximize model diversity, which as previously discussed, has been shown to be of great utility for UQ.

During initial cross-validation studies, we observed that occasionally individual estimators would produce completely unphysical results. Such results include, but are not limited to, spikes in the spectral intensity an order of magnitude larger than the most intense spectrum in our data and "vanishing" spectra with mostly zero intensity. Likely due to a combination of the random model initialization and training set downsampling, these aberrant results do not contribute meaningful information to either the overall accuracy of the prediction or the uncertainty estimate. Therefore, during

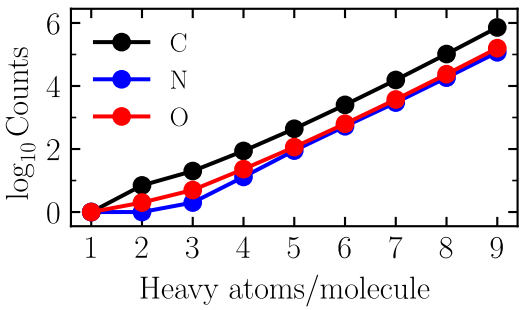

FIG. 4. The total number of molecules contained in QM9 as a function of the number of heavy atoms (C, N, O and F)/molecule for each of the three datasets considered in this work. Each dataset, $\mathcal{D}_A$ (labeled by A for brevity) corresponds to the subset of QM9 in which each molecule contains at least one atom of type A.

inference, these faulty prediction-estimator pairs are discarded when computing ensemble-averaged quantities. We discuss the details of this procedure in Appendix C.

IV. RESULTS

A. Random partitioning

The NNE predictions for spectrum i on spectral grid point j is given by an average over the individual estimators,

$$\hat{\mu}_j^{(i)} = \frac{1}{|\mathcal{E}(i)|} \sum_{k \in \mathcal{E}(i)} \hat{\mu}_j^{(i,k)}, \tag{5}$$

where k is the estimator index, and $\mathcal{E}(i) \subseteq \mathcal{E}$ is the set of estimator indexes corresponding to nonoutlier, physical predictions, for example, i (and $|\mathcal{E}(i)|$ is the size of this set and $\mathcal{E}$ is the set of all estimators). The ensemble-averaged error is

$$\varepsilon^{(i)} = \frac{1}{M} \sum_{j=1}^{M} \varepsilon_j^{(i)}, \tag{6}$$

where

$$\varepsilon_j^{(i)} = \left| \mu_j^{(i)} - \hat{\mu}_j^{(i)} \right|. \tag{7}$$

We note that the vector representation of the predicted XANES spectrum is given in a similar form to Eq. (2),

$$\hat{\boldsymbol{\mu}}^{(i)} = \left[\hat{\mu}_1^{(i)}, \hat{\mu}_2^{(i)}, \ldots, \hat{\mu}_M^{(i)} \right]. \tag{8}$$

During inference, we use the ensemble-averaged quantity as the overall ensemble prediction. In order to demonstrate the NNE's superiority in raw predictive accuracy, we compare the ensemble prediction above to that of the average prediction error of each individual estimator,

$$\varepsilon_{\text{est}}^{(i)} = \frac{1}{|\mathcal{E}|M} \sum_{k=1}^{|\mathcal{E}|} \sum_{j=1}^{M} \left| \mu_j^{(i)} - \hat{\mu}_j^{(i,k)} \right|. \tag{9}$$

Equation (9) can be best thought of as a rough measure of how any *single* model would perform on average. To quantify this, we define the average test error on a logarithmic scale over N_{test} structure-spectrum pairs,

$$\bar{\varepsilon} = \frac{1}{N_{\text{test}}} \sum_i \log_{10} \varepsilon^{(i)} \tag{10a}$$

and

$$\bar{\varepsilon}_{\text{est}} = \frac{1}{N_{\text{test}}} \sum_i \log_{10} \varepsilon_{\text{est}}^{(i)}. \tag{10b}$$

We highlight that $\bar{\varepsilon}$ (-1.45, -1.40, and -1.55) clearly outperforms $\bar{\varepsilon}_{\text{est}}$ (-1.36, -1.32, and -1.46) for the C, N, and O datasets. The details of the distributions of these errors are discussed in Appendix D (Fig. 11).

In order to ground the discussion of overall model performance, we present waterfall plots in Fig. 5 with samples randomly chosen from the worst cases in each decile of the the testing set of $\mathcal{D}_{\text{C}}^{\text{R}}$. One of the worst performers (bottom figure) clearly originates from a rare, challenging (from the electronic structure perspective) structure: a seven-membered fully conjugated ring containing five heteroatoms. Given the chemical space covered in QM9, it is not unsurprising that

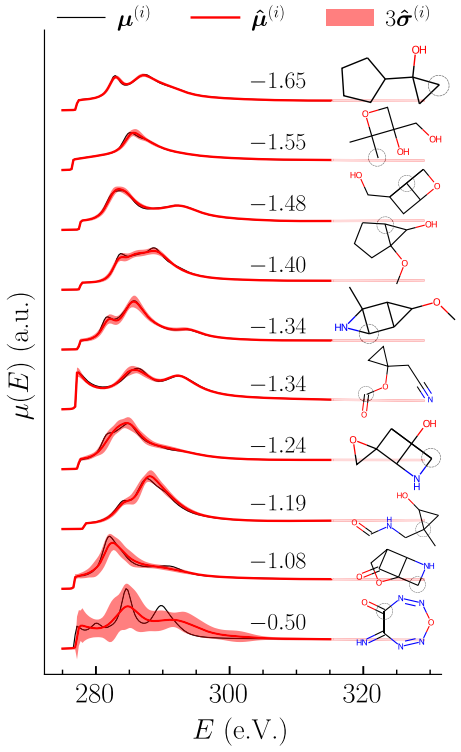

FIG. 5. Waterfall plot sampled from the bottom (worst) of each decile of the testing set results on $\mathcal{D}_{\text{C}}^{\text{R}}$, where decile are sorted from best (top) to worst (bottom). Absorbing carbon atom sites are indicated by a dashed circle in the molecular diagram. The ground truth (black), prediction (red), and 3× the spread (shaded red) are displayed. The value for $\log_{10} \varepsilon^{(i)}$ is also shown. The vector predictions and uncertainties are given by Eqs. (8) and (13).

the prediction is not accurate. However, it appears that the uncertainty estimate yields the qualitatively correct trend, as the prediction appropriately presents with relatively large error bars. On the other hand, all other predictions are relatively accurate, and present with error bars roughly commensurate with the prediction accuracy.

The pointwise NNE spread for spectrum i is defined as

$$\hat{\sigma}_j^{(i)} = \sqrt{\frac{1}{|\mathcal{E}(i)|} \sum_{k \in \mathcal{E}(i)} \left(\hat{\mu}_j^{(i)} - \hat{\mu}_j^{(i,k)} \right)^2}, \tag{11}$$

from which the overall uncertainty of the prediction can be computed,

$$\hat{\sigma}^{(i)} = \frac{1}{M} \sum_{j=1}^{M} \hat{\sigma}_j^{(i)}. \tag{12}$$

Similar to Eqs. (2) and (8), the vector uncertainty for a single spectrum can be represented as

$$\hat{\boldsymbol{\sigma}}^{(i)} = \left[\hat{\sigma}_1^{(i)}, \hat{\sigma}_2^{(i)}, \ldots, \hat{\sigma}_M^{(i)} \right]. \tag{13}$$

It is important to note that Eqs. (11)–(13) are independent of the ground truth predictions. Additionally, we note that unlike, e.g., a GP, the spreads $\sigma_j^{(i)}$ are *not* proper standard deviations, since the distribution of estimator outputs is not guaranteed to

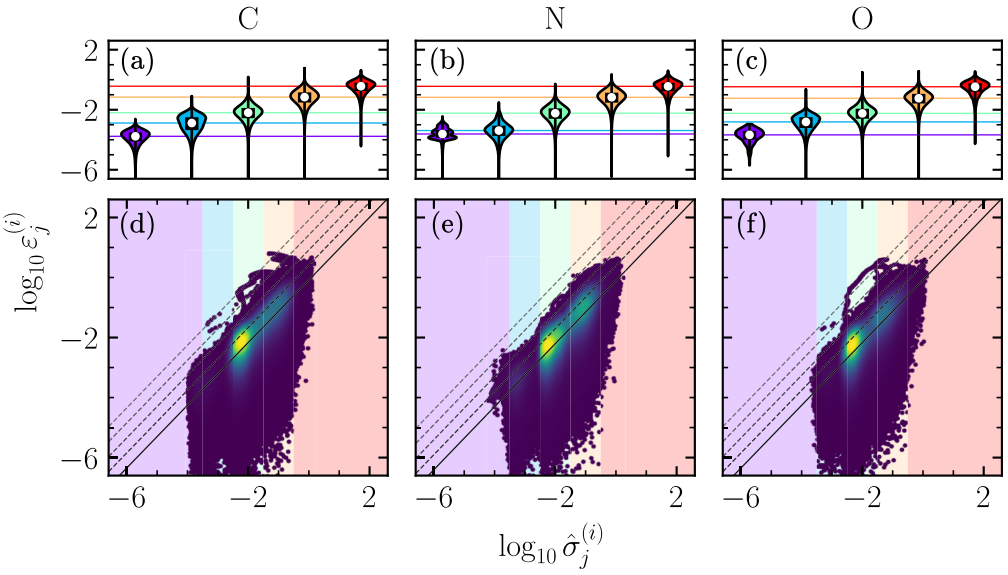

FIG. 6. Violin plots of the $\log_{10}$-ensemble error as a function of the binned $\log_{10}$ spreads of the ensemble prediction [(a)–(c)], and parity plots (2D density histograms) of the pointwise $\log_{10}$-ensemble error as a function of the $\log_{10}$-estimator spread [(d)–(f)]. Violin plot colors are coordinated with the bins in the scatterplots, which are given by $\{(-\infty, -3.5), [-3.5, -2.5], [-2.5, -1.5), [-1.5, -0.5), [-0.5, \infty)\}$, and are colored purple, blue, green, orange and red, respectively; the medians of each of the bins are shown for reference as horizontal solid lines. In the scatterplots, regions of high (low) density are shown in yellow (purple). The best linear fit to the scatterplot data (solid line) is shown in addition to four guidelines at half orders of magnitude intervals above it (dashed lines). The origins of the systemic outliers in (d) and (f), those falling about the third and fourth dashed guidelines, are discussed in Appendix E, and do not have a meaningful effect on the presented analysis.

be Gaussian. They are simply a measure of how different each output is from the others.

We present a quantitative analysis of the NNE's capability of accurately capturing uncertainty measures in Fig. 6 for all three datasets $\mathcal{D}_A^{\mathrm{R}}$. In (a)–(c), violin plots of $\log_{10}\varepsilon_j^{(i)}$ for 5 bins of $\log_{10}\hat{\sigma}_j^{(i)}$ (shown as the background colors of the violin plots) are presented. From left to right, the average uncertainty estimate (spread) increases. As the spread increases, so does the estimate of the error, spanning multiple orders of magnitude on each axis. Critically, the distributions are mostly nonoverlapping, meaning the uncertainty estimate can be used to produce a robust estimate for the actual error of the prediction. A more fine-grained presentation of the same data is presented in (d)–(f). These are error parity plots comparing the NNE spread with that of the actual error. To guide the eye, we show the best linear fit to the log-scaled data (solid), as well as four successive parallel lines offset by a half an order of magnitude (dashed). The majority (>88%) of the points fall below the first of these upper bounds (half an order of magnitude above the best fit line), suggesting that most of the time, the error estimate given by this linear trend is an appropriate representation of the worst case scenario. We also note that even when using 10% of the overall training set, the ability of the NNE to accurately quantify uncertainty is unaffected (see Fig. 13).

It is also noteworthy to analyze why roughly 14% of the data fall half an order of magnitude *below* the best fit line. Even when making accurate predictions, each estimator will still predict slightly different values given the same input. These predictions can be accurate overall, but still produce noticeable values for an uncertainty estimate due to their slightly different predictions. This is a consequence of the way that neural networks train and make predictions. Each estimator finds some local minimum in its vast "weight space," and each produces slightly different estimations even when the estimators and ensemble as a whole is making predictions to suitable accuracy. That said, underconfidence in a prediction is not nearly as problematic as overconfidence, and the amount of underconfidence as a function of $\hat{\sigma}_j^{(i)}$ decreases as uncertainty increases. In summary, our results suggest that the ensembles provide a robust, general upper bound for the error.

B. Generalization test

The ability to generalize to previously unseen data is a key feature of any ML model. Generalization can be understood through the lens of data distributions: while the data used during testing must be unseen, the data used to train some model must, in a distributional sense, "look like" the data it is expected to perform on. A simple way to test whether or not two sets of data are in-sample with respect to each other is to combine them and sample randomly. If a source of truth, e.g., a domain expert, can tell the distribution of origin given some random sample, then it is likely that the deployment case, which hopefully is represented by the testing set, is out-of-sample and will lead to poor performance. This is not a catch-all test (e.g., adversarial examples [68]), but it is a useful thought experiment. For example, the testing sets as constructed randomly in Sec. IV A were on balance, by the above definition, in-sample.

There is also a key difference between generalization and extrapolation which is worth noting. No ML models extrapolate beyond the information-theoretic union of the data and prior information they are trained on [69]. Two examples of

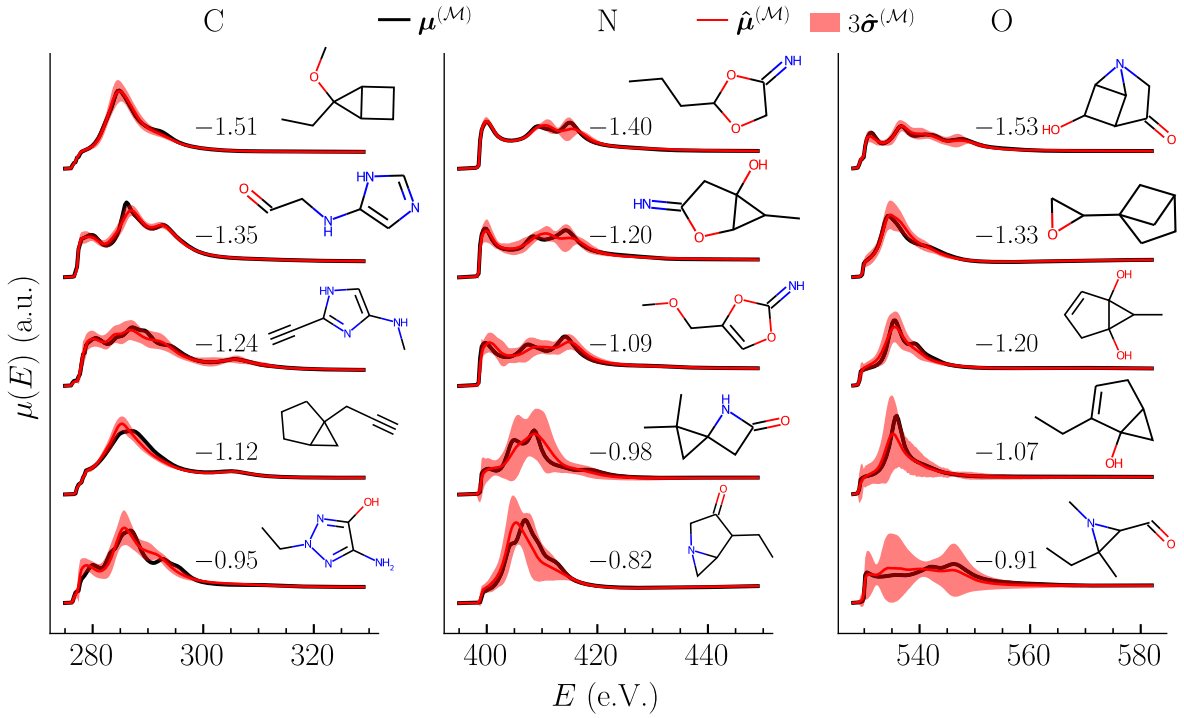

FIG. 7. Waterfall plots similar to that of Fig. 5 showcasing examples from the middle of each pentile on the testing set predictions of $\mathcal{D}_A^{\mathrm{G}}$ for all $A = \{\mathrm{C}, \mathrm{N}, \mathrm{O}\}$. The training sets used in these results consists of absorber-spectrum pairs originating from molecules with at most 6 heavy atoms, while the testing set is the entire subset of QM9 with 9 heavy atoms. The molecules of origin are shown in addition to the $\log_{10}$ error between the predicted molecular spectrum and the ground truth, along with the NNE predicted spread.

"information-theoretic extrapolation" in our work would be (a) predicting on a molecule containing zwitterionic species and (b) predicting on un-relaxed structures. In case (a), the model has not seen any molecules with major charge gradients, and thus it will not understand how to treat those cases. Stated differently, it will neither understand which structural motifs correspond to a zwitterion nor how to treat them once detected. Similarly, in case (b), while it is possible that many structural configurations found in unrelaxed structures are captured in QM9 due to the large diversity of molecules in the dataset, there is no guarantee, since QM9 contains only relaxed geometries.

In this section, we push the boundaries of our NNEs to generalize to new data in a specific way, by training on sites from smaller molecules than what we test on. To do this, we train and cross-validate on subsets of the C, N, and O databases in which there are at most 8 heavy atoms per molecule, and then test on sites originating from molecules containing 9 heavy atoms per molecule (see Sec. III D for details). The specifics of the training and evaluation procedures is identical to those presented in Sec. IV A, except for the particular training/validation/testing split used.

Furthermore, the true test of the ability of the NNEs to generalize is to evaluate performance on *molecular* spectra, defined as the average of the sitewise spectra [see Eq. (14)] (in Sec. IV A, we only present results on site-spectra). For any $\mathcal{D}_A^{\mathrm{G}}$, a molecule is given by $\mathcal{M} \in \mathcal{D}_A^{\mathrm{G}}$, and is defined by a collection of sites. We define the subset of theses sites of atom type A as $\mathcal{M}_A \subset \mathcal{M}$. Given these definitions, the pointwise ground truth molecular XANES spectrum is

$$\mu_j^{(\mathcal{M}_A)} = \frac{1}{|\mathcal{M}_A|} \sum_{i \in \mathcal{M}_A} \mu_j^{(i)}, \tag{14}$$

where $|\mathcal{M}_A|$ is the number of absorbing sites of type A in the molecule. Furthermore, the pointwise molecular XANES spectrum prediction is the average of each of the ensemble predictions for each site,

$$\hat{\mu}_j^{(\mathcal{M}_A)} = \frac{1}{|\mathcal{M}_A|} \sum_{i \in \mathcal{M}_A} \hat{\mu}_j^{(i)}. \tag{15}$$

The estimate of the pointwise spread for the molecular XANES prediction can be calculated using propagation of errors,

$$\hat{\sigma}_j^{(\mathcal{M}_A)} = \frac{1}{|\mathcal{M}_A|} \sqrt{\sum_{i \in \mathcal{M}_A} \left[\hat{\sigma}_j^{(i)}\right]^2}, \tag{16}$$

with an analogous vector representation to that of Eq. (13),

$$\hat{\boldsymbol{\sigma}}^{(\mathcal{M}_A)} = \left[\hat{\sigma}_1^{(\mathcal{M}_A)}, \hat{\sigma}_2^{(\mathcal{M}_A)}, \ldots, \hat{\sigma}_M^{(\mathcal{M}_A)}\right]. \tag{17}$$

Finally, the error of the molecular spectrum is similarly given by a straightforward analog of Eqs. (6) and (7). For brevity, we will often suppress the subscript A where it is clear which atom type/database is being referred to.

We present waterfall plots of the ground truth molecular spectra, and the NNE predictions and spreads in Fig. 7 in the $\mathcal{D}_A^{\mathrm{G}}$ databases. To demonstrate the ability of the NNE to

generalize, we train only on absorbing site-spectrum pairs originating from molecules with at most 6 heavy atoms, but the presented testing set results come from absorbing site-spectrum pairs originating from molecules with 9 heavy atoms. This experiment demands an extreme degree of generalization from the NNE: the subsets of QM9 with only 6 heavy atoms is extremely small, containing only $\approx 10^3$ total structures, three orders of magnitude less than the testing set in this case (see Fig. 4). The dependence of the testing set error on the maximum number of atoms per molecule used in the training data (along with a similar analysis to that presented in Fig. 6) is explored in Appendix D. The key result is that adding orders of magnitude more data results in only slight improvement in testing set error. For example, using the $\mathcal{D}_{\mathrm{C}}^{\mathrm{G}}$ dataset with up to 5 heavy atoms in the training set includes 437 data points, and produces an error of $\bar{\varepsilon} \approx 0.09$. training with up to 8 heavy atoms uses a training set of 102 253 data points, three orders of magnitude more data, and produces an error of 0.03 on the same testing set. Using roughly 10^3 times as much data produces only a factor of 3 improvement in the testing error. Combined with the results in Fig. 7, this indicates that the NNE is already able to generalize to larger molecules at a relatively low training cost, and is incredibly data-efficient. These results are further rigorously quantified in Appendix D (Fig. 12).

In particular the results for $\mathcal{D}_{\mathrm{C}}^{\mathrm{G}}$ present with impressive accuracy given that each spectrum is an average over many carbon atoms (as many as 9 in one of the presented cases). Qualitatively, peak heights and locations are predicted to reasonable accuracy. In contrast, the results for $\mathcal{D}_{\mathrm{N}}^{\mathrm{G}}$ and $\mathcal{D}_{\mathrm{O}}^{\mathrm{G}}$ showcases where the NNE struggles to make accurate predictions for the number of absorbing atoms per molecule. This is largely due to there being far fewer training examples in these two cases relative to $\mathcal{D}_{\mathrm{C}}^{\mathrm{G}}$. Even so, while some peaks are occasionally missed, the spectral trends are still reproduced, and uncertainties are captured.

C. Out-of-equilibrium geometry analysis

To further study how the NNE responds to out-of-sample data, we used the 10 molecules corresponding to sites whose spectra were presented in Fig. 5, and randomly distorted the geometries of the molecules to see how the NNE performs. Our procedure is as follows. First, given a distortion parameter δ, for each coordinate direction and atom in some molecule $\mathcal{M}$, a direction on the unit sphere is chosen at random, scaled by δ, and then used to perturb that atom's coordinate. For each of the 10 molecules and each value of δ, we sample 50 distorted molecules. Second, FEFF calculations are then run on each of these new geometries. Finally, the trained NNE used in the $\mathcal{D}_{\mathrm{C}}^{\mathrm{R}}$ experiments is then used to predict the XANES spectrum and spread for each of the site-molecule pairs.

We use $\delta \in \{0.01, 0.02, \ldots, 0.1\}$ Å, and present averaged results analogous to Eq. (10) in Fig. 8 as a function of δ. Most importantly, as the average error increases across the average of all results for a given δ, so does the uncertainty measure. This trend is significant and covers roughly a half of an order of magnitude. This shows quantitatively that on average, the uncertainty estimate tracks how out-of-sample a dataset is with respect to its training data. However, it does not quantify the relative difference between a certain and uncertain prediction. To address this, we sample a single example for each molecule-δ pair, and plot the ground truth, NNE prediction and spreads in Fig. 9. Appendix F contains more fine-grained details relating to this analysis, including density parity plots of the errors and uncertainty measures (Fig. 15), and a waterfall plot of distorted spectra (Fig. 16).

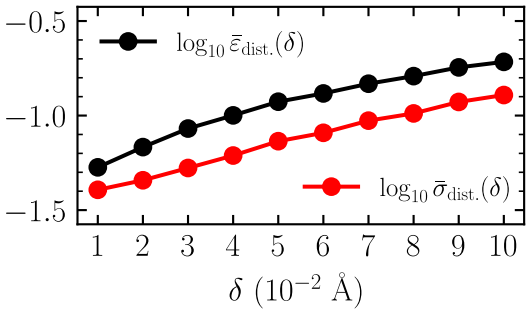

FIG. 8. The average $\log_{10}$ errors and uncertainties (across the 10 example molecules listed in Fig. 5 and 50 random distortions per molecule per value of delta) plotted as a function of δ.

While it is only a small sample of the dataset of distorted molecules, the results in Fig. 9 clearly show that on average, the NNE is able to detect when distorted geometries are sufficiently out-of-sample to render a prediction inaccurate. Overall accuracy varies visually as the distortion increases, but the uncertainty estimate gets significantly larger after $\delta = 0.03$. In most cases, this uncertainty stays relatively large compared to e.g. $\delta = 0.01$, indicating the NNE recognizes that the geometry is likely unseen. We highlight that detecting when a geometry is out-of-sample is not a trivial task, since while the ACSF vectors are human-interpretable, they are not *easily* so. Defining necessary heuristics to detect an out-of-sample geometry is likely not feasible. The fact that the NNE can perform this task verifies its potential usefulness in situations where detecting, e.g., change points [70] (where the statistical distribution of data changes) is required. This could be of particular use in active learning loops, where new training data are sampled based on the uncertainty measure of the ML model.

V. CONCLUSIONS AND OUTLOOK

In this work, we use NNEs to make quantitatively accurate predictions of molecular XANES spectra from local atomistic geometry, and to accurately quantify the uncertainty of those predictions under a variety of conditions. Simulating XANES spectra of a large number of molecules and clusters at a high level of theory is computationally demanding, and UA surrogate modeling provides an avenue for greatly accelerating the simulations while reliably quantifying model confidence. Often, for comparison with experimental measurements it requires an expensive averaging over a large number of structures, such as when computing a thermal average, which necessitates the use of surrogate model acceleration. We anticipate that the NNE approach, and UA modeling in general, can be particularly useful for large, complex systems, such as the dynamical evolution of protein structures, organic liquids,

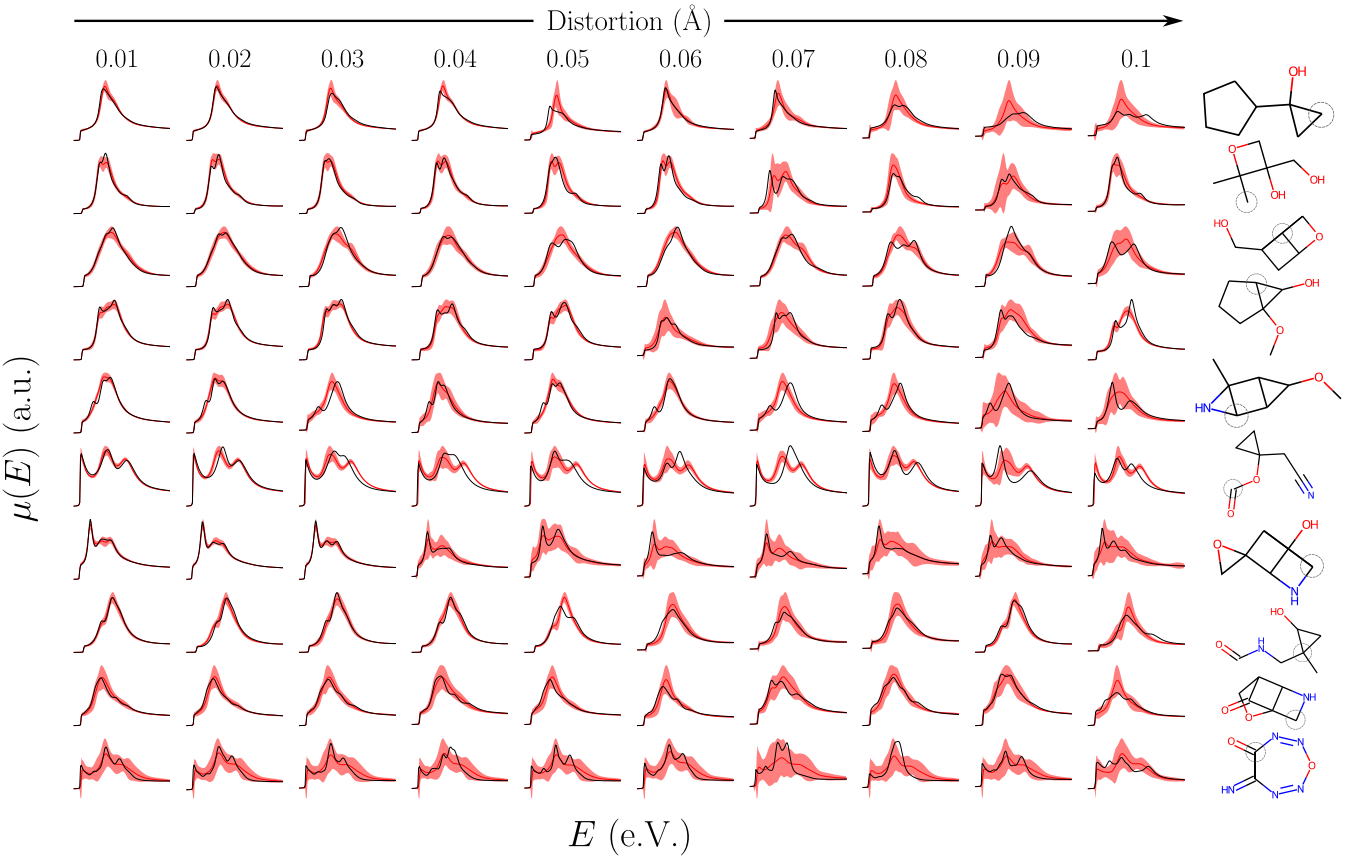

FIG. 9. Predicted XANES spectra (red) and uncertainty estimations (red shaded regions) compared with the ground truth (black) for single examples for each site-δ pairing. Predictions on the same molecule-site pairs as Fig. 5 were computed in inference mode using models trained on the $\mathcal{D}_{\mathrm{C}}^{\mathrm{R}}$ dataset.

and solvated molecules, allowing users to make predictions efficiently and with confidence.

Although our work falls into the space of ML-driven spectral function prediction, an accurate surrogate also has important implications for the inverse problem, where physical descriptors are extracted from the spectral function. For example, one strategy to solve the inverse problem in XAS is to identify candidate structures that produce results *consistent* with a target. This is more broadly known as structure refinement. Various sampling methods are widely used for this purpose, such as reverse Monte Carlo [71,72] and genetic algorithms [73]. Combining these sampling methods with an accurate and efficient forward surrogate model opens new avenues to tackle the inverse problem.

Beyond the problems presented in this manuscript, we believe that the general principles of UQ methods could have broad implications not only for the case of datasets from *in silico* experiments, but also for laboratory measurements in experimental science. While modeling aleatoric noise in experimental data is required for statistically robust predictions, UQ techniques can also be applied to, e.g., quality assurance and control. For example, the detection and elimination of data that results from a variety of experimental sources such as misalignment and errors in control settings (similar to Appendices C and E). UQ-enabled models could also be useful for predicting experiment-quality data. As long as noise and other sources of uncertainty can be *accurately* modeled, it would mitigate the risk otherwise posed by the immense challenge of modeling experiments with these types of data-driven techniques.

As problems become more complicated, and calculations become more expensive (and thus the stakes become higher), ensuring model confidence becomes evermore important. Conveniently, there exists a vast array of UA models and methodologies for quantifying uncertainty, each with their own strengths and weaknesses. In the case of NNEs, the cost of training multiple models is more than worth the payoff. The application of UQ techniques to vector targets, and to a wider variety of problems in the physical sciences is certainly still an open problem in general. However, we have found that one can gain a significant amount of utility through the straightforward use of a NNE: an ensemble of independent estimators. In this case, if you can train one model, you can train a sufficient number of models to produce a reasonable measure of uncertainty with a low overhead.

In conclusion, our case study demonstrates the robust performance of a NNE with uncertainty quantification for predicting complex targets (XANES spectra of small molecules) from the descriptors of local chemical and structural information. More generally, this expands the scope of uncertainty-aware machine learning methods to the case of predicting vector quantities in physical modeling, an area that

is largely unexplored to date. UQ modeling offers compelling advantages over traditional more boilerplate machine learning techniques at an acceptable cost.

All software used in this work, as well as all data used in this work, including FEFF spectra input/output files, featured data, and the neural network ensembles, can be found in Ref. [74].

ACKNOWLEDGMENTS

M.R.C. would like to thank Nongnuch Artrith and Alexander Urban for helpful discussions regarding active learning. This research is based upon work supported by the U.S. Department of Energy, Office of Science, Office Basic Energy Sciences, under Award No. FWP PS-030. This research also used theory and computational resources of the Center for Functional Nanomaterials, which is a U.S. Department of Energy Office of Science User Facility, and the Scientific Data and Computing Center, a component of the Computational Science Initiative, at Brookhaven National Laboratory under Contract No. DE-SC0012704. This project was supported in part by the U.S. Department of Energy, Office of Science, Office of Workforce Development for Teachers and Scientists (WDTS) under the Science Undergraduate Laboratory Internships Program (SULI).

APPENDIX A: DATABASE CONSTRUCTION

1. ACSF feature vector details

Molecular geometry files were read directly from the QM9 [35] database, which is freely available for download at Ref. [75]. We utilize the PYMATGEN [76], DSCRIBE [56], and ASE [77] libraries to construct our ACSF feature vectors. Zwitterionic molecules, those which contain an equal number of positively and negatively charged motifs, are discarded and not included in the machine learning databases. The initialization of the ACSF object is shown below.

```
from dscribe.descriptors import ACSF # v 1.2.1
neighbors = [''H'', ''C'', ''O'', ''N'', ''F'']
rcut = 6.0 # Angstroms
g2_params = [
[1.0, 0],
[0.1, 0],
[0.01, 0]
]
g4_params=[
[0.001, 1.0, -1.0],
[0.001, 2.0, -1.0],
[0.001, 4.0, -1.0],
[0.01, 1.0, -1.0],
[0.01, 2.0, -1.0],
[0.01, 4.0, -1.0],
[0.1, 1.0, -1.0],
[0.1, 2.0, -1.0],
[0.1, 3.0, -1.0]
]
acsf = ACSF(
species=neighbors,
rcut=rcut,
g2_params=g2_params,
g4_params=g4_params
)
```

We iterate through every possible atom in each molecule, and construct the atom's ACSF vector if it matches an absorbing atom used in this work (C, N, or O).

2. Spectra target vector details

Sitewise spectra for each of C, N, and O absorbing atoms were computed using the FEFF9 code [55]. A common preamble to a FEFF calculation is shown below. Particularly, we use a corehole approximation at the random phase approximation (RPA) level of theory, full multiple scattering up to 9 Å, and self-consistency up to 7 Å.

```
TITLE...

EDGE K
S02 1.0
COREHOLE RPA
CONTROL 1 1 1 1 1 1

XANES 4 0.04 0.1

FMS 9.0
EXCHANGE 0 0.0 0.0 2
SCF 7.0 1 100 0.2 3
RPATH -1
```

Once the initial spectral databases were constructed, we screen for extreme outliers or unphysical spectra using methods similar to those described in Ref. [39] (though we note that these screening procedures are not entirely robust, see the discussion corresponding to Fig. 14). Spectra are then interpolated onto common grids using cubic splines. The grids for each absorbing atom type are shown below using Python+NumPy.

```
import numpy as np
M = 200
grids =
''C'': np.linspace(275, 329, M),
''N'': np.linspace(395, 449, M),
''O'': np.linspace(528, 582, M)

```

APPENDIX B: TRAINING DETAILS

In this Appendix, we highlight the important details of our training procedures which can be used to reproduce the work presented in this manuscript. All training was performed on Tesla V100 GPUs using Pytorch+PyTorch Lightning, and the summary of the training and software used for our machine learning pipeline are shown in Table II.

Each estimator, a FFNN, was trained independently. Each estimator in the ensemble was randomly initialized, with a minimum of 4 layers, a maximum of 20 layers, a minimum of 160 neurons/layer, and a maximum of 200 neurons/layers. The Adam optimizer, L1 loss function, Leaky ReLU activations (except the last layer, which is a softplus) and batch

TABLE II. Machine software and hardware details.

GPU	Tesla V100-SXM2-32GB
CUDA	11.4
PyTorch [78]	1.11.0 + cu113
PyTorch Lightning [79]	1.6.4

normalization (except the last layer) were used for every instance of training.

During training, learning rates were multiplied by a factor of 0.95 after 20 successive epochs in which the validation loss failed to decrease. A maximum of 2000 epochs proved sufficient with early stopping criteria monitoring when the validation loss plateaued during training (with a patience of 100 epochs).

APPENDIX C: ENSEMBLE PREDICTION DETAILS

In Sec. IV A, we defined the set of estimator indexes to be $\mathcal{E}$, and the set of estimator indexes corresponding to "reasonable" predictions to be $\mathcal{E}(i)$. We now define precisely how we determine whether or not a prediction is reasonable.

We utilize three criteria to screen for unreasonable predictions.

(1) For a set of predicted spectra (with fixed i) $\{\hat{\boldsymbol{\mu}}^{(i,k)}\}_{k=1}^{|\mathcal{E}|}$, we define the estimator spread $\hat{\boldsymbol{\sigma}}^{(i)}$. Any spectra in which 70% of the total grid points fall outside a region defined by $\hat{\boldsymbol{\mu}}^{(i)} \pm 2\hat{\boldsymbol{\sigma}}^{(i)}$ are discarded.

(2) Given a predicted spectrum $\hat{\boldsymbol{\mu}}^{(i,k)}$, if any point $\hat{\mu}_j^{(i,k)}$ is greater than 20 (a.u.; roughly an order of magnitude greater than the largest spectral intensity in our datasets) that prediction is discarded.

(3) If more than half of the predictions $\hat{\mu}_j^{(i,k)}$ for a given i, k fall below an intensity of 0.05, that prediction is discarded.

These three rules are purely empirical and capture three different kinds of model failure scenarios, each of which is completely independent of the ground truth data. In point 1, we screen for statistical outliers. The spread of the estimator predictions is treated as a Gaussian standard deviation (we note again that this treatment is purely empirical), and any spectra in which a substantial portion of the predicted values fall outside of an ≈95% confidence interval are discarded. Points 2 and 3 screen on physical grounds. We know from experience and quantum mechanical principles that the FEFF code will only output predictions of certain intensities. Model failure situations routinely included single estimators predicting intensities of $\mu > 100$, which are clearly unphysical. Similarly, it is known that the XAS is mostly positive. While model outputs are hard-constrained to be non-negative by the Softplus activation functions, they can be approximately zero. If a sufficient portion of the spectrum is close to zero, we know that prediction is unphysical and hence it is discarded.

The set of estimator indexes that correspond to reasonable predictions is thus defined as $\mathcal{E}(i)$, and is inherently dependent on the training example. It is worth highlighting that if estimator k is discarded for training example i, it likely will not be discarded, for example, $i' \neq i$.

APPENDIX D: TRAINING SET SIZE DEPENDENCE OF $\mathcal{D}_A^{\mathrm{R}}$ AND $\mathcal{D}_A^{\mathrm{G}}$

A useful sanity check when performing any ML modeling is to ensure that error decreases as the training set size increases. For any single estimator, this is usually tested by downsampling the training set by some proportion p and evaluating on a fixed testing set. The value for p is then scanned and the testing set error as a function of p evaluated. We evaluate this metric for both $\mathcal{D}_A^{\mathrm{R}}$ and $\mathcal{D}_A^{\mathrm{G}}$, by randomly downsampling and showcasing how testing set error changes as a function of the number of atoms per molecule used during training.

Instead of training a single estimator, we evaluate these metrics on the ensembles. For the random partitioning datasets $\mathcal{D}_A^{\mathrm{R}}$, we randomly downsample the fixed training set by proportion p independently for each of the 30 estimators, train them on the pN_{train} training examples, and evaluate on the fixed testing set (the same testing set used in the random partitioning dataset, see Sec. IV A). Each ensemble was randomly initialized for each p. Ensemble predictions are made in accordance with the protocol described in Appendix C, and the results averaged over all N_{test} testing examples. Similarly, for the generalization test databases $\mathcal{D}_A^{\mathrm{G}}$, we follow the procedure as outlined in Sec. IV B, and for each instance of the number of atoms per molecule, compute the amount of training data used. All of these results are plotted in Fig. 10.

Immediately, it is clear that the testing set error *trend* is decreasing with increasing p and with increasing amounts of training data, exactly as expected. However in Fig. 10(a), the decrease is not monotonic for the N and O datasets, though it is for the C dataset. Given that the C dataset is much larger (see Fig. 4), it is more probable that even small samples of the training dataset capture general trends, even for small p. The N and O datasets are significantly smaller, and hence for small p it is less likely that significant trends are captured on balance. Finally, the relative decrease in the testing error as a function of p is surprisingly small. For example, the percent change in the N results from $p = 0.1$ to 0.9 is only roughly 12%. While this is a relatively small change (see, e.g., Supplementary Material in Ref. [40]), it is actually an encouraging result. It is likely that failures of individual estimators in certain regions of the input parameter space at small p are compensated for by other estimators in the ensemble, making the overall inference procedure at low p surprisingly robust.

In Fig. 10(b), the trend is much more clear because the amount of training data spans multiple orders of magnitude. However, a key takeaway is that the decrease in the error is incredibly small relative to the amount of training data used: roughly a single order of magnitude compared to 3 orders of magnitude increase in the amount of training data. As discussed in the main text, this is a testament to the NNE method's ability to generalize when trained on absorbing site data, and the diversity of the chemical space of QM9's molecules containing even $\sim 10^3$ heavy atoms per molecule.

We present the error histograms for most of the relevant training procedures in this work in Fig. 11. Subfigures (a)–(c) depict the $\log_{10}$ sitewise testing set error distributions for $\mathcal{D}_A^{\mathrm{R}}$, both for the ensemble-averaged error (black) and the average single estimator error (red). One can see clearly that the

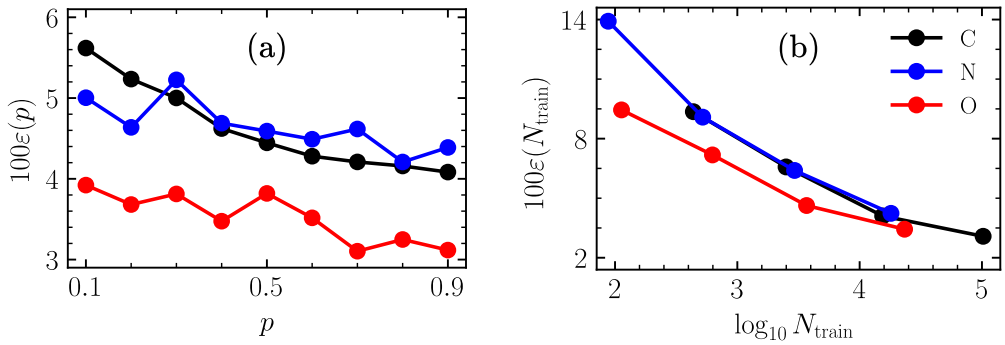

FIG. 10. (a) The testing set error on $\mathcal{D}_A^{\text{R}}$ as a function of the downsampling proportion p, which determines the random proportion of the training set actually used during training. (b) The testing set error on $\mathcal{D}_A^{\text{G}}$ as a function of the training set size, $|\mathcal{M}|$ (which is tied to the number of atoms per molecule in the training set).

ensemble predictions are systematically better than any single estimator on average, as expected of ensemble-based models. Subfigures (d)–(f) showcase the $\log_{10}$ molecular testing set error distributions for $\mathcal{D}_A^{\text{G}}$, as a function of the total number of atoms per molecule used during training (see legend). Figure 12 is the analog of Fig. 6 for the molecular data. It shows the same trends and overall behavior as the aforementioned site-spectrum figure in the main text.

Finally, we present a qualitative analysis of how the error and standard deviations correlate as a function of training set size pN_{train} in Fig. 13. Even for $p = 0.1$, we see that the correlation between the average errors and standard deviations remains intact. This is an indication that even with limited training data, uncertainty-quantifying models can still accurately gauge when they are out of sample, and provide a reasonable estimate as to their own uncertainty.

APPENDIX E: EXPLANATION OF FIG. 6 OUTLIERS

The C and O results in Fig. 6 present with some interesting outlier behavior (for example, the data above the fourth dashed lines). These patterns clearly indicate something has gone systematically wrong, and as such, as a pedagogical exercise we investigate the cause of such significant underestimation of the errors. As an example, we look at the O results; it turns out that most of the outliers (captured by taking $\log_{10} \hat{\sigma}_j^{(i)} < -2$ and $\log_{10} \hat{\varepsilon}_j^{(i)} > -1$ in Fig. 6) come from a single spectrum. This failure case is plotted in Fig. 14.

The significant underestimation of the error is visually obvious by comparing the spread of the predictions (red) with the supposed ground truth (black). However, this ground truth spectrum is actually an outlier with respect to other training data, suggesting the possibility that the ground truth is incorrect. To test this hypothesis, we also plot the closest

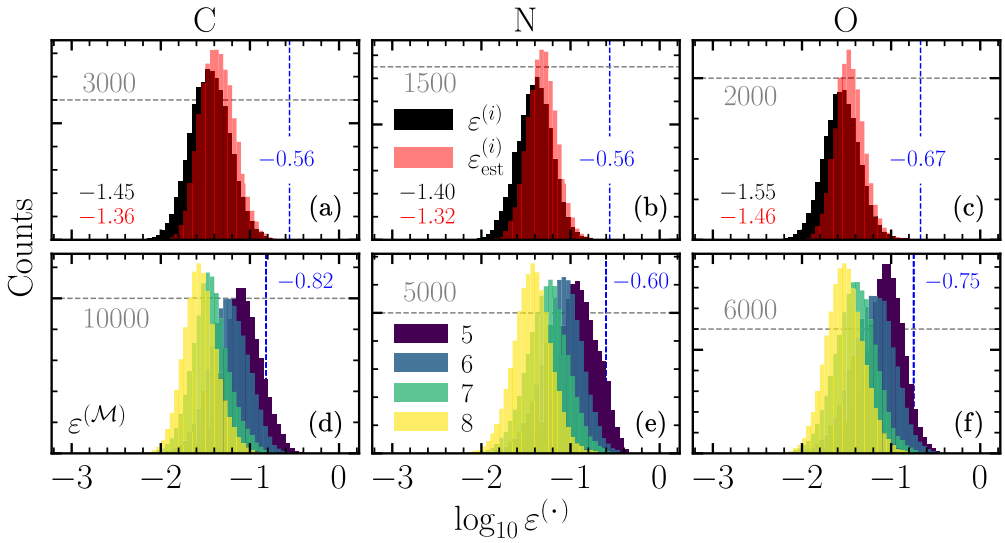

FIG. 11. Histograms of the $\log_{10}$ testing set error of the molecular predictions between the predicted and ground truth spectra for the $\mathcal{D}_A^{\text{R}}$ and $\mathcal{D}_A^{\text{G}}$, (a)–(c) and (d)–(f), respectively. Results in (a)–(c) showcase the distribution of ensemble errors (black) and distribution of the average estimator errors (red), which correspond to Eqs. (6) and (9), respectively. The median values for these results are also shown. Results in (d)–(f) are resolved by the number of heavy atoms/molecule $|\mathcal{M}| \in \{5, 6, 7, 8\}$ in the training set. All plots show the respective "dummy model" prediction (average of the testing set spectra) in dashed blue.

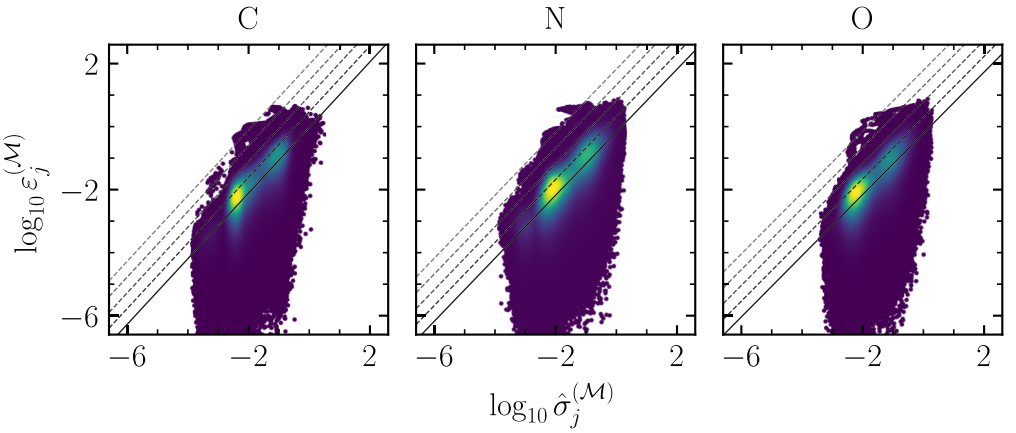

FIG. 12. Parity density plot for the $\mathcal{D}_A^{\mathrm{G}}$ datasets in which molecules with at most 6 heavy atoms were used for training.

ground truth spectrum to the mean of the predicted spectra from the *training* set (dashed black, $\boldsymbol{\mu}^\star$). These two spectra are in almost perfect agreement. We then compare the molecular structures of the inputs, which are also shown in Fig. 14. The SMILES string corresponding to the ground truth and best training set example, `CCCC1(C)COC=N1` and `CCCC1COC=N1`, respectively, are chemically almost identical, differing by a single methyl group. Additionally, the top 3 closest spectra to the ensemble-averaged prediction all contain a `N=C-O` motif contained in a five-membered ring, strongly suggesting that the FEFF calculation used to generate the ground truth spectrum failed to properly converge, and indicating that the NNE prediction, and low uncertainty estimate, are actually correct.

In order to confirm this hypothesis, we performed two sanity checks. First, we reran the FEFF calculations to see if the convergence failure was systemic. Second, we double checked that a different computational spectroscopy software (we chose to use the Vienna *ab initio* simulation package, or VASP [80]), produced similar spectra for these absorbing sites as well. In summary, our original FEFF calculation failed to converge, the rerun produced the expected result (similar to $\boldsymbol{\mu}^\star$ in Fig. 14), and the two VASP calculations (VASP calculations were performed using PBE pseudopotentials a $2 \times 2 \times 2$ grid for the k-points, and a square supercell volume of $20 \times 20 \times 20$; all quantities are in units of angstroms) also produced qualitatively similar spectra, confirming that indeed the O-XANES spectrum of each of these molecules is essentially the same, and that the NNE prediction is correct.

Of course, for a real world deployment scenario, any piece of ground truth data that ends up being unphysical would be removed from the training datasets. However, in this case, we highlight that the NNE model was robust to these outliers. Future work could be dedicated to exploring how robust NNEs or related methods are for outlier detection in a database in general, especially in cases where it is suspected the source of truth can actually be incorrect.

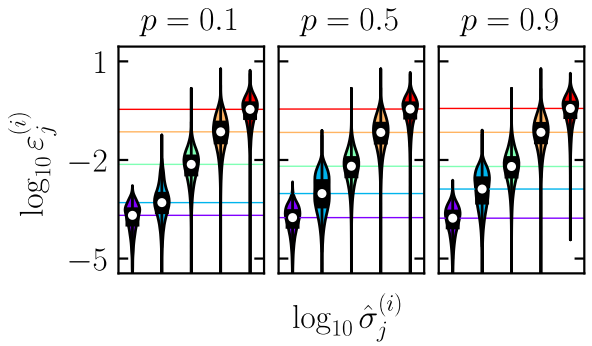

FIG. 13. Violin plots similar to that of Fig. 6 in the main text (with the same bins), but resolved by the proportion of the training set size, p. Results are presented for $\mathcal{D}_{\mathrm{C}}^{\mathrm{R}}$.

APPENDIX F: SUPPLEMENTARY ANALYSIS OF OUT-OF-EQUILIBRIUM DISTORTION TESTS

In this Appendix, we present two useful figures for the analysis of the out-of-equilibrium geometry tests. First, in Fig. 15, we present a parity plot of the pointwise $\log_{10}$ errors vs. the $\log_{10}$ uncertainty estimates. The same positive linear trend found in Figs. 6 and 12 is realized here. Additionally, it is subtle, but as the distortion amount increases, so does the overall error *and* uncertainty estimate, continuing to substantiate the results in Sec. IV C.

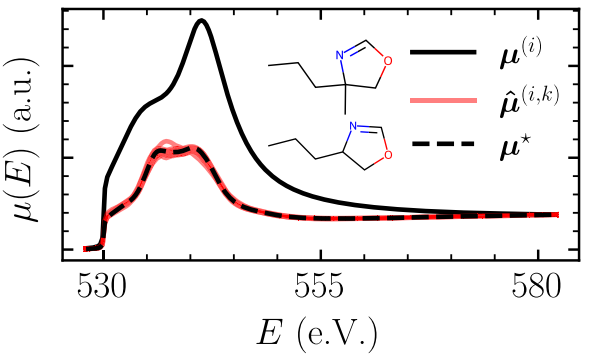

FIG. 14. An example from $\mathcal{D}_0^{\mathrm{R}}$ of the NNE making an accurate prediction even when the source of ground truth used to train the ensemble was incorrect. The supposed ground truth (solid black) and the predictions of each estimator (red) are compared to the closest spectrum to $\hat{\boldsymbol{\mu}}^{(i)}$ in the training database (dashed black, $\boldsymbol{\mu}^\star$). The $\log_{10}$ error between $\boldsymbol{\mu}^\star$ and the prediction $\hat{\boldsymbol{\mu}}^{(i)}$ is -2.16, corroborating the relatively low $\log_{10}$ uncertainty estimate -1.79.

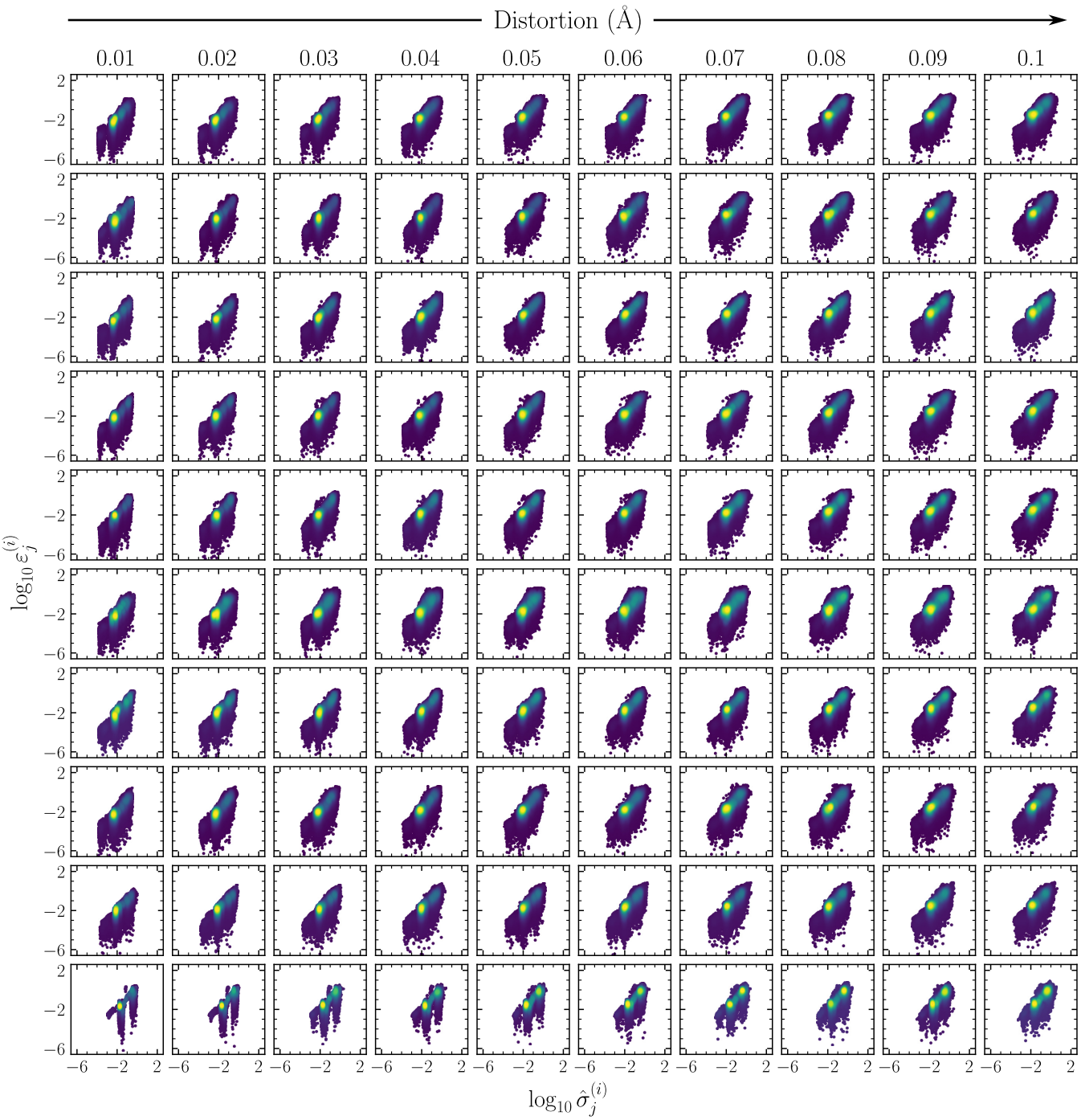

FIG. 15. Parity density plots similar to those in Figs. 6 and 12. Each column is indexed by a distortion value δ, which is given in units of angstroms and represents the maximum value of uniformally sampled random noise used to distort the locations of each atom along the x, y, and z axes. Approximately 50 random samples for each value of δ were used (less than 1% of the time a FEFF calculation would fail). Each row corresponds to a different site from the $\mathcal{D}_{\mathrm{C}}^{\mathrm{R}}$ dataset, specifically those of Fig. 5.

Second, to provide an idea of what distorted spectra look like, in Fig. 16, we showcase a waterfall plot of random sampling of the database of distorted spectra, resolved once again by values of δ and molecule (in the same order as the previous related figures). As a result of the geometry distortion, every molecule begins to exhibit significant new spectral trends. Given that these new geometries are necessarily not in their equilibrium state, they will be much more challenging (if not infeasible) for the NNE to predict, and provide a good test for the NNE's ability to quantify epistemic uncertainty.

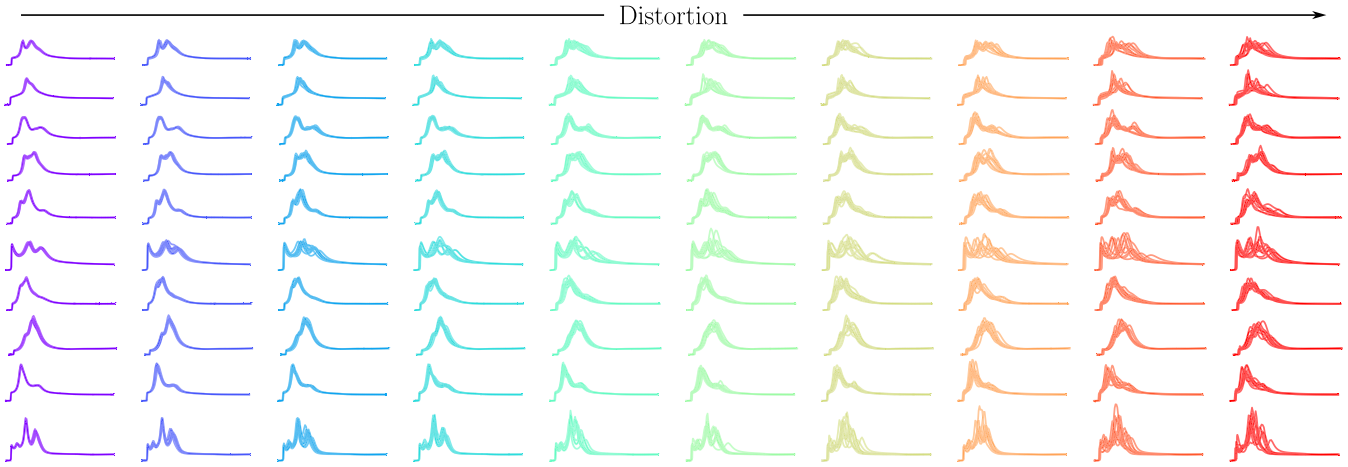

FIG. 16. Waterfall plot of the XANES spectra corresponding to the distorted geometries as outlined in Sec. IV C. Each row corresponds to a specific site on a different molecule (the molecules are in the same order as presented in Fig. 5), and each column corresponds to a different degree of distortion, $\delta \in \{0.01, 0.02, \ldots, 0.1\}$ Å.

[1] K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev, and A. Walsh, Machine learning for molecular and materials science, Nature (London) **559**, 547 (2018).

[2] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, Machine learning and the physical sciences, Rev. Mod. Phys. **91**, 045002 (2019).

[3] J. Behler and M. Parrinello, Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces, Phys. Rev. Lett. **98**, 146401 (2007).

[4] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. Von Lilienfeld, Big data meets quantum chemistry approximations: the δ-machine learning approach, J. Chem. Theory Comput. **11**, 2087 (2015).

[5] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, and V. Pande, Moleculenet: A benchmark for molecular machine learning, Chem. Sci. **9**, 513 (2018).

[6] R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik, Automatic chemical design using a data-driven continuous representation of molecules, ACS Cent. Sci. **4**, 268 (2018).

[7] Y. Nomura, A. S. Darmawan, Y. Yamaji, and M. Imada, Restricted boltzmann machine learning for solving strongly correlated quantum systems, Phys. Rev. B **96**, 205152 (2017).

[8] L. F. Arsenault, A. Lopez-Bezanilla, O. A. von Lilienfeld, and A. J. Millis, Machine learning for many-body physics: The case of the anderson impurity model, Phys. Rev. B **90**, 155136 (2014).

[9] D.-L. Deng, X. Li, and S. Das Sarma, Machine learning topological states, Phys. Rev. B **96**, 195145 (2017).

[10] K. Ch'ng, J. Carrasquilla, R. G. Melko, and E. Khatami, Machine Learning Phases of Strongly Correlated Fermions, Phys. Rev. X **7**, 031038 (2017).

[11] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya *et al.*, A review of uncertainty quantification in deep learning: Techniques, applications and challenges, Inf. Fusion **76**, 243 (2021).

[12] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning* (MIT Press, Cambridge, 2006).

[13] S. Krishnadasan, R. J. C. Brown, A. J. Demello, and J. C. Demello, Intelligent routes to the controlled synthesis of nanoparticles, Lab Chip **7**, 1434 (2007).

[14] D. E. Fitzpatrick, C. Battilocchio, and S. V. Ley, A novel internet-based reaction monitoring, control and autonomous self-optimization platform for chemical synthesis, Org. Process Res. Dev. **20**, 386 (2016).

[15] R. W. Epps, M. S. Bowen, A. A. Volk, K. Abdel-Latif, S. Han, K. G. Reyes, A. Amassian, and M. Abolhasani, Artificial chemist: an autonomous quantum dot synthesis bot, Adv. Mater. **32**, 2001626 (2020).

[16] A. E. Gongora, B. Xu, W. Perry, C. Okoye, P. Riley, K. G. Reyes, E. F. Morgan, and K. A. Brown, A bayesian experimental autonomous researcher for mechanical design, Sci. Adv. **6**, eaaz1708 (2020).

[17] A. E. Gongora, K. L. Snapp, E. Whiting, P. Riley, K. G. Reyes, E. F. Morgan, and K. A. Brown, Using simulation to accelerate autonomous experimentation: A case study using mechanics, iScience **24**, 102262 (2021).

[18] J. Behler, Atom-centered symmetry functions for constructing high-dimensional neural network potentials, J. Chem. Phys. **134**, 074106 (2011).

[19] N. Artrith, T. Morawietz, and J. Behler, High-dimensional neural-network potentials for multicomponent systems: Applications to zinc oxide, Phys. Rev. B **83**, 153101 (2011).

[20] N. Artrith and J. Behler, High-dimensional neural network potentials for metal surfaces: A prototype study for copper, Phys. Rev. B **85**, 045439 (2012).

[21] C. Schran, K. Brezina, and O. Marsalek, Committee neural network potentials control generalization errors and enable active learning, J. Chem. Phys. **153**, 104105 (2020).

[22] J. Behler, Four generations of high-dimensional neural network potentials, Chem. Rev. **121**, 10037 (2021).

[23] P. Friederich, F. Häse, J. Proppe, and A. Aspuru-Guzik, Machine-learned potentials for next-generation matter simulations, Nat. Mater. **20**, 750 (2021).

[24] E. V. Podryabinkin and A. V. Shapeev, Active learning of linearly parametrized interatomic potentials, Comput. Mater. Sci. **140**, 171 (2017).

[25] G. Sivaraman, A. N. Krishnamoorthy, M. Baur, C. Holm, M. Stan, G. Csányi, C. Benmore, and Á. Vázquez-Mayagoitia, Machine-learned interatomic potentials by active learning: amorphous and liquid hafnium dioxide, npj Comput. Mater. **6**, 104 (2020).

[26] B. Settles, *Active Learning Literature Survey* (Computer Science Technical Report 1648, University of Wisconsin-Madison Department of Computer Sciences, 2009).

[27] K. Gubaev, E. V. Podryabinkin, and A. V. Shapeev, Machine learning of molecular properties: Locality and active learning, J. Chem. Phys. **148**, 241727 (2018).

[28] A. L. Ankudinov, J. J. Rehr, J. J. Low, and S. R. Bare, Sensitivity of Pt x-ray absorption near edge structure to the morphology of small Pt clusters, J. Chem. Phys. **116**, 1911 (2002).

[29] A. L. Ankudinov and J. J. Rehr, Development of xafs theory, J. Synchrotron Radiat. **10**, 366 (2003).

[30] D. Bazin and J. J. Rehr, Limits and advantages of x-ray absorption near edge structure for nanometer scale metallic clusters, J. Phys. Chem. B **107**, 12398 (2003).

[31] G. Ciatto, A. Di Trolio, E. Fonda, P. Alippi, A. M. Testa, and A. A. Bonapasta, Evidence of Cobalt-Vacancy Complexes in $Zn_{1-x}Co_xO$ Dilute Magnetic Semiconductors, Phys. Rev. Lett. **107**, 127206 (2011).

[32] Q. Ma, J. T. Prater, C. Sudakar, R. A. Rosenberg, and J. Narayan, Defects in room-temperature ferromagnetic Cu-doped ZnO films probed by x-ray absorption spectroscopy, J. Phys.: Condens. Matter **24**, 306002 (2012).

[33] A. Kuzmin and J. Chaboy, Exafs and xanes analysis of oxides at the nanoscale, IUCrJ **1**, 571 (2014).

[34] C. D. Rankine and T. J. Penfold, Accurate, affordable, and generalizable machine learning simulations of transition metal x-ray absorption spectra using the xanes-net deep neural network, J. Chem. Phys. **156**, 164102 (2022).

[35] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. Von Lilienfeld, Quantum chemistry structures and properties of 134 kilo molecules, Sci. Data **1**, 140022 (2014).

[36] T. J. Penfold and C. D. Rankine, A deep neural network for valence-to-core x-ray emission spectroscopy, Mol. Phys., e2123406 (2022).

[37] A. Y.-T. Wang, R. J. Murdock, S. K. Kauwe, A. O. Oliynyk, A. Gurlo, J. Brgoch, K. A. Persson, and T. D. Sparks, Machine learning for materials scientists: An introductory guide toward best practices, Chem. Mater. **32**, 4954 (2020).

[38] N. Artrith, K. T. Butler, F.-X. Coudert, S. Han, O. Isayev, A. Jain, and A. Walsh, Best practices in machine learning for chemistry, Nat. Chem. **13**, 505 (2021).

[39] M. R. Carbone, S. Yoo, M. Topsakal, and D. Lu, Classification of local chemical environments from x-ray absorption spectra using supervised machine learning, Phys. Rev. Mater. **3**, 033604 (2019).

[40] M. R. Carbone, M. Topsakal, D. Lu, and S. Yoo, Machine-Learning X-Ray Absorption Spectra to Quantitative Accuracy, Phys. Rev. Lett. **124**, 156401 (2020).

[41] S. B. Torrisi, M. R. Carbone, B. A. Rohr, J. H. Montoya, Y. Ha, J. Yano, S. K. Suram, and L. Hung, Random forest machine learning models for interpretable x-ray absorption near-edge structure spectrum-property relationships, npj Comput. Mater. **6**, 109 (2020).

[42] E. J. Sturm, M. R. Carbone, D. Lu, A. Weichselbaum, and R. M. Konik, Predicting impurity spectral functions using machine learning, Phys. Rev. B **103**, 245118 (2021).

[43] C. Miles, M. R. Carbone, E. J. Sturm, D. Lu, A. Weichselbaum, K. Barros, and R. M. Konik, Machine learning of kondo physics using variational autoencoders and symbolic regression, Phys. Rev. B **104**, 235111 (2021).

[44] E. Hüllermeier and W. Waegeman, Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods, Mach. Learn. **110**, 457 (2021).

[45] R. Egele, R. Maulik, K. Raghavan, P. Balaprakash, and B. Lusch, Autodeuq: Automated deep ensemble with uncertainty quantification, arXiv:2110.13511.

[46] D. A. Nix and A. S. Weigend, Estimating the mean and variance of the target probability distribution, in *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)* (IEEE, 1994), Vol. 1, pp. 55–60.

[47] L. V. Jospin, H. Laga, F. Boussaid, W. Buntine, and M. Bennamoun, Hands-on bayesian neural networks-a tutorial for deep learning users, IEEE Comput. Intell. Mag. **17**, 29 (2022).

[48] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. **15**, 1929 (2014).

[49] R. Hu, Q. Huang, S. Chang, H. Wang, and J. He, The mbpep: A deep ensemble pruning algorithm providing high quality uncertainty prediction, Appl. Intell. **49**, 2942 (2019).

[50] A. G. Wilson and P. Izmailov, Bayesian deep learning and a probabilistic perspective of generalization, Adv. Neural Inf. Process. Syst. **33**, 4697 (2020).

[51] M. Tschannen, O. Bachem, and M. Lucic, Recent advances in autoencoder-based representation learning, arXiv:1812.05069 (2018).

[52] Y. Zhu, N. Zabaras, P.-S. Koutsourelakis, and P. Perdikaris, Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data, J. Comput. Phys. **394**, 56 (2019).

[53] X. Luo, B. T. Nadiga, Y. Ren, J. H. Park, W. Xu, and S. Yoo, A bayesian deep learning approach to near-term climate prediction, J. Adv. Model. Earth Syst. **14**, e2022MS003058 (2022).

[54] L. Ruddigkeit, R. Van Deursen, L. C. Blum, and J.-L. Reymond, Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17, J. Chem. Inf. Model. **52**, 2864 (2012).

[55] J. J. Rehr, J. J. Kas, F. D. Vila, M. P. Prange, and K. Jorissen, Parameter-free calculations of x-ray spectra with FEFF9, Phys. Chem. Chem. Phys. **12**, 5503 (2010).

[56] L. Himanen, M. O. J. Jäger, E. V. Morooka, F. Federici Canova, Y. S. Ranawat, D. Z. Gao, P. Rinke, and A. S. Foster, DScribe: Library of descriptors for machine learning in materials science, Comput. Phys. Commun. **247**, 106949 (2020).

[57] A. P. Bartók, R. Kondor, and G. Csányi, On representing chemical environments, Phys. Rev. B **87**, 184115 (2013).

[58] H. Huo and M. Rupp, Unified representation of molecules and crystals for machine learning, Mach. Learn. Sci. Technol. **3**, 045017 (2022).

[59] M. Gastegger, L. Schwiedrzik, M. Bittermann, F. Berzsenyi, and P. Marquetand, wacsf-weighted atom-centered symmetry functions as descriptors in machine learning potentials, J. Chem. Phys. **148**, 241709 (2018).

[60] N. Artrith and A. Urban, An implementation of artificial neural-network potentials for atomistic materials simulations: Performance for TiO_2, Comput. Mater. Sci. **114**, 135 (2016).

[61] A. L. Ankudinov, B. Ravel, J. J. Rehr, and S. D. Conradson, Real-space multiple-scattering calculation and interpretation of x-ray-absorption near-edge structure, Phys. Rev. B **58**, 7565 (1998).

[62] J. J. Rehr and R. C. Albers, Theoretical approaches to x-ray absorption fine structure, Rev. Mod. Phys. **72**, 621 (2000).

[63] L. Van der Maaten and G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. **9**, 2579 (2008).

[64] M. Wattenberg, F. Viégas, and I. Johnson, How to use t-SNE effectively, Distill **1**, e2 (2016).

[65] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. **12**, 2825 (2011).

[66] R. J. Tibshirani and B. Efron, An Introduction to the Bootstrap, Monogr. Stat. Appl. Probab. **57**, 1 (1993).

[67] B. Bakker and T. Heskes, Clustering ensembles of neural network models, Neural Netw. **16**, 261 (2003).

[68] I. J. Goodfellow, J. Shlens, and C. Szegedy, Explaining and harnessing adversarial examples, arXiv:1412.6572 (2014).

[69] M. R. Carbone, When not to use machine learning: A perspective on potential and limitations, MRS Bull. **47**, 968 (2022).

[70] T. Flynn and S. Yoo, Change detection with the kernel cumulative sum algorithm, in *2019 IEEE 58th Conference on Decision and Control (CDC)* (IEEE, 2019), pp. 6092–6099.

[71] R. L. McGreevy and L. Pusztai, Reverse monte carlo simulation: a new technique for the determination of disordered structures, Mol. Simul. **1**, 359 (1988).

[72] R. L. McGreevy, Reverse monte carlo modelling, J. Phys.: Condens. Matter **13**, R877 (2001).

[73] D. Whitley, A genetic algorithm tutorial, Stat. Comput. **4**, 65 (1994).

[74] All software used in this work can be found open source at github.com/AI-multimodal/XAS-NNE. All data used in this work, including FEFF spectra input/output files, featurized data, and the neural network ensembles can be found open access at https://dx.doi.org/10.5281/zenodo.7554888.

[75] http://quantum-machine.org/datasets/

[76] S. P. Ong, W. D. Richards, A. Jain, G. Hautier, M. Kocher, S. Cholia, D. Gunter, V. L. Chevrier, K. A. Persson, and G. Ceder, Python materials genomics (pymatgen): A robust, open-source python library for materials analysis, Comput. Mater. Sci. **68**, 314 (2013).

[77] A. H. Larsen, J. J. Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, J. Friis, M. N. Groves, B. Hammer, C. Hargus *et al.*, The atomic simulation environment-a python library for working with atoms, J. Phys.: Condens. Matter **29**, 273002 (2017).

[78] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai *et al.*, Pytorch: An imperative style, high-performance deep learning library, Adv. Neural Inf. Process. Syst. **32**, 8026 (2019).

[79] W. Falcon and The PyTorch Lightning team, PyTorch Lightning (2019), doi: 10.5281/zenodo.3828935.

[80] G. Kresse and D. Joubert, From ultrasoft pseudopotentials to the projector augmented-wave method, Phys. Rev. B **59**, 1758 (1999).