Chemprop: A Machine Learning Package for Chemical Property Prediction

Esther Heid, Kevin P. Greenman, Yunsie Chung, Shih-Cheng Li, David E. Graff, Florence H. Vermeire, Haoyang Wu, William H. Green, and Charles J. McGill*

Cite This: https://doi.org/10.1021/acs.jcim.3c01250

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: Deep learning has become a powerful and frequently employed tool for the prediction of molecular properties, thus creating a need for open-source and versatile software solutions that can be operated by nonexperts. Among the current approaches, directed message-passing neural networks (D-MPNNs) have proven to perform well on a variety of property prediction tasks. The software package Chemprop implements the D-MPNN architecture and offers simple, easy, and fast access to machine-learned molecular properties. Compared to its initial version, we present a multitude of new Chemprop functionalities such as the support of multimolecule properties, reactions, atom/bond-level properties, and spectra. Further, we incorporate various uncertainty quantification and calibration methods along with related metrics as well as pretraining and transfer learning workflows, improved hyperparameter optimization, and other customization options concerning loss functions or atom/bond features. We benchmark D-MPNN models trained using Chemprop with the new reaction, atom-level, and spectra functionality on a variety of property prediction data sets, including MoleculeNet and SAMPL, and observe state-of-the-art performance on the prediction of water-octanol partition coefficients, reaction barrier heights, atomic partial charges, and absorption spectra. Chemprop enables out-of-the-box training of D-MPNN models for a variety of problem settings in fast, user-friendly, and open-source software.

Molecule, atom, reaction, and multimolecule properties from molecular graphs

chemprop

MPNN → FFN → Property

1. INTRODUCTION

Machine learning in general and especially deep learning has become a powerful tool in various fields of chemistry. Applications range from the prediction of physicochemical[1−9] and pharmacological[10] properties of molecules to the design of molecules or materials with certain properties,[11−13] the exploration of chemical synthesis pathways,[14−27] or the prediction of properties important for chemical analysis like IR,[28] UV/vis,[29] or mass spectra.[30−33]

Many combinations of molecular representations and model architectures have been developed to extract features from molecules and predict molecular properties. Molecules can be represented as graphs, strings, precomputed feature vectors, or sets of atomic coordinates and processed using graph-convolutional neural networks, transformers, or feed-forward neural networks to train predictive models. While early works focused on handmade features or simple fingerprinting methods combined with kernel regression or neural networks,[34] the current state-of-the-art has shifted to end-to-end trainable models which directly learn to extract their own features.[35] Here, the models can achieve extreme complexity based on the mechanisms of information exchange between parts of the molecule. For example, graph convolutional neural networks (GCNNs) extract local information from the molecular graph for single or small groups of atoms and use

that information to update the immediate neighborhood.[1−3,36] They offer robust performance for properties dependent on the local structure and if the three-dimensional conformation of a molecule is not known or not relevant for a prediction task. Graph attention transformers allow for a less local information exchange via attention layers, which learn to accumulate the features of atoms both close and far away in the graph.[37,38] Another important line of research comprises the prediction of properties dependent on the three-dimensional conformation of a molecule, such as the prediction of properties obtained from quantum mechanics.[2,39−41] Finally, transformer models from natural language processing can be trained on string representations such as SMILES or SELFIES, also leading to promising results.[42−45] In this work, we discuss our application of GCNNs, namely, Chemprop,[36] a directed-message passing algorithm derived from the seminal work of Gilmer et al.[1]

An early version of Chemprop was published in ref 36. Since then, the software has substantially evolved and now includes a
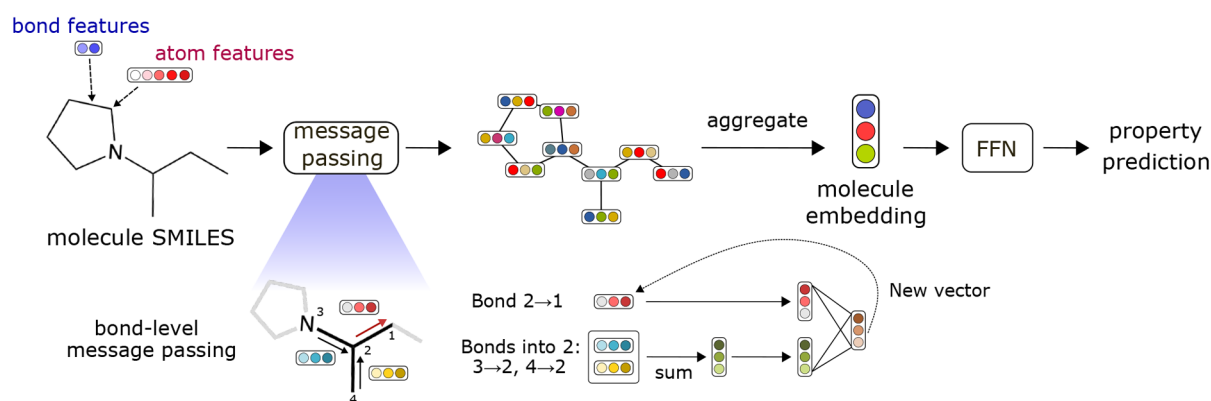
Figure 1. Overview of the architecture of Chemprop. The message passing update of the hidden vector for directed edge $2 \rightarrow 1$ is expanded for demonstration.

vast collection of new features. For example, Chemprop is now able to predict properties for systems containing multiple molecules, such as solute/solvent combinations or reactions with and without solvent. It can train on molecular targets, spectra, or atom/bond-level targets and output the latent representation for analysis of the learned feature embedding. Available uncertainty metrics include popular approaches, such as ensembling, mean-variance estimation, and evidential learning. Chemprop is thus a general and versatile deep learning toolbox and enjoys a wide user base.

The remainder of the article is structured as follows: First, we summarize the architecture of Chemprop. We discuss a selection of Chemprop features with a focus on features introduced after the initial release of Chemprop. We then conclude the main body of the article and provide details on the data and software, which we have open-sourced including all scripts to allow for full reproducibility. Alongside the main body of this article, we provide Supporting Information that contains further model design details; descriptions of the data acquisition, preprocessing, and splitting of all data sets used in benchmarking; and the results of Chemprop benchmarks on a variety of data sets showcasing its performance on both simple and advanced prediction tasks.

2. MODEL STRUCTURE

Chemprop consists of four modules: (1) a local features encoding function, (2) a directed message passing neural network (D-MPNN) to learn atomic embeddings from the local features, (3) an aggregation function to join atomic embeddings into molecular embeddings, and (4) a standard feed-forward neural network (FFN) for the transformation of molecular embeddings to target properties, summarized in Figure 1. The D-MPNN is a class of graph-convolutional neural networks (GCNN), which updates hidden representations of the vertices V and edges E of a graph G based on the local environment. In the following, we use bold lower case to denote vectors, bold upper case to denote matrices, and italic light font for scalars and objects.

For a molecule, the molecule SMILES string is used as input, which is then transformed to a molecular graph using RDKit,[46] where atoms correspond to vertices and bonds to edges. Initial features are constructed based on the identity and topology of each atom and bond. For each vertex v, initial feature vectors $\{\mathbf{x}_v | v \in V\}$ are obtained from a one-hot encoding of the atomic number, number of bonds linked to each atom, formal charge,

chirality (if encoded in the SMILES), number of hydrogens, hybridization, and aromaticity of the atom, as well as the atomic mass (divided by 100 for scaling). For each edge e, initial feature vectors $\{\mathbf{e}_{vw} | \{v, w\} \in E\}$ arise from the bond type, whether the bond is conjugated or in a ring, and whether it contains stereochemical information, such as a cis/trans double bond. The D-MPNN uses directed edges in a graph to pass information, where each undirected edge (bond) has two corresponding directed edges, one in each direction. Initial directed edge features $\mathbf{e}_{vw}^{d}$ are obtained via simple concatenation of the atom features of the first atom of a bond $\mathbf{x}_v$ to the respective undirected bond features $\mathbf{e}_{vw}$

$$\mathbf{e}_{vw}^{d} = \text{cat}(\mathbf{x}_v, \mathbf{e}_{vw}) \tag{1}$$

where $\text{cat}()$ denotes simple concatenation. The directed edges $\mathbf{e}_{vw}^{d}$ and $\mathbf{e}_{wv}^{d}$ are distinguished only by the choice of which atom to use in eq 1. Chemprop also offers the option to read in custom atom and bond features in addition to or as a replacement for the default features, as described in SI Section S1.2, and thus offers full control of the initial features. In summary, Module 1 of Chemprop constructs atom and directed bond feature vectors $\mathbf{x}_v$ and $\mathbf{e}_{vw}^{d}$ from the input molecules.

The initial atom and bond features are then passed to a D-MPNN. In a D-MPNN structure, messages are passed between directed edges rather than between nodes as would be done in a traditional MPNN. To construct the hidden directed edge features $\mathbf{h}_{vw}^{0}$ of hidden size h, the initial directed edge features $\mathbf{e}_{vw}^{d}$ are passed through a single neural network layer with learnable weights $\mathbf{W}_i \in \mathbb{R}^{h \times h_i}$

$$\mathbf{h}_{vw}^{0} = \tau(\mathbf{W}_i \mathbf{e}_{vw}^{d}) \tag{2}$$

and a nonlinear activation function τ which can be chosen by the user (default ReLU). The size h of $\mathbf{h}_{vw}^{0}$ can be chosen by the user (default 300). The size of $\mathbf{e}_{vw}^{d}$, which we term h_i, is set by the lengths of initial feature encodings, per eq 1. The directed edge features are then iteratively updated based on the local environment via T (default 3) message passing steps

$$\mathbf{h}_{vw}^{t+1} = \tau \left(\mathbf{h}_{vw}^{0} + \mathbf{W}_h \sum_{k \in \{N(v) \backslash w\}} \mathbf{h}_{kv}^{t} \right) \tag{3}$$

until $t + 1 = T$, where $\mathbf{W}_h \in \mathbb{R}^{h \times h}$ and $N(v) \backslash w$ denotes the neighbors of node v excluding w. The opposite facing directed

Table 1. Selected Published Studies Based on Chemprop

ref	Year	Prediction	ref	Year	Prediction
10	2020	Growth inhibitory activity against *E. coli*; led to an identification of a potential new drug	51	2022	Absorption, distribution, metabolism, excretion (ADME) properties for drug discovery
52	2021	Chemical synergy against SARS-CoV-2; identified two drug combinations with antiviral synergy in vitro	53	2023	Growth inhibitory activity against *A. baumannii*; led to an identification of a potential new drug
28	2021	IR spectra of molecules	54	2023	Fuel properties
55	2021	Atomic charges, Fukui indices, NMR constants, bond lengths, and bond orders	56	2023	Critical properties, acentric factor, and phase change properties
57	2021	Lipophilicity	58	2023	Molecular optical peaks
59	2022	Reaction rates and barrier heights	60	2023	Lipophilicity
61	2022	Barrier heights of reactions	62	2023	Solvent effects on reaction rate constants
29	2022	Molecular optical peaks	63	2023	Vapor pressure in the low volatility regime
64	2022	Solvation free energy, solvation enthalpy, and Abraham solute descriptors	65	2023	Molecular optical peaks and partition coefficients for closed-loop active learning
66	2022	Solid solubility of organic solutes in water and organic solvents	67	2023	Senolytic activity of compounds to selectively target senescent cells
68	2022	Activity coefficients	69	2023	Toxicity measurements using 12 nuclear receptor signaling and stress response pathways

edge is excluded from the message passing update for increased numerical stability (see Mahé et al.[47]). Finally, the updated hidden states $\mathbf{h}_{vw}^T$ are aggregated into atomic embeddings via

$$\mathbf{h}_v = \tau(\mathbf{W}_o \mathbf{q}) \tag{4}$$

where $\mathbf{q}$ is a concatenation of the initial atom features $\mathbf{x}_v$ and the sum of all incoming directed edge hidden states

$$\mathbf{q} = \mathrm{cat}\left(\mathbf{x}_v, \sum_{w \in N(v)} \mathbf{h}_{wv}^T\right) \tag{5}$$

with $\mathbf{W}_o \in \mathbb{R}^{h \times h_o}$. Here, h_o is the size of $\mathbf{q}$, i.e., the sum of the hidden size h and the size of $\mathbf{x}_v$. In summary, in Module 2, the D-MPNN weights $\mathbf{W}_i$, $\mathbf{W}_h$, and $\mathbf{W}_o$ are learned from the training data, outputting learnable atomic embeddings $\mathbf{h}_v$. Customizations and hyperparameter tuning include the choice of the activation function τ, the hidden size h, and the number of message passing steps T. Chemprop offers the option to add bias terms to all neural network layers (defaults to False).

The atomic embeddings $\mathbf{h}_v$ of all atoms in a molecule are then aggregated into a single molecular embedding $\mathbf{h}_m$ via

$$\mathbf{h}_m = \mathrm{cat}(\mathbf{h}_m', \mathbf{x}_m) \tag{6}$$

with

$$\mathbf{h}_m' = \sum_{v \in V} \mathbf{h}_v \tag{7}$$

where $\mathbf{x}_m$ is an optional vector of additional molecular features. Chemprop offers three aggregation options: summation (as shown in eq 7), a scaled sum (divided by a user specified scaler, called a norm within Chemprop), or an average (the default aggregation). Schweidtmann et al.[48] compared the performance of such aggregation functions for different data sets. The optional additional molecular features, $\mathbf{x}_m$, may be provided features from outside sources (SI Section S2) or generated engineered fingerprints (Morgan circular fingerprints[49] and RDKIT 2D fingerprints[46] are implemented in Chemprop). By default, $\mathbf{x}_m$ is empty such that the molecular embedding is simply an aggregation over atomic embeddings, i.e., $\mathbf{h}_m = \mathbf{h}_m'$. In summary, Module 3 produces molecular embeddings $\mathbf{h}_m$ of length h plus the size of $\mathbf{x}_m$. Chemprop offers the option to circumvent Modules 1−3 and only using $\mathbf{x}_m$ as fixed molecular embedding, so that $\mathbf{h}_m = \mathbf{x}_m$.

Finally, in the last module, molecular target properties are learned from the molecular embeddings $\mathbf{h}_m$ via a feed-forward neural network, where the number of layers (default 2) and the number of hidden neurons (default 300) can be chosen by the user. The number of input neurons is set by the length of $\mathbf{h}_m$, and the number of output neurons is set by the number of targets. The activation function between linear layers is set to be the same as in the D-MPNN, and bias is turned on per default. For binary classification tasks, the final model output is passed through a sigmoid function to constrain values to the range (0,1). For multiclass classification, the final model output is transformed with a softmax function, such that the classification scores sum to 1 across classes.

Chemprop is fully end-to-end trainable, so that the weights for D-MPNN and FFN are updated simultaneously. Users have the option to train models using cross-validation and ensembles of submodels. By default, a single model is trained on a random data split for 30 epochs. We note that small data sets need a much larger number of epochs to train and advise to check for convergence of the learning curve. Chemprop uses the Adam optimizer.[50] The default learning rate schedule increases the learning rate linearly from 10^{-4} to 10^{-3} for the first two warmup epochs and then decreases the learning rate exponentially from 10^{-3} to 10^{-4} for the remaining epochs. By default, a batch size of 50 data points is used for each optimizer step. Early stopping and dropout are available as means of regularization. The PyTorch backend of Chemprop enables seamless GPU acceleration of both model training and inference. The acceleration of training and inference processes when used with a GPU can be significant, as shown in SI Section S3.3.

3. DISCUSSION OF FEATURES

Table 1 lists a nonexhaustive selection of studies based on Chemprop, showcasing its versatility and applicability for the prediction of a large variety of chemical properties, but also its ease of use. Models can be trained and tested with a single line on the command line (or a few lines of python code) and a user-supplied CSV file (see SI for some examples). In the following, we discuss specialty options introduced since its first release.

3.1. Additional Features. Chemprop can take additional features at the molecule-, atom-, or bond-level as input. While Chemprop often generates accurate models without requiring
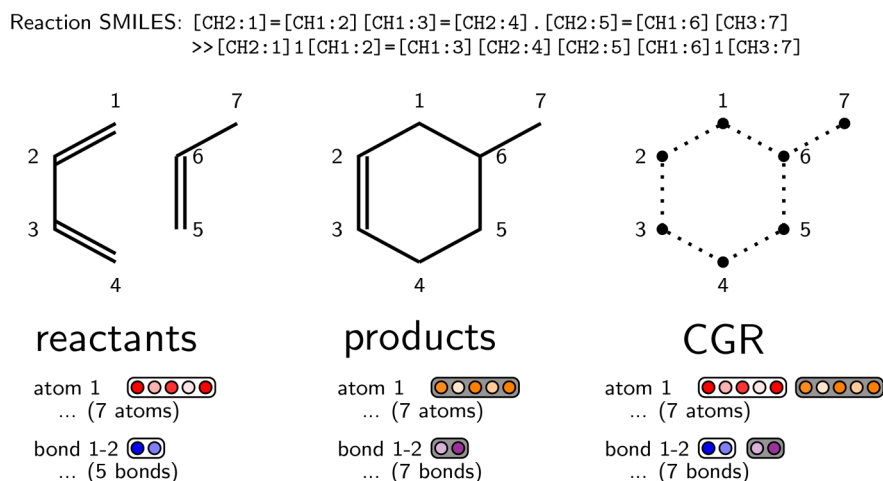
Reaction SMILES: [CH2:1]=[CH1:2][CH1:3]=[CH2:4].[CH2:5]=[CH1:6][CH3:7]
>>[CH2:1]1[CH1:2]=[CH1:3][CH2:4][CH2:5][CH1:6]1[CH3:7]

Figure 2. Construction of the condensed graph of reaction (CGR) of an example reaction. The vertices and edges are obtained as the union of the respective reactant and product vertices and edges. The features are obtained as a combination of the reactant (white background) and product (gray background) features for atoms and bonds.

any input beyond the SMILES, it has been shown that outside information added as additional features can further improve performance.[36,64] Users can provide their custom additional features by adding keywords and paths to the data files containing the features. See SI for command-line arguments and details.

3.2. Multimolecule Models. Chemprop can also train on a data set containing more than one molecule as input. For example, when properties related to solvation need to be predicted, both a solute and a solvent are required as input to the model. Users can provide multiple molecules as inputs to Chemprop. When multiple molecules are used, by default Chemprop trains a separate D-MPNN for each molecule (Figure S1a). If the option --mpn_shared is specified, then the same D-MPNN is used for all molecules (Figure S1b). The embeddings of the different molecules are then concatenated prior to the FFN. Note that the current implementation of multiple molecules in Chemprop does not ensure permutational invariance toward the input molecules. This is suited to situations where the input molecules have different roles, e.g., molecule 1 = solute, molecule 2 = solvent. For additional input information and a figure depicting the multimolecule model structure, see SI.

3.3. Reaction Support. Chemprop supports the input of atom-mapped reactions, i.e., pairs of reactants and products SMILES connected via the "≫" symbol, by using the keyword --reaction. The pair of reactants and products is transformed into a single pseudomolecule, namely, the condensed graph of reaction (CGR), and then passed to a regular D-MPNN block. The construction of a CGR within Chemprop is described in detail in ref 59 and summarized in the following. In general, the input of a reaction vs a molecule only affects the setup of the graph object and its initial features, but not any other part of the architecture. The graph of a reaction has a different set of edges E as the graph of a molecule, as shown in Figure 2 for an example Diels–Alder reaction. To build the CGR pseudomolecule, the set of atoms is obtained as the union of the sets of atoms in the reactants and products. Similarly, the set of bonds is obtained as the union of the sets of bonds in the reactants and products. Once constructed, the CGR is passed through the D-MPNN and other model architecture components in the same way a

molecular graph would. Optionally, Chemprop accepts an additional molecule object as input, such as a solvent, a reagent, etc. which is passed to its own D-MPNN similar to the multimolecule model. The output of the reaction D-MPNN and molecule D-MPNN is concatenated after atomic aggregation, before the FFN. This option is available via the reaction_solvent keyword. See SI for further commandline options and details.

3.4. Spectra Data Support. Chemprop supports the prediction of whole-spectrum properties for molecules. An initial version of this capability was discussed in ref 28 for use with IR absorbance spectra. Targets for the spectra data set type are composed of an array of intensity values, set at fixed bin locations typically specified in terms of wavelengths, frequencies, or wavenumbers. Spectral Information Divergence was originally developed as a method of comparing spectra to reference databases[70] and is adapted in Chemprop to be used as a loss function, considering the deviation of the spectrum as a whole rather than independently at each bin location. The treatment of spectra can handle targets with gaps or missing values within a data set. With the expectation that spectra will often be collected in systems where a portion of the range will be obscured or invalid (e.g., from solvent absorbance), Chemprop can create exclusion regions in specified spectra where no predictions are provided and targets are ignored for training purposes.

3.5. Latent Representations. Graph neural networks enable learning both molecular representation and property end-to-end directly from the molecular graph. As detailed above in Section 2, the learned node representations are aggregated into a molecule-level representation after the message-passing phase, which we refer to as the "learned fingerprint." This embedding is then further fed into the FFN network. Within the FFN, we consider the final hidden representation, which we refer to as the "ffn embedding". Both of these vectors are latent representations of a molecule as it relates to a particular trained model. Molecule latent representations can be useful for data clustering or used as additional features in other models. Chemprop supports the calculation of either from a trained model for a given set of molecules.

3.6. Loss Function Options. Chemprop can train models according to many common loss functions. The loss functions available for a given task are determined by the data set type (regression, classification, multiclass, or spectra). Regression models can be trained with mean squared error (MSE), bounded MSE (which allows inequalities as targets), or negative log-likelihood (NLL) based on prediction uncertainty distributions consistent with mean-variance estimation (MVE)[71] or evidential uncertainty.[72] Classification tasks default to the binary cross entropy loss and have additional options of Matthews correlation coefficient (MCC) and Dirichlet (evidential classification).[73] Cross entropy and MCC are also available for multiclass problems. There are two options available for training on spectra: spectral information divergence (SID)[70] and first-order Wasserstein distance (a.k.a. earthmover's distance).[74] Loss functions must be differentiable since they are used to calculate gradients that update the model parameters, but Chemprop also provides the option to use several nondifferentiable metrics for model evaluation.

3.7. Transfer Learning. Transfer learning is a general strategy of using information gained through the training of one model to inform and improve the training of a related model. Often, this strategy is used to transfer information from a previously trained model of a large data set to a model of a small data set in order to improve the performance of the model of the small data set. The simplest method of transfer learning would be taking predictions or latent representations from one model and supplying them as additional features to another model (Sections 3.1 and 3.5).

In Chemprop, different strategies are available to transfer learned model parameters from a previously trained model to a new model as a form of transfer learning, as shown in Figure 3.
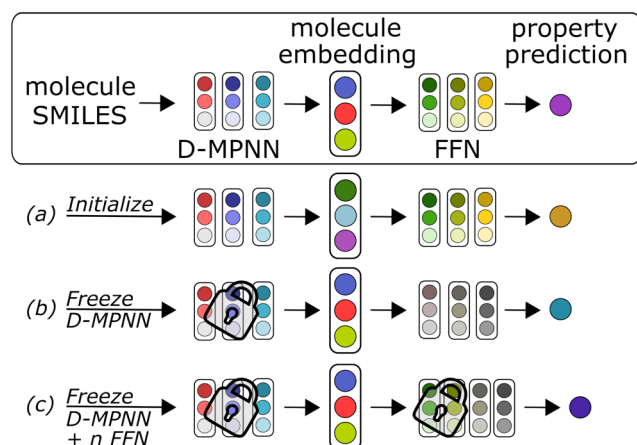

Figure 3. Options to transfer model parameters from a pretrained model (squared) to a new model, by (a) initializing the new model parameters or by freezing the (b) D-MPNN layers and (c) *n* FFN layers.

A pretrained model may be used to initialize a new model with normal updating of the transferred weights in training. Alternatively, parameters from the transferred model can be frozen, holding them constant during training. Freezing parameters always include the D-MPNN weights but can be specified to include some FFN layers as well. See SI for the corresponding arguments.

3.8. Hyperparameter Optimization. Chemprop provides a command-line utility, allowing for the simple initiation of hyperparameter optimization jobs. Options for the hyperparameter job such as how many trials to carry out and which hyperparameters to include in the search (options in Table S3) can be specified with simple command-line arguments. The optimization initially uses randomly sampled trials, followed by targeted sampling using the Tree-structured Parzen Estimator algorithm.[75,76]

Hyperparameter optimization is often the most resource-intensive step in model training. In order to search a large parameter space adequately, a large number of trials is needed. Chemprop allows for parallel operation of multiple hyperparameter optimization instances, removing the need to carry out all trials in series and reducing the wall time needed to perform the optimization significantly. See SI for the available hyperparameter options and other details.

3.9. Uncertainty Tools. Chemprop includes a variety of popular uncertainty estimation, calibration, and evaluation tools. The estimation methods include deep ensembles,[77] dropout,[78] mean-variance estimation (MVE),[71] and evidential,[72] as well as a special version of ensemble variance for spectral predictions,[28] and the inherently probabilistic outputs of classification models. After estimating the uncertainty in a model's predictions, it is often helpful to calibrate these uncertainties to improve their performance on new predictions. We provide four such methods for regression tasks (*z*-scaling,[79] t-scaling, Zelikman's CRUDE,[80] and MVE weighting[81]) and two for classification (single-parameter Platt scaling[82] and isotonic regression[83]). In addition to standard metrics such as RMSE, MAE, etc. for evaluating predictions, Chemprop also includes several metrics specifically for evaluating the quality of uncertainty estimates. These include negative log likelihood, Spearman rank correlation, expected normalized calibration error (ENCE),[84] and miscalibration area.[84] Any valid classification or multiclass metric used to assess predictions can also be used to assess uncertainties.

3.10. Atom/Bond-Level Targets. Chemprop supports a multitask constrained D-MPNN architecture for predicting atom- and bond-level properties, such as charge density or bond length. This model enables a D-MPNN to be trained on multiple atomic and bond properties simultaneously, though unlike molecular property targets, they do not share a single FFN. Optionally, an attention-based constraining method may be used to enforce that predicted atomic or bond properties sum to a specified molecular net value, such as the overall charge of a molecule. An initial, more limited version of this capability was developed in ref 55. For details on the input formats to be used for atom/bond targets, both for training and for inference, see SI.

4. BENCHMARKING

See SI for general performance benchmarks, benchmarks using specific Chemprop features (atom/bond-level targets, reaction support, multimolecule models, spectra prediction, and uncertainty estimation), and system timing benchmarks.

5. CONCLUSION

We have presented the software package Chemprop, a powerful toolbox for machine learning of the chemical properties of molecules and reactions. Significant improvements have been made to the software since its initial release

and study,[36] including the support of multimolecule properties, reactions, atom/bond-level properties, and spectra. Additionally, several state-of-the-art approaches to estimate the uncertainty of predictions have been incorporated as well as pretraining and transfer learning procedures. Furthermore, the code now offers a variety of customization options, such as custom atom and bond features, a large variety of loss functions, and the ability to save the learned feature embeddings for subsequent use with different algorithms. We have showcased and benchmarked Chemprop on a variety of example tasks and data sets and have found competitive performances for molecular property prediction compared to other approaches available on public leaderboards. In summary, Chemprop is a powerful, fast, and convenient tool to learn conformation-independent properties of molecules, sets of molecules, or reactions.

ASSOCIATED CONTENT

Data Availability Statement

Chemprop, including all features described in this paper, is available under the open-source MIT License on GitHub, github.com/chemprop/chemprop. An extensive documentation including tutorials is available online,[85] including a workshop on YouTube.[86] Scripts and data splits to fully reproduce this study are available on GitHub, github.com/chemprop/chemprop_benchmark, and on Zenodo, doi.org/10.5281/zenodo.8174267, respectively.

Ⓢ Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.3c01250.

> Additional software details and usage examples, data set and data handling details for benchmarking, and results of software benchmarks (general performance, feature demonstrations, timing) (PDF)

AUTHOR INFORMATION

Corresponding Author

Charles J. McGill − *Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, United States; Department of Chemical and Life Science Engineering, Virginia Commonwealth University, Richmond, Virginia 23284, United States;* ⓞ orcid.org/0000-0003-2704-7717; Email: mcgillc2@vcu.edu

Authors

Esther Heid − *Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, United States; Institute of Materials Chemistry, TU Wien 1060 Vienna, Austria;* ⓞ orcid.org/0000-0002-8404-6596

Kevin P. Greenman − *Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, United States;* ⓞ orcid.org/0000-0002-6466-1401

Yunsie Chung − *Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, United States;* ⓞ orcid.org/0000-0002-3097-010X

Shih-Cheng Li − *Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, United States; Department of Chemical Engineering, National Taiwan University, Taipei 10617, Taiwan*

David E. Graff − *Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, United States; Department of Chemistry and Chemical Biology, Harvard University, Cambridge, Massachusetts 02138, United States;* ⓞ orcid.org/0000-0003-1250-3329

Florence H. Vermeire − *Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, United States; Department of Chemical Engineering, KU Leuven, B-3001 Leuven, Belgium*

Haoyang Wu − *Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, United States;* ⓞ orcid.org/0000-0002-0644-7554

William H. Green − *Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, United States;* ⓞ orcid.org/0000-0003-2603-9694

Complete contact information is available at:
https://pubs.acs.org/10.1021/acs.jcim.3c01250

Notes

The authors declare no competing financial interest.

REFERENCES

(1) Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. Neural Message Passing for Quantum Chemistry. *Proceedings of the International Conference on Machine Learning* **2017**, 1263−1272.

(2) Gasteiger, J.; Groß, J.; Günnemann, S. Directional Message Passing for Molecular Graphs. *Proceedings of the International Conference on Learning Representations*, **2003**, arXiv:2003.03123.

(3) Zhang, S.; Liu, Y.; Xie, L. Molecular Mechanics-Driven Graph Neural Network with Multiplex Graph for Molecular Structures. *Machine Learning for Molecules Workshop at NeurIPS*, **2020**, arXiv:2011.07457.

(4) Vermeire, F. H.; Chung, Y.; Green, W. H. Predicting Solubility Limits of Organic Solutes for a Wide Range of Solvents and Temperatures. *J. Am. Chem. Soc.* **2022**, 144, 10785−10797.

(5) Dobbelaere, M. R.; Ureel, Y.; Vermeire, F. H.; Tomme, L.; Stevens, C. V.; Van Geem, K. M. Machine Learning for Physicochemical Property Prediction of Complex Hydrocarbon Mixtures. *Ind. Eng. Chem. Res.* **2022**, 61, 8581−8594.

(6) Dobbelaere, M. R.; Plehiers, P. P.; Van de Vijver, R.; Stevens, C. V.; Van Geem, K. M. Learning Molecular Representations for Thermochemistry Prediction of Cyclic Hydrocarbons and Oxygenates. *J. Phys. Chem. A* **2021**, *125*, 5166−5179.

(7) Rittig, J. G.; Ben Hicham, K.; Schweidtmann, A. M.; Dahmen, M.; Mitsos, A. Graph Neural Networks for Temperature-Dependent Activity Coefficient Prediction of Solutes in Ionic Liquids. *Comput. Chem. Eng.* **2023**, *171*, 108153.

(8) Rittig, J. G.; Ritzert, M.; Schweidtmann, A. M.; Winkler, S.; Weber, J. M.; Morsch, P.; Heufer, K. A.; Grohe, M.; Mitsos, A.; Dahmen, M. Graph Machine Learning for Design of High-Octane Fuels. *AIChE J.* **2023**, *69*, e17971.

(9) Fleitmann, L.; Ackermann, P.; Schilling, J.; Kleinekorte, J.; Rittig, J. G.; vom Lehn, F.; Schweidtmann, A. M.; Pitsch, H.; Leonhard, K.; Mitsos, A.; Bardow, A.; Dahmen, M. Molecular Design of Fuels for Maximum Spark-Ignition Engine Efficiency by Combining Predictive Thermodynamics and Machine Learning. *Energ. Fuel.* **2023**, *37*, 2213−2229.

(10) Stokes, J. M.; Yang, K.; Swanson, K.; Jin, W.; Cubillos-Ruiz, A.; Donghia, N. M.; MacNair, C. R.; French, S.; Carfrae, L. A.; Bloom-Ackerman, Z.; et al. A Deep Learning Approach to Antibiotic Discovery. *Cell* **2020**, *180*, 688−702.

(11) De Cao, N.; Kipf, T. MolGAN: An Implicit Generative Model for Small Molecular Graphs. *arXiv Preprint*, **2022**, arXiv:1805.11973.

(12) Dan, Y.; Zhao, Y.; Li, X.; Li, S.; Hu, M.; Hu, J. Generative Adversarial Networks (GAN) Based Efficient Sampling of Chemical Composition Space for Inverse Design of Inorganic Materials. *Npj Comput. Mater.* **2020**, *6*, 1−7.

(13) Gómez-Bombarelli, R.; Wei, J. N.; Duvenaud, D.; Hernández-Lobato, J. M.; Sánchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Adams, R. P.; Aspuru-Guzik, A. Automatic Chemical Design using a Data-Driven Continuous Representation of Molecules. *ACS Cent. Sci.* **2018**, *4*, 268−276.

(14) Wei, J. N.; Duvenaud, D.; Aspuru-Guzik, A. Neural Networks for the Prediction of Organic Chemistry Reactions. *ACS Cent. Sci.* **2016**, *2*, 725−732.

(15) Segler, M. H.; Waller, M. P. Neural-Symbolic Machine Learning for Retrosynthesis and Reaction Prediction. *Chem. Eur. J.* **2017**, *23*, 5966−5971.

(16) Coley, C. W.; Barzilay, R.; Jaakkola, T. S.; Green, W. H.; Jensen, K. F. Prediction of Organic Reaction Outcomes using Machine Learning. *ACS Cent. Sci.* **2017**, *3*, 434−443.

(17) Coley, C. W.; Green, W. H.; Jensen, K. F. Machine Learning in Computer-Aided Synthesis Planning. *Acc. Chem. Res.* **2018**, *51*, 1281−1289.

(18) Szymkuć, S.; Gajewska, E. P.; Klucznik, T.; Molga, K.; Dittwald, P.; Startek, M.; Bajczyk, M.; Grzybowski, B. A. Computer-Assisted Synthetic Planning: The End of the Beginning. *Angew. Chem., Int. Ed.* **2016**, *55*, 5904−5937.

(19) Segler, M. H.; Preuss, M.; Waller, M. P. Planning Chemical Syntheses with Deep Neural Networks and Symbolic AI. *Nature* **2018**, *555*, 604−610.

(20) Kayala, M. A.; Azencott, C.-A.; Chen, J. H.; Baldi, P. Learning to Predict Chemical Reactions. *J. Chem. Inf. Model.* **2011**, *51*, 2209−2222.

(21) Kayala, M. A.; Baldi, P. ReactionPredictor: Prediction of Complex Chemical Reactions at the Mechanistic Level using Machine Learning. *J. Chem. Inf. Model.* **2012**, *52*, 2526−2540.

(22) Fooshee, D.; Mood, A.; Gutman, E.; Tavakoli, M.; Urban, G.; Liu, F.; Huynh, N.; Van Vranken, D.; Baldi, P. Deep Learning for Chemical Reaction Prediction. *Mol. Syst. Des. Eng.* **2018**, *3*, 442−452.

(23) Bradshaw, J.; Kusner, M. J.; Paige, B.; Segler, M. H.; Hernández-Lobato, J. M. A Generative Model for Electron Paths. *Proceedings of the International Conference on Learning Representations*, **2019**, arXiv:1805.10970.

(24) Bi, H.; Wang, H.; Shi, C.; Coley, C.; Tang, J.; Guo, H. Non-Autoregressive Electron Redistribution Modeling for Reaction Prediction. *Proceedings of the International Conference on Machine Learning* **2021**, *139*, 904−913.

(25) Coley, C. W.; Jin, W.; Rogers, L.; Jamison, T. F.; Jaakkola, T. S.; Green, W. H.; Barzilay, R.; Jensen, K. F. A Graph-Convolutional Neural Network Model for the Prediction of Chemical Reactivity. *Chem. Sci.* **2019**, *10*, 370−377.

(26) Sacha, M.; Błaz, M.; Byrski, P.; Dabrowski-Tumanski, P.; Chrominski, M.; Loska, R.; Włodarczyk-Pruszynski, P.; Jastrzebski, S. Molecule Edit Graph Attention Network: Modeling Chemical Reactions as Sequences of Graph Edits. *J. Chem. Inf. Model.* **2021**, *61*, 3273−3284.

(27) Schwaller, P.; Gaudin, T.; Lanyi, D.; Bekas, C.; Laino, T. "Found in Translation": Predicting Outcomes of Complex Organic Chemistry Reactions using Neural Sequence-to-Sequence Models. *Chem. Sci.* **2018**, *9*, 6091−6098.

(28) McGill, C.; Forsuelo, M.; Guan, Y.; Green, W. H. Predicting Infrared Spectra with Message Passing Neural Networks. *J. Chem. Inf. Model.* **2021**, *61*, 2594−2609.

(29) Greenman, K. P.; Green, W. H.; Gómez-Bombarelli, R. Multi-Fidelity Prediction of Molecular Optical Peaks with Deep Learning. *Chem. Sci.* **2022**, *13*, 1152−1162.

(30) Dührkop, K. Deep Kernel Learning Improves Molecular Fingerprint Prediction from Tandem Mass Spectra. *Bioinf.* **2022**, *38*, i342−i349.

(31) Nguyen, D. H.; Nguyen, C. H.; Mamitsuka, H. ADAPTIVE: leArning DAta-dePendenT, concIse molecular VEctors for Fast, Accurate Metabolite Identification from Tandem Mass Spectra. *Bioinf.* **2019**, *35*, i164−i172.

(32) Nguyen, D. H.; Nguyen, C. H.; Mamitsuka, H. Recent Advances and Prospects of Computational Methods for Metabolite Identification: A Review with Emphasis on Machine Learning Approaches. *Brief. Bioinf.* **2019**, *20*, 2028−2043.

(33) Stravs, M. A.; Dührkop, K.; Böcker, S.; Zamboni, N. MSNovelist: De Novo Structure Generation from Mass Spectra. *Nat. Methods* **2022**, *19*, 865−870.

(34) Muratov, E. N.; Bajorath, J.; Sheridan, R. P.; Tetko, I. V.; Filimonov, D.; Poroikov, V.; Oprea, T. I.; Baskin, I. I.; Varnek, A.; Roitberg, A.; et al. QSAR without Borders. *Chem. Soc. Rev.* **2020**, *49*, 3525−3564.

(35) Kearnes, S.; McCloskey, K.; Berndl, M.; Pande, V.; Riley, P. Molecular Graph Convolutions: Moving Beyond Fingerprints. *J. Comput. Aided Mol. Des.* **2016**, *30*, 595−608.

(36) Yang, K.; Swanson, K.; Jin, W.; Coley, C.; Eiden, P.; Gao, H.; Guzman-Perez, A.; Hopper, T.; Kelley, B.; Mathea, M.; Palmer, A.; Settels, V.; Jaakkola, T.; Jensen, K.; Barzilay, R. Analyzing Learned Molecular Representations for Property Prediction. *J. Chem. Inf. Model.* **2019**, *59*, 3370−3388.

(37) Maziarka, Ł.; Danel, T.; Mucha, S.; Rataj, K.; Tabor, J.; Jastrzebski, S. Molecule Attention Transformer. *arXiv Preprint*, **2020**, arXiv:2002.08264.

(38) Kreuzer, D.; Beaini, D.; Hamilton, W.; Létourneau, V.; Tossou, P. Rethinking Graph Transformers with Spectral Attention. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 21618−21629.

(39) Schütt, K. T.; Sauceda, H. E.; Kindermans, P.-J.; Tkatchenko, A.; Müller, K.-R. Schnet−A Deep Learning Architecture for Molecules and Materials. *J. Chem. Phys.* **2018**, *148*, 241722.

(40) Unke, O. T.; Meuwly, M. PhysNet: A Neural Network for Predicting Energies, Forces, Dipole Moments, and Partial Charges. *J. Chem. Theory Comput.* **2019**, *15*, 3678−3693.

(41) Bigi, F.; Pozdnyakov, S. N.; Ceriotti, M. Wigner Kernels: Body-Ordered Equivariant Machine Learning without a Basis. *arXiv Preprint*, **2023**, arXiv:2303.04124.

(42) Winter, B.; Winter, C.; Schilling, J.; Bardow, A. A Smile is All you Need: Predicting Limiting Activity Coefficients from SMILES with Natural Language Processing. *Digital Discovery* **2022**, *1*, 859−869.

(43) Bagal, V.; Aggarwal, R.; Vinod, P.; Priyakumar, U. D. MolGPT: Molecular Generation using a Transformer-Decoder Model. *J. Chem. Inf. Model.* **2022**, *62*, 2064−2076.

(44) Honda, S.; Shi, S.; Ueda, H. R. Smiles Transformer: Pre-trained Molecular Fingerprint for Low Data Drug Discovery. *arXiv Preprint*, **2019**, arXiv:1911.04738.

(45) Chithrananda, S.; Grand, G.; Ramsundar, B. Chemberta: Large-Scale Self-Supervised Pretraining for Molecular Property Prediction. *Machine Learning for Molecules Workshop at NeurIPS*, **2020**, arXiv:2010.09885.

(46) Landrum, G. RDKit: Open-Source Cheminformatics. **2006**. https://www.rdkit.org/.

(47) Mahé, P.; Ueda, N.; Akutsu, T.; Perret, J.-L.; Vert, J.-P. Extensions of marginalized graph kernels. *Proceedings of the International Conference on Machine Learning*, **2004**, 70.

(48) Schweidtmann, A. M.; Rittig, J. G.; Weber, J. M.; Grohe, M.; Dahmen, M.; Leonhard, K.; Mitsos, A. Physical Pooling Functions in Graph Neural Networks for Molecular Property Prediction. *Comput. Chem. Eng.* **2023**, *172*, 108202.

(49) Morgan, H. L. The Generation of a Unique Machine Description for Chemical Structures — A Technique Developed at Chemical Abstracts Service. *J. Chem. Doc.* **1965**, *5*, 107−113.

(50) Kingma, D. P.; Ba, J. Adam: A Method for Stochastic Optimization. *Proceedings of the International Conference on Learning Representations*, **2017**, arXiv:1412.6980.

(51) Lim, M. A.; Yang, S.; Mai, H.; Cheng, A. C. Exploring Deep Learning of Quantum Chemical Properties for Absorption, Distribution, Metabolism, and Excretion Predictions. *J. Chem. Inf. Model.* **2022**, *62*, 6336−6341.

(52) Jin, W.; Stokes, J. M.; Eastman, R. T.; Itkin, Z.; Zakharov, A. V.; Collins, J. J.; Jaakkola, T. S.; Barzilay, R. Deep Learning Identifies Synergistic Drug Combinations for Treating COVID-19. *Proc. Natl. Acad. Sci. U.S.A.* **2021**, *118*, e2105070118.

(53) Liu, G.; Catacutan, D. B.; Rathod, K.; Swanson, K.; Jin, W.; Mohammed, J. C.; Chiappino-Pepe, A.; Syed, S. A.; Fragis, M.; Rachwalski, K.; et al. Deep Learning-Guided Discovery of an Antibiotic Targeting Acinetobacter Baumannii. *Nat. Chem. Biol.* **2023**, *19*, 1342−1350.

(54) Larsson, T.; Vermeire, F.; Verhelst, S. Machine Learning for Fuel Property Predictions: A Multi-Task and Transfer Learning Approach. *SAE Technical Paper* **2023**.

(55) Guan, Y.; Coley, C. W.; Wu, H.; Ranasinghe, D.; Heid, E.; Struble, T. J.; Pattanaik, L.; Green, W. H.; Jensen, K. F. Regio-Selectivity Prediction with a Machine-Learned Reaction Representation and On-the-Fly Quantum Mechanical Descriptors. *Chem. Sci.* **2021**, *12*, 2198−2208.

(56) Biswas, S.; Chung, Y.; Ramirez, J.; Wu, H.; Green, W. H. Predicting Critical Properties and Acentric Factor of Fluids using Multi-Task Machine Learning. *J. Chem. Inf. Model.* **2023**, *63*, 4574−4588.

(57) Lenselink, E. B.; Stouten, P. F. W. Multitask machine learning models for predicting lipophilicity (logP) in the SAMPL7 challenge. *Journal of Computer-Aided Molecular Design* **2021**, *35*, 901−909.

(58) McNaughton, A. D.; Joshi, R. P.; Knutson, C. R.; Fnu, A.; Luebke, K. J.; Malerich, J. P.; Madrid, P. B.; Kumar, N. Machine Learning Models for Predicting Molecular UV−Vis Spectra with Quantum Mechanical Properties. *J. Chem. Inf. Model.* **2023**, *63*, 1462−1471.

(59) Heid, E.; Green, W. H. Machine Learning of Reaction Properties via Learned Representations of the Condensed Graph of Reaction. *J. Chem. Inf. Model.* **2022**, *62*, 2101−2110.

(60) Isert, C.; Kromann, J. C.; Stiefl, N.; Schneider, G.; Lewis, R. A. Machine Learning for Fast, Quantum Mechanics-Based Approximation of Drug Lipophilicity. *ACS Omega* **2023**, *8*, 2046−2056.

(61) Spiekermann, K. A.; Pattanaik, L.; Green, W. H. Fast Predictions of Reaction Barrier Heights: Toward Coupled-Cluster Accuracy. *J. Phys. Chem. A* **2022**, *126*, 3976−3986.

(62) Chung, Y.; Green, W. H. Machine Learning from Quantum Chemistry to Predict Experimental Solvent Effects on Reaction Rates. *ChemRxiv Preprint*, **2023**.

(63) Lansford, J. L.; Jensen, K. F.; Barnes, B. C. Physics-informed Transfer Learning for Out-of-sample Vapor Pressure Predictions. *Propellants, Explosives, Pyrotechnics* **2023**, *48*, e202200265.

(64) Chung, Y.; Vermeire, F. H.; Wu, H.; Walker, P. J.; Abraham, M. H.; Green, W. H. Group Contribution and Machine Learning Approaches to Predict Abraham Solute Parameters, Solvation Free Energy, and Solvation Enthalpy. *J. Chem. Inf. Model.* **2022**, *62*, 433−446.

(65) Koscher, B. A.; Canty, R. B.; McDonald, M. A.; Greenman, K. P.; McGill, C. J.; Bilodeau, C. L.; Jin, W.; Wu, H.; Vermeire, F. H.; Jin, B.; Hart, T.; Kulesza, T.; Li, S.-C.; Jaakkola, T. S.; Barzilay, R.; Gomez-Bombarelli, R.; Green, W. H.; Jensen, K. F. Autonomous, Multiproperty-Driven Molecular Discovery: From Predictions to Measurements and Back. *Science* **2023**, *382* (6677), eadi1407.

(66) Vermeire, F. H.; Chung, Y.; Green, W. H. Predicting Solubility Limits of Organic Solutes for a Wide Range of Solvents and Temperatures. *J. Am. Chem. Soc.* **2022**, *144*, 10785−10797.

(67) Wong, F.; Omori, S.; Donghia, N. M.; Zheng, E. J.; Collins, J. J. Discovering Small-Molecule Senolytics with Deep Neural Networks. *Nature Aging* **2023**, *3*, 734−750.

(68) Felton, K. C.; Ben-Safar, H.; Alexei, A. DeepGamma: A Deep Learning Model for Activity Coefficient Prediction. *1st Annual AAAI Workshop on AI to Accelerate Science and Engineering (AI2ASE)*, **2022**.

(69) Colomba, M.; Benedetti, S.; Fraternale, D.; Guidarelli, A.; Coppari, S.; Freschi, V.; Crinelli, R.; Kass, G. E. N.; Gorassini, A.; Verardo, G.; Roselli, C.; Meli, M. A.; Di Giacomo, B.; Albertini, M. C. Nrf2-Mediated Pathway Activated by Prunus spinosa L. (Rosaceae) Fruit Extract: Bioinformatics Analyses and Experimental Validation. *Nutrients* **2023**, *15*, 2132.

(70) Chang, C.-I. An Information-Theoretic Approach to Spectral Variability, Similarity, and Discrimination for Hyperspectral Image Analysis. *IEEE Transactions on Information Theory* **2000**, *46*, 1927−1932.

(71) Nix, D. A.; Weigend, A. S. Estimating the Mean and Variance of the Target Probability Distribution. *Proceedings of IEEE International Conference on Neural Networks* **1994**, *1*, 55−60.

(72) Amini, A.; Schwarting, W.; Soleimany, A.; Rus, D. Deep Evidential Regression. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 14927−14937.

(73) Sensoy, M.; Kaplan, L.; Kandemir, M. Evidential Deep Learning to Quantify Classification Uncertainty. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, na.

(74) Villani, C. *Optimal Transport: Old and New*; Springer, **2009**; Vol. *338*.

(75) Bergstra, J.; Bardenet, R.; Bengio, Y.; Kégl, B. Algorithms for Hyper-Parameter Optimization. *Adv. Neural Inf. Process. Syst.* **2011**, *24*, na.

(76) Bergstra, J.; Yamins, D.; Cox, D. Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. *Proceedings of the International Conference on Machine Learning* **2013**, 115−123.

(77) Lakshminarayanan, B.; Pritzel, A.; Blundell, C. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. *Adv. Neural Inf. Process. Syst* **2017**, *30*, na.

(78) Gal, Y.; Ghahramani, Z. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. *Proceedings of the International Conference on Machine Learning* **2016**, *48*, 1050−1059.

(79) Levi, D.; Gispan, L.; Giladi, N.; Fetaya, E. Evaluating and calibrating uncertainty prediction in regression tasks. *Sensors* **2022**, *22*, 5540.

(80) Zelikman, E.; Healy, C.; Zhou, S.; Avati, A. CRUDE: Calibrating Regression Uncertainty Distributions Empirically. *arXiv Preprint*, **2020**, arXiv:2005.12496.

(81) Wang, D.; Yu, J.; Chen, L.; Li, X.; Jiang, H.; Chen, K.; Zheng, M.; Luo, X. A Hybrid Framework for Improving Uncertainty Quantification in Deep Learning-Based QSAR Regression Modeling. *J. Cheminf.* **2021**, *13*, 1−17.

(82) Guo, C.; Pleiss, G.; Sun, Y.; Weinberger, K. Q. On Calibration of Modern Neural Networks. *Proceedings of the International Conference on Machine Learning* **2017**, 1321−1330.

(83) Zadrozny, B.; Elkan, C. Transforming Classifier Scores into Accurate Multiclass Probability Estimates. *Proceedings of the International Conference on Knowledge Discovery and Data Mining* **2002**, 694−699.

(84) Scalia, G.; Grambow, C. A.; Pernici, B.; Li, Y.-P.; Green, W. H. Evaluating Scalable Uncertainty Estimation Methods for Deep Learning-Based Molecular Property Prediction. *J. Chem. Inf. Model.* **2020**, *60*, 2697−2717.

(85) Chemprop. https://chemprop.readthedocs.io/en/latest/ (accessed April 6 2023).

(86) Chemprop Workshop. https://www.youtube.com/watch?v=TeOl5E8Wo2M (accessed April 6 2023).

(87) Reuther, A.; Kepner, J.; Byun, C.; Samsi, S.; Arcand, W.; Bestor, D.; Bergeron, B.; Gadepally, V.; Houle, M.; Hubbell, M.; et al. Interactive Supercomputing on 40,000 Cores for Machine Learning and Data Analysis. *Proceedings of the IEEE High Performance Extreme Computing Conference* **2018**, 1−6.