Supporting Information

Evidential Deep Learning for Guided Molecular Property Prediction and Discovery

Ava P. Soleimany,[†,‡,¶,@] Alexander Amini,[§,@] Samuel Goldman,[‖,@] Daniela Rus,[§] Sangeeta N. Bhatia,[†,§,⊥] and Connor W. Coley[*,#]

†*Harvard-MIT Division of Health Sciences and Technology, MIT, Cambridge, MA*

‡*Graduate Program in Biophysics, Harvard University, Boston, MA*

¶*Microsoft Research New England, Cambridge, MA*

§*Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA*

‖*Computational and Systems Biology, MIT, Cambridge, MA*

⊥*Howard Hughes Medical Institute, Cambridge, MA*

#*Department of Chemical Engineering, MIT, Cambridge, MA*

@*Equal contribution.*

*E-mail: ccoley@mit.edu

Supporting Information (33 pages)

Code availability; Methods; Table S1: Extended uncertainty benchmarking results; Table S2: Statistical significance tests for Bayesian optimization; Figure S1: Uncertainty benchmarking and calibration for lower-N datasets; Figure S2: Uncertainty benchmarking and calibration for lower-N Therapeutic Data Commons datasets; Figure S3: Spearman rank correlation between error and uncertainty; Figure S4: Task-specific cutoffs for QM9 dataset; Figure S5: Uncertainty benchmarking and calibration for higher-N 2D and 3D datasets; Figure S6: Effect of λ on uncertainty calibration; Figure S7: Cutoff RMSE is robust to small λ shifts; Figure S8: Task-specific calibration for QM9 dataset; Figure S9: Antibiotic discovery datasets and uncertainty predictions.

Code availability

All code to reproduce experiments and results can be found at `https://github.com/aamini/chemprop`.

Methods

Evidential deep learning formulations

Evidential deep learning approaches seek to directly learn prediction uncertainties by formulating learning as an evidence acquisition process.[26,27] This is achieved by training models to infer the parameters of a higher-order *evidential* distribution that models the evidence behind individual predictions. That is, individual observations of training examples lend support to this higher-order distribution, such that the predictions of the neural network learner are represented as a distribution over the prediction likelihood function itself. Estimates of uncertainty are then formulated using the parameters of the learned evidential distribution and thus can be obtained directly from a single forward pass through the model.

Evidential learning is achieved through two modifications to a standard forward prediction model. First, the network's output layer is modified to output the parameters of the evidential distribution, rather than a point estimate of a target label. Second, the resultant model is trained with a specific loss function that jointly maximizes the model's fit to the data and also minimizes its evidence on errors, i.e., increases uncertainty when predictions should not be trusted.

Evidential learning for regression

In regression, we are given a dataset of paired training examples $\mathcal{D} = \{x_i, y_i\}_{i=1}^{N}$ where the targets are assumed to be drawn i.i.d. from a Gaussian distribution with unknown mean and variance $\theta = \{\mu, \sigma^2\}$. We seek to probabilistically estimate the mean and variance assuming that the mean is drawn from a Gaussian and the variance is drawn from an Inverse-Gamma distribution. The joint higher-order, evidential distribution is thus represented as a Normal-Inverse-Gamma. Specifically, the Normal-Inverse-Gamma distribution, $p(\theta|\mathbf{m})$, which is also the conjugate prior to the Gaussian,[49] is parametrized by $\mathbf{m} = \{\gamma, \upsilon, \alpha, \beta\}$ and represents a distribution over $\theta = \{\mu, \sigma^2\}$. Therefore, in this work, the final layers of evidential D-MPNN and atomistic networks were modified to output these Normal-Inverse-Gamma hyperparameters. Thus, the network has 4 outputs for every target task. Given a Normal-Inverse-Gamma distribution, the prediction and uncertainty are formulated according to the distribution moments:

$$\underbrace{\mathbb{E}[\mu] = \gamma,}_{\text{prediction}} \qquad \underbrace{\mathrm{Var}[\mu] = \frac{\beta}{\upsilon(\alpha - 1)}}_{\text{uncertainty}}.$$

Evidential models are trained using a dual-objective loss $\mathcal{L}(x)$ that consists of two loss terms, to both maximize model fit according to the negative log-likelihood and regularize evidence on errors:

$$\mathcal{L}(x) = \mathcal{L}^{\mathrm{NLL}}(x) + \lambda \mathcal{L}^{R}(x)$$

where $\mathcal{L}^{\mathrm{NLL}}(x)$ is the negative log-likelihood and $\mathcal{L}^{R}(x)$ is an evidence regularizer.[27] The regularization coefficient λ controls the strength of uncertainty inflation relative to model fit. All evidential models were trained according to this loss function, with λ values specified in the figure captions and corresponding Methods sections. We refer to the work of Amini et al. for more details on the evidential regression formulation.[27] We also note that the evidential method has been demonstrated in the context of discrete, multiclass classification problems,[26] despite the focus of this work (along with relevant prior literature in molecular property prediction) being on continuous regression tasks.

Network architectures

To show its broad applicability in molecular property prediction, evidential regression was integrated into networks operating on 2D molecular graphs and 3D conformers – directed message passing neural network (D-MPNN) and atomistic neural network models, respectively.

Directed message passing neural networks

To investigate performance on 2D molecular graphs, evidential methods were integrated into a state-of-the-art D-MPNN model.[50] The D-MPNN architecture is a variant of a message passing neural network (MPNN). MPNNs operate on molecular graphs to first learn an encoded molecular representation (i.e., molecule-level feature vector) by passing "messages" between atoms and/or bonds and their direct neighbors. These messages build up a hidden state for each atom and/or bond, and repeated message passing iterations yield a molecule-level feature vector. A feed-forward network operating on this feature is used to produce a task-specific representation of an input molecule.

D-MPNN models were implemented in PyTorch[47] within the Chemprop library.[50] D-MPNNs were implemented using standard settings: messages passed on directed bonds, messages subjected to ReLU activation, a learned hidden dimension of 300, 3 layers, no dropout, and the output of the message-passing phase fully connected to the output layer. For evidential regression models, the final output layer was modified to infer a single evidential distribution for each task, with each task parametrized by four outputs (e.g., prediction of 12 tasks uses 48 outputs). Models were trained using the Adam optimization algorithm. Target values were normalized with a standard scaler for training. For evaluations, model state was reloaded from the epoch with the lowest validation score after training was completed.

Atomistic neural networks

To investigate performance on 3D conformer representations, evidential regression was integrated into the end-to-end atomistic neural network SchNet.[44] Rather than operating only on the 2D graph, SchNet instead builds internal representations of a molecule using 3D coordinate positions of each atom as inputs to the model. SchNet employs "continuous filters", where information is shared between molecules not based upon discrete edges but rather continuous spatial distance between molecules (Fig. 1). This model architecture has proved beneficial for predicting energies and forces, as it is rotationally invariant and equivariant.[44] Similar to the message passing networks, SchNet alternates between transforming atom-wise representations individually and integrating interaction information. At the end of these internal hidden layers, information at each atom is aggregated through summation to produce a fixed dimension hidden representation, upon which a single feed forward layer is applied to produce an output value. In the case of energy predictions, SchNet predicts energy at each atom separately and adds all energies to predict the energy for the molecule. We refer the reader to the work of Schutt et al. for more details.[44,51]

The final layer of SchNet was modified to infer a single evidential distribution over the task of predicting U_0 for the QM9 dataset, outputting four values rather than one, corresponding to the parameters of the higher-order evidential distribution.

Unlike in the Chemprop library, where normalization is conducted prior to any training, the SchNet model uses a customized scaler that normalizes both according to the target values *and* a precomputed "atomref" value. Instead of transforming the target values and training the model with modified target values, SchNet adds these normalization numbers to outputs for each atom separately as part of its final layer by default. SchNet builds up final energy values for each atom m, $G(x_i)_m$, and the final prediction for molecule x_i is computed according to:

$$\hat{y}_i = \sum_{m \in \text{Atoms}(x_i)} [\sigma_y G(x_i)_m + \text{atomrefs}(x_{i,m}) + \mu_y]$$

where

$$\mu_y = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{|x_i|} \Big(y_i - \sum_{m \in \text{Atoms}(x_i)} \text{atomrefs}(x_{i,m}) \Big)$$

$$\sigma_y = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(\mu_y - \frac{1}{|x_i|} \Big(y_i - \sum_{m \in \text{Atoms}(x_i)} \text{atomrefs}(x_{i,m}) \Big) \right)^2}$$

Here, SchNet tries to learn residual values at each atom. Because the evidential regression method is most easily adapted for predicting and training on data with standard normalization procedures, a custom standardization method was implemented to incorporate atomref values, which we found to be empirically helpful. Specifically, the value U_0 for each input molecule was adjusted before inference according to:

$$y_i' = \frac{y - \mu_y - \sum_{m \in \text{Atoms}(x_i)} \text{atomrefs}(x_{i,m})}{\sigma_y}$$

At inference time, the inverse of this scaler was computed to predict $\hat{y}$. Accounting for such discrepancies in the scaling function could provide opportunities for future development of atomistic neural network architectures.

Datasets

Lower-N 2D datasets

The lower-N 2D datasets used in this study were extracted from MoleculeNet[29] and used as prepared by Wang et al.[50] For regression tasks (Table 1), datasets of aqueous solubility (Delaney), solvation energy (freesolv), lipophilicity (lipo), and atomization energy (QM7) were evaluated. The datasets were split randomly using a 80/10/10 split for training/validation/testing. SMILES strings were used as input to D-MPNN models.

Lower-N 2D TDC Datasets

In addition to the MoleculeNet lower-N datasets, evidential regression was additionally evaluated on three datasets from the Therapeutics Data Commons[30] (TDC). For regression tasks (Table S1), datasets of heaptocyte clearance (“clearance”),[52] plasma protein binding rates (“PPBR”),[52] and the lethal dosage of drugs (“LD50”)[53] were used. The datasets were extracted from the TDC and split randomly using a 80/10/10 split for training/validation/testing. SMILES strings were used as input to D-MPNN models.

Higher-N 2D datasets

For the 2D setting, the QM9[32,54] dataset with SMILES strings input was extracted from MoleculeNet.[29] For both benchmarking and active learning D-MPNN experiments using this dataset, all 12 output tasks of the QM9 dataset, which reflect computer-generated quantum mechanical properties,[29,32] were predicted. A ligand docking dataset based on Enamine’s Diversity Collection of 50,240 molecules was used as a second higher-N 2D dataset. Target values consisted of docking scores of compounds against thymidylate kinase (PDB ID: 4UNN31) using AutoDock Vina.[35] This dataset was used as prepared by Graff et al.[34]

Higher-N 3D (atomistic) datasets

For the 3D setting, a variation of the QM9 dataset was utilized wherein atomic coordinates, not molecular graphs generated from SMILES strings, were used as inputs. Atomic coordinates correspond to the coordinates of molecular conformers in 3D space. In this setting, the single task of total formation energy at 0K, U_0,[51] was predicted for a given molecule input.

Antibiotic discovery datasets

For virtual screening experiments for antibiotic discovery, D-MPNN models were trained on a dataset of $2,335$ small molecules and their *in vitro* growth inhibitory activity against *Escherichia coli*, as generated and reported by Stokes et al.[39] In this dataset, growth inhibitory activity is reported as endpoint OD_{600}, where lower OD_{600} values correspond to stronger growth inhibitory activity, and models were trained to predict this as a continuous target (i.e., formulated as a regression problem). The Broad Drug Repurposing Hub[40] was used as a discovery dataset, as prepared by Stokes et al.[39] Model predictions were compared to empirically determined growth inhibitory activity against *E. coli* for a subset of molecules from the Broad Drug Repurposing Hub, as measured by Stokes et al.[39]

Uncertainty quantification baselines

Evaluations are focused on regression tasks, defined by a dataset $\mathcal{D}$ containing data points (x_i, y_i) where $y_i \in \mathbb{R}$ is a scalar-valued target property and x_i is a molecule representation, represented as either a SMILES string or a set of coordinates for the 2D or 3D settings, respectively.

For baselines, we use gold-standard epistemic uncertainty quantification methods that rely on sampling, i.e., creating a set of predictions that together constitute an ensemble from which estimates of predictive variance can be obtained. Specifically, a set of predictions $\mathcal{E} = \{G_1(x), G_2(x), \cdots, G_n(x)\}$, where each $G_i(x)$ is an inference sample, is obtained such that the individual samples (e.g., individual model predictions) can yield a final prediction defined by:

$$\hat{G}(x) = \sum_{G \in \mathcal{E}} \frac{G(x)}{n}.$$

As previously proposed,[13,14] from the multiple samples obtained from this set of models, the uncertainty

$U(x)$ is defined as the variance across predictions:

$$U(x) = \sum_{G \in \mathcal{E}} \frac{(\hat{G}(x) - G(x))^2}{n}.$$

Traditional model ensembling

For the ensemble baseline, distinct models $G_i \in \mathcal{E}$ were trained on different splits of the same training data and initialized with different sets of randomly selected weights, as previously proposed.[13] Greater variance among model outputs reflects greater uncertainties, due to the fact that for out-of-distribution regions not well represented in the training data, each ensemble member will be more significantly affected by its initialization, ultimately resulting in more variable predictions. The computational cost of this approach scales linearly with the size of the ensemble and clearly exceeds the cost of training a single model. All evaluations utilized an ensemble size of 5.

Monte Carlo dropout sampling

For the dropout baseline, a single model G is trained with dropout, in which individual network weights are randomly set to zero at every training step with probability p, also known as the “dropout rate”. At inference time, a set of predictions $\mathcal{E}$ is obtained for an input x_i by application of randomly-generated dropout masks to a trained model G. This strategy approximates Bayesian inference and can thus be used to obtain prediction samples from which uncertainty can be estimated.[14] All evaluations utilized a dropout rate of 0.2 and a set size of 5.

Uncertainty benchmarking experiments

Each dataset was randomly partitioned using an 80/10/10 split for training, validation, and testing set splits respectively. For D-MPNN experiments, test set target values were normalized using statistics from the train set such that the train set target values have mean 0 and standard deviation 1 in each case (atomistic normalization previously described separately).

For the D-MPNN models, each model was trained for 100 epochs using the Adam optimizer with default parameters, and the best model was selected based upon validation loss. We use the default Chemprop architecture and training procedure parameters as set by Yang et al.,[50] including a Noam learning rate scheduler with final learning rate of 10^{-4}, batch size of 50, hidden size of 300, depth of 3, and ReLU activations.

For the atomistic neural network models, the default SchNet architecture parameters and preparations of the U_0 calculations from the QM9 dataset were used. For training, the same training procedure as in Chemprop (optimizer, learning rate, and learning rate scheduler) was used, with the learning rate in the Noam scheduler modified to 2×10^{-4}, rather than 1×10^{-4}, for greater stability in early model epochs.

Test set predictions and corresponding uncertainties were generated in each experiment for downstream analysis.

Error vs. confidence cutoffs

Test set predictions from each model run were sorted by uncertainty, such that r_i represents the index of the test set molecule, x_{r_i} that has the i^{th} highest predictive uncertainty (e.g., x_{r_1} is predicted with highest uncertainty and x_{r_n} with most confidence where n is the total number of molecules evaluated). For every value of i, we compute the error for the set of all predicted test set molecules of $\{x_{r_j} : j \geq i\}$. We compute cutoff mean average error (MAE) and root mean squared error (RMSE) for different cutoff values

i, corresponding to different confidence cutoffs (e.g., 50% confidence cutoff can be computed by setting $i = \lceil 0.5n \rceil$):

$$\mathrm{MAE}_i = \frac{1}{n-i} \sum_{j \geq i}^{n} |y_{r_j} - G(x_{r_j})|$$

$$\mathrm{RMSE}_i = \sqrt{\frac{1}{n-i} \sum_{j \geq i}^{n} (y_{r_j} - G(x_{r_j}))^2}$$

Confidence cutoff errors were computed at even intervals of 30, exclusive, for all datasets except for the lower-N datasets, in which case cutoff errors for *all* values of i were computed.

This procedure was repeated for different random starts of the model and random training/validation/testing splits, corresponding to independent experimental trials ($n = 10$ for lower-N data, $n = 5$ for all else). Cutoff values at each confidence percentile were computed separately for each random initialization in order to estimate standard deviations over multiple trials at each confidence percentile.

In the multitask setting, cutoff errors were computed separately across each task to avoid avoid artifacts due to averaging uncertainties across tasks. Thus, for $n = 5$ trials on the QM9 dataset, cutoff errors were computed separately for each of the $d = 12$ tasks in each of the n trials, leading to $n * d = 60$ computed cutoff error values at each confidence percentile. Performance on separate tasks in QM9 is shown in Fig. S4.

Spearman's rank correlation coefficient

We desire that, on average, predictions for molecules made with higher certainty should also be more accurate. Therefore, Spearman's rank correlation coefficient was used as an additional metric to assess the ability of models to rank errors, and was computed following the procedure outlined in Hirschfeld et al.[15]

Spearman's rank correlation coefficient is defined by creating two vectors L_1 and L_2 and corresponding rank vectors r_{L_1} and r_{L_2} that sort the dataset in ascending order. The correlation measures the agreement between these two ranking lists:

$$\rho(L_1, L_2) = \frac{\mathrm{cov}(r_{L_1}, r_{L_2})}{\sigma(r_{L_1})\sigma(r_{L_2})}$$

If these two lists are perfectly correlated, then $\rho = 1$, whereas if they are perfectly inversely correlated, $\rho = -1$. We desire an uncertainty estimation method where ρ between error and predictive uncertainty is highest. Spearman's rank correlation coefficients were computed using the scipy.stats module.[55]

Calibration analysis & miscalibration area

In the regression case, the empirical probability of observing the true target values y_i around the predicted values $\hat{y}_i$ should match the posterior predictive probability distribution $p(\hat{y}_i)$ defined by the uncertainty quantification method for a well-calibrated model.[36] That is, if we create 50% credible intervals around each predicted point value $\hat{y}_i$, the true value y_i should fall within that credible interval 50% of the time. To evaluate this, we assess uncertainty calibration following the procedure outlined in Tran et al.[11]

We assume the posterior predictive distributions to be Gaussian around the predicted point value $\hat{y}_i$. We find the lower and upper bound values between which we expect to observe a fraction e of the true values. For each data point x_i, given an inverse CDF function for the predictive distribution, F_i^{-1}, we define lower bound, Lb, and upper bound, Ub, values for each predicted point in the test set:

$$\mathrm{Lb}_i = F_i^{-1}(0.5 - e)$$

$$\mathrm{Ub}_i = F_i^{-1}(0.5 + e)$$

We count the fraction of true predictive values where $\mathrm{Lb}_i < y_i < \mathrm{Ub}_i$, which we denote as the "estimated confidence". We repeat this value at various expected probability values, e, to create the calibration plots which show the expected proportion correct e against the estimated confidence, i.e. the estimated cumulative probability.

To quantify the degree of calibration, we measure each model's deviation from ideal calibration by computing the area away from the parity line in which the estimated confidence and observed proportion correct are matched. This integration was computed using the scipy.stats.simps function.[55]

For the lower-N datasets, the effect of the evidential regularizer strength was determined by varying the regularization parameter λ. Individual evidential D-MPNNs were trained with varying regularization coefficients. Lower λ leads to overconfident predictors, whereas higher λ leads to underconfident predictions, as they are penalized more for attributing higher evidence to erroneous predictions.

Active learning on QM9 dataset

Experiments were conducted on the QM9 dataset with the objective of improving the predictive accuracy of the model in terms of its learning efficiency. In other words, the objective of these experiments was to select a training dataset intelligently such that higher predictive accuracy could be achieved with less data. Indeed, because data in the chemical sciences can often be limited and expensive to acquire, active learning can yield powerful predictive models that require minimal data generation.

The general experimental set up consists of iterative rounds of model training, data acquisition, and model assessment. All active learning experiments were conducted in the 2D setting using D-MPNN models. Briefly, models were initially trained on a randomly selected subset of data from QM9, and performance was assessed on a held-out test set and quantified via RMSE. At each iteration, m new data samples were selected and added to the training set on the basis of an acquisition function α. Models were then re-trained from scratch, and performance was again assessed on the held-out test set. This process was then repeated until the entire training dataset (consisting of 80% of all of QM9) was acquired. All experiments were conducted with $n = 10$ independent trials.

Two general acquisition strategies were tested in this study: first, an explorative strategy in which the algorithm chooses to acquire instances about which it is most uncertain, and second, a baseline strategy in which new data instances are acquired at random. These strategies are labeled as “Explorative” and “Random”, respectively, in Fig. 5. Evidential deep learning, model ensembling, and dropout sampling were used as the UQ methods for explorative selection. In all three instances, the value of the acquisition function for a particular point x is given by the estimated uncertainty: $\alpha(x) = \hat{\sigma}(x)$. For each method, the m samples with the greatest uncertainties were acquired at each iteration. Explorative acquisition was compared to each method’s respective random acquisition baseline, given differences in training and model output for each UQ approach.

Bayesian optimization on docking dataset

Bayesian optimization is an active learning approach that seeks to optimize an objective function by iteratively selecting experiments to perform according to a model’s predictions. In this work, Bayesian optimization was performed on a set of candidate molecules with the objective of selecting molecules that optimize a target property $f(x)$, in this case the ligand docking score against thymidylate kinase from AutoDock Vina. We follow the general procedure outlined by Graff et al.[34] The objective function $f(x)$ is calculated for n randomly-selected molecules $\{x\}_{i=1}^{n}$, yielding a dataset $\mathcal{D} = \{(x_i, f(x_i))\}_{i=1}^{n}$. A D-MPNN model is then trained on these data, and predictions $\hat{f}(x)$ are passed to an acquisition function α which describes the utility of acquiring a new point. A set of m new points are subsequently selected according to the acquisition function; the objective function for each of these points is calculated; and these points are used to grow the dataset $\mathcal{D}$. This process was repeated iteratively for a fixed number of iterations. We refer the reader to recent works for more details on the Bayesian optimization procedure.[34,37]

The following acquisition functions were tested in this study:

$$\text{Random}(x) \sim \mathcal{U}(0,1); \qquad \text{Greedy}(x) = \hat{\mu}(x); \qquad \text{UCB}(x) = \hat{\mu}(x) + \beta\hat{\sigma}(x).$$

Here $\hat{\mu}(x)$ and $\hat{\sigma}(x)$ are the model's predicted mean and uncertainty at point x, respectively. $\beta = 2$ for all experiments reported in the paper.

Search trajectory performance was evaluated by the fraction of top-k scores identified during Bayesian optimization, calculated as the size of the intersection of the list of true top-k scores and the list of top-k scores found, then divided by k. Acquisition sample diversity was measured as the average 10-nearest training set neighbors (10-NN) Tanimoto distance for batch samples after the first round of acquisition in Bayesian optimization.

Uncertainty-guided virtual screening for antibiotic discovery

Evidential D-MPNNs were trained to predict a molecule's growth inhibitory effect on *E. coli*, following the general virtual screening pipeline presented by Stokes et al.[39] Growth inhibitory activity was measured as *in vitro* OD_{600} of *E. coli* following incubation with a compound, where lower values correspond to more potent inhibition, and to estimate the uncertainty associated with that prediction. The evidential D-MPNN model was trained following the procedure outlined by Stokes et al.,[39] with the notable exception that the prediction task was formulated as a regression problem. Briefly, the molecular representation learned by the D-MPNN was augmented with 200 additional molecule-level features computed in RDKit.[48] Models were trained on the primary dataset of the OD_{600} (target values y) of *E. coli* following incubation with each of $2,335$ small molecules (input values x in SMILES representation) using a $80/20$ training/validation split. Models were trained for 30 epochs with five-fold cross validation and a regularization coefficient $\lambda = 0.1$. Due to the imbalanced distribution of OD_{600} values in this dataset (Fig. S9A), the training dataset was rebalanced by sampling molecules with $\mathrm{OD}_{600} > 0.2$ with probability 0.1.

The trained evidential D-MPNN was applied to the Broad Drug Repurposing Hub[40] to predict OD_{600} values and uncertainties for molecules in this discovery dataset. t-SNE analysis was conducted using scikit-learn's implementation of t-Distributed Stochastic Neighbor Embedding. Morgan fingerprints for each molecule using a radius of 2 and 2048-bit fingerprint vectors were first computed in RDKit, and t-SNE with Tanimoto (Jaccard) distance metric and default parameters was then used to reduce the data from 2048 dimensions to two dimensions. The distance between points in the t-SNE plots thus reflects the Tanimoto distance of the corresponding molecules.

For the uncertainty-guided virtual screen, molecules from the Broad discovery dataset were first ranked based on predicted OD_{600} values (lower is better), and the top 50 ranking molecules were downselected. Confidence percentiles across this set were computed based on uncertainty estimates returned by the evidential D-MPNN, and the set of 50 molecules was subsequently filtered according to varying confidence percentiles. Specifically, for a given confidence threshold p, molecules with estimated confidences below the associated p^{th} percentile are removed from the list of top 50 molecules, with p ranging from the 50^{th} to 100^{th} percentiles of greatest predictive confidence. The experimental hit rate for both the initial set of 50 molecules (i.e., no confidence filtering) and each filtered set was determined using empirically determined OD_{600} values reported in Stokes et al.[39] The hit rate was defined as the proportion of molecules in each candidate set with $\mathrm{OD}_{600} < 0.2$ (as previously reported[39]) relative to the total number of molecules in that candidate set. This screening procedure was also used for the dropout and ensemble-based baseline methods, following training and testing on the Stokes' training and Broad discovery datasets, respectively.

Additional Results

Table S1: **Extended model error at various confidence percentile cutoffs including TDC lower-N data.** For a given confidence percentile cutoff, top performing methods based on prediction standard error of the mean (± s.e.m.) are bolded. A cutoff of 0.95 indicates that only the top 5% most confident predictions are considered. Full confidence plots for all datasets are shown in Fig. 3 and Figs. S1, S4, S5. Mean ± s.e.m. (RMSE for all D-MPNN models, MAE for atomistic); $n = 10$ independent trials for lower-N datasets, $n = 5$ independent trials for higher-N datasets.

	Delaney			**Freesolv**			**Lipo**			**QM7** ($\times 10^2$)		
Cutoff	**Dropout**	**Ensemble**	**Evidence**	**Dropout**	**Ensemble**	**Evidence**	**Dropout**	**Ensemble**	**Evidence**	**Dropout**	**Ensemble**	**Evidence**
0.0	**0.68 ± 0.02**	**0.65 ± 0.03**	**0.66 ± 0.02**	**1.00 ± 0.06**	**0.94 ± 0.06**	**0.96 ± 0.07**	**0.55 ± 0.01**	**0.53 ± 0.02**	**0.55 ± 0.02**	1.18 ± 0.02	**1.12 ± 0.02**	**1.15 ± 0.03**
0.5	0.62 ± 0.03	0.55 ± 0.03	**0.44 ± 0.01**	0.79 ± 0.07	**0.45 ± 0.04**	**0.42 ± 0.04**	0.52 ± 0.01	**0.40 ± 0.01**	0.50 ± 0.01	0.88 ± 0.06	0.88 ± 0.06	**0.39 ± 0.03**
0.75	0.59 ± 0.03	0.50 ± 0.05	**0.35 ± 0.02**	0.85 ± 0.12	**0.41 ± 0.05**	**0.36 ± 0.04**	0.50 ± 0.02	**0.38 ± 0.02**	0.51 ± 0.02	0.65 ± 0.03	0.81 ± 0.06	**0.23 ± 0.04**
0.90	0.55 ± 0.03	0.51 ± 0.09	**0.28 ± 0.02**	0.66 ± 0.20	**0.40 ± 0.06**	**0.35 ± 0.08**	0.46 ± 0.03	**0.38 ± 0.02**	0.53 ± 0.03	0.69 ± 0.05	0.71 ± 0.11	**0.10 ± 0.04**
0.95	0.53 ± 0.06	0.45 ± 0.06	**0.22 ± 0.02**	0.75 ± 0.30	**0.27 ± 0.04**	**0.38 ± 0.12**	0.49 ± 0.04	**0.36 ± 0.03**	0.50 ± 0.04	0.73 ± 0.08	0.69 ± 0.11	**0.10 ± 0.04**

	Enamine D-MPNN			**QM9 D-MPNN**			**QM9 Atomistic** ($\times 10^{-2}$)	
Cutoff	**Dropout**	**Ensemble**	**Evidence**	**Dropout**	**Ensemble**	**Evidence**	**Ensemble**	**Evidence**
0.0	3.40 ± 0.12	**4.47 ± 0.18**	5.60 ± 0.20	0.35 ± 0.00	**0.33 ± 0.00**	0.35 ± 0.00	**2.04 ± 0.03**	2.98 ± 0.08
0.5	3.64 ± 0.05	2.12 ± 0.02	**1.55 ± 0.12**	0.33 ± 0.00	0.32 ± 0.00	**0.30 ± 0.00**	**1.45 ± 0.02**	1.52 ± 0.02
0.75	3.42 ± 0.04	1.94 ± 0.04	**1.04 ± 0.13**	0.33 ± 0.00	0.32 ± 0.00	**0.28 ± 0.00**	1.36 ± 0.02	**1.33 ± 0.02**
0.90	3.30 ± 0.06	1.80 ± 0.03	**0.63 ± 0.12**	0.32 ± 0.00	0.32 ± 0.01	**0.27 ± 0.00**	1.31 ± 0.03	**1.18 ± 0.03**
0.95	3.26 ± 0.05	1.79 ± 0.05	**0.42 ± 0.01**	0.33 ± 0.01	0.32 ± 0.01	**0.26 ± 0.01**	1.29 ± 0.03	**1.12 ± 0.03**

	Clearance			**LD50**			**PPBR**		
Cutoff	**Dropout**	**Ensemble**	**Evidence**	**Dropout**	**Ensemble**	**Evidence**	**Dropout**	**Ensemble**	**Evidence**
0.0	47.45 ± 0.90	**44.05 ± 0.68**	5.60 ± 0.20	0.58 ± 0.01	**0.56 ± 0.05**	0.56 ± 0.01	**11.67 ± 0.42**	**11.17 ± 0.40**	**11.35 ± 0.25**
0.5	**33.20 ± 2.12**	38.81 ± 1.60	37.68 ± 2.35	0.50 ± 0.01	0**0.47 ± 0.01**	0.49 ± 0.01	**7.57 ± 0.68**	**6.42 ± 0.62**	**6.83 ± 0.59**
0.75	**30.33 ± 2.94**	37.23 ± 2.94	**32.02 ± 3.60**	0.48 ± 0.011	**0.44 ± 0.01**	**0.45 ± 0.02**	**6.47 ± 0.87**	**4.90 ± 0.71**	**4.90 ± 0.86**
0.90	**27.71 ± 3.62**	38.87 ± 5.50	**29.75 ± 5.64**	**0.47 ± 0.02**	**0.46 ± 0.018**	**0.44 ± 0.03**	5.99 ± 1.12	4.97 ± 0.96	**2.88 ± 0.87**
0.95	**27.94 ± 6.13**	**34.18 ± 5.96**	**25.84 ± 7.29**	**0.47 ± 0.03**	**0.47 ± 0.04**	**0.45 ± 0.038**	5.26 ± 1.23	4.59 ± 1.24	**0.91 ± 0.16**

Table S2: **Statistical significance tests for Enamine 50k Bayesian optimization.** Pairwise comparison of uncertainty quantification methods for upper confidence bound (UCB) acquisition in Bayesian optimization on Enamine 50k data (Fig. 5D). Fold changes (FC; mean ± s.d.) reflect the fold change between the mean percentage of top-500 scores found for the first method listed relative to the second method listed. Significance values reflect the result of two-tailed unpaired t-tests over $n = 10$ independent trials.

	Evidence vs. Dropout		**Ensemble vs. Evidence**		**Ensemble vs. Dropout**	
Ligands Explored	**FC**	**p-value**	**FC**	**p-value**	**FC**	**p-value**
550	0.959 ± 0.56	0.003669	0.926 ± 0.33	0.195383	0.845 ± 0.60	0.477471
775	1.052 ± 0.17	0.259450	1.144 ± 0.17	0.016443	1.196 ± 0.12	0.001081
1000	1.039 ± 0.11	0.242781	1.104 ± 0.11	0.013200	1.142 ± 0.09	0.000964
1225	1.057 ± 0.05	0.009547	1.073 ± 0.08	0.013934	1.131 ± 0.06	0.000046
1450	1.055 ± 0.05	0.007914	1.065 ± 0.06	0.007536	1.122 ± 0.05	0.000007
1675	1.056 ± 0.05	0.006281	1.045 ± 0.04	0.003669	1.104 ± 0.04	0.000010

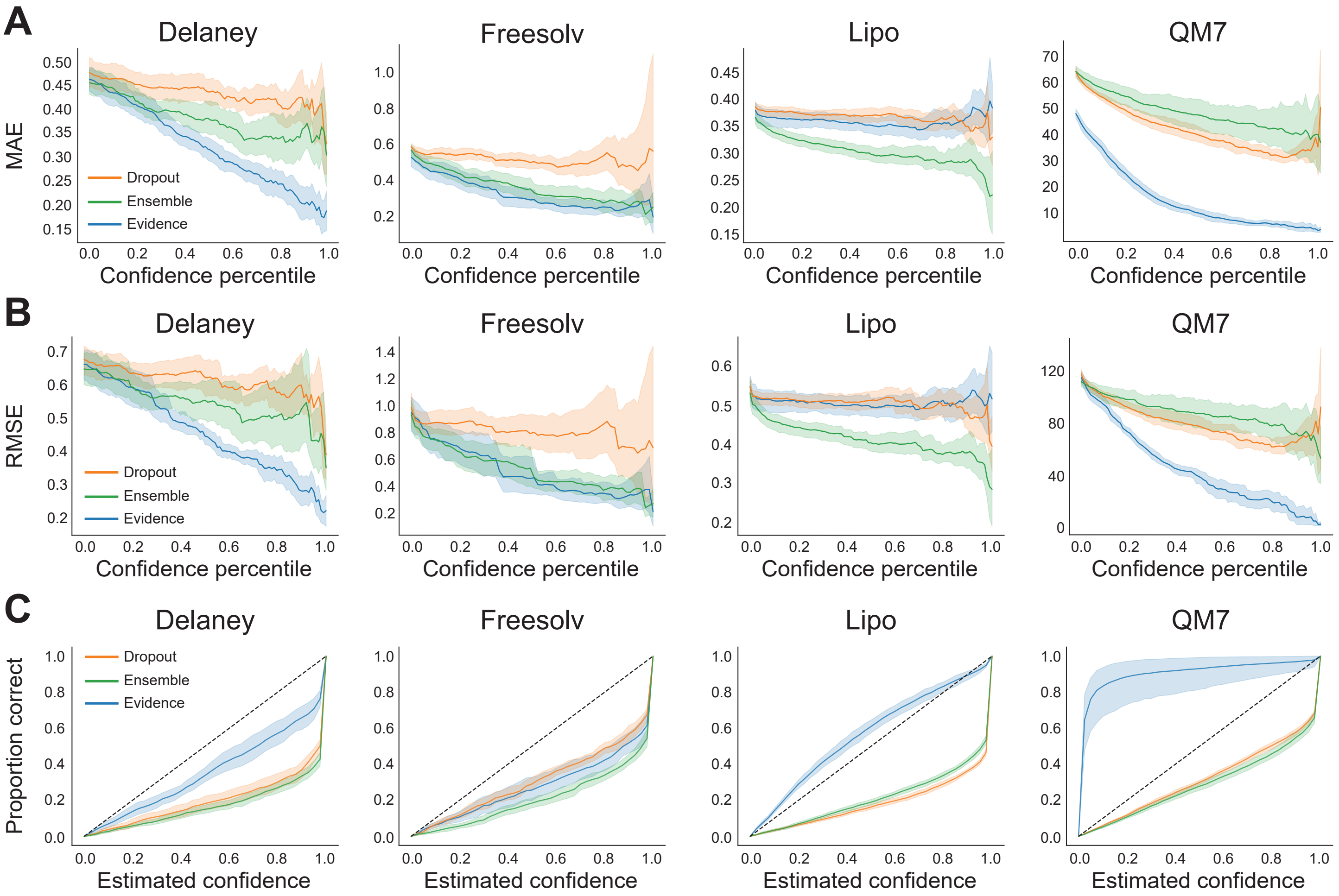

Figure S1: **Uncertainty benchmarking and calibration for lower-N datasets. (A, B)** Prediction error, measured as MAE (A) or RMSE (B), at different confidence percentile cutoffs for D-MPNNs evaluated on each of the 2D lower-N datasets. **(C)** Estimated confidence (cumulative probability) against the observed proportion correct for an evidential D-MPNN evaluated on each of the 2D lower-N datasets, with regularization parameter $\lambda = 0.2$. Mean ± 95% c.i., $n = 10$ independent trials.

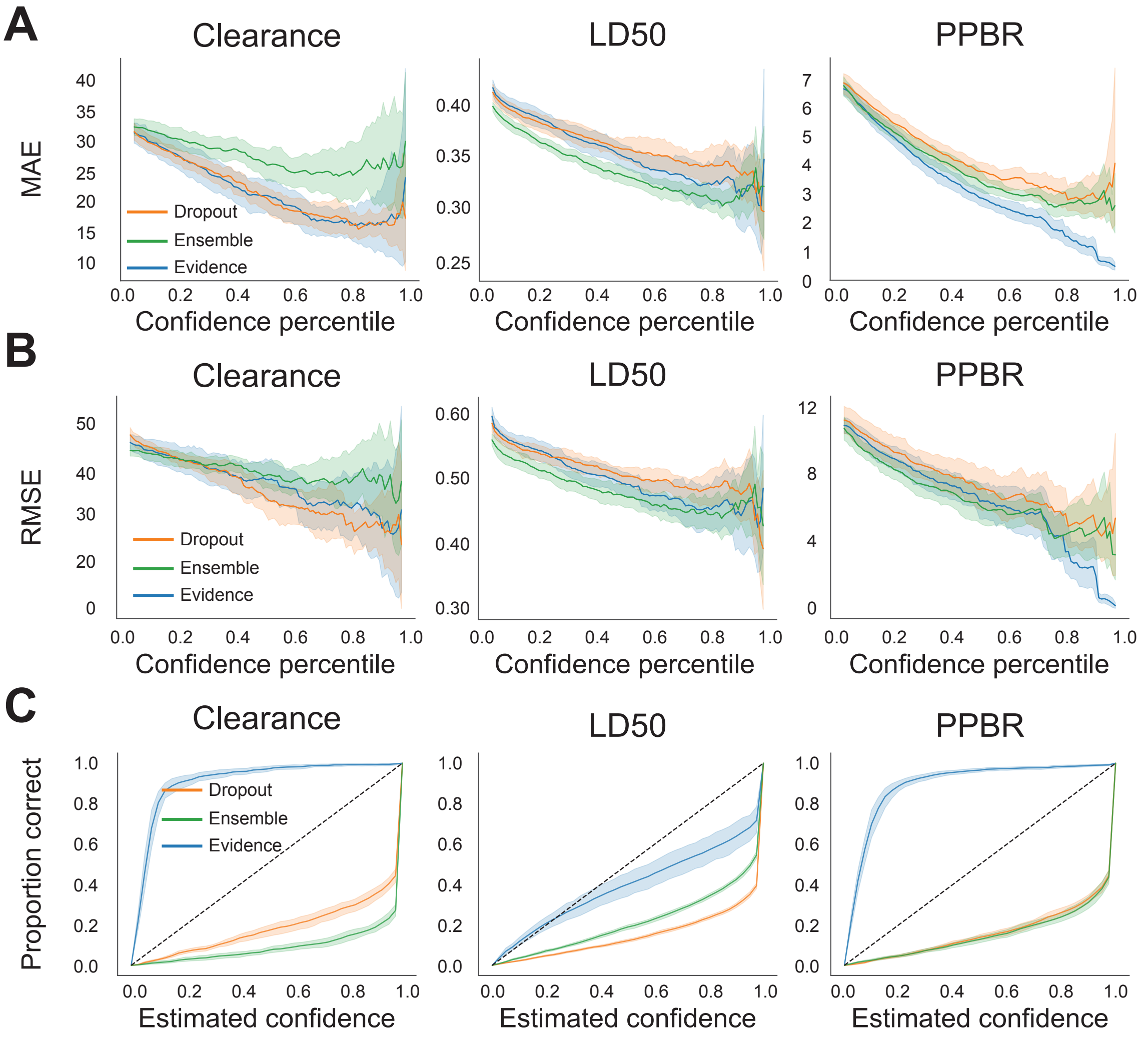

Figure S2: **Uncertainty benchmarking and calibration for additional lower-N Therapeutics Data Commons datasets.** **(A, B)** Prediction error, measured as MAE (A) or RMSE (B), at different confidence percentile cutoffs for D-MPNNs evaluated on each of the 2D lower-N datasets. **(C)** Estimated confidence (cumulative probability) against the observed proportion correct for an evidential D-MPNN evaluated on each of the 2D lower-N datasets, with regularization parameter $\lambda = 0.2$. Mean ± 95% c.i., $n = 10$ independent trials.

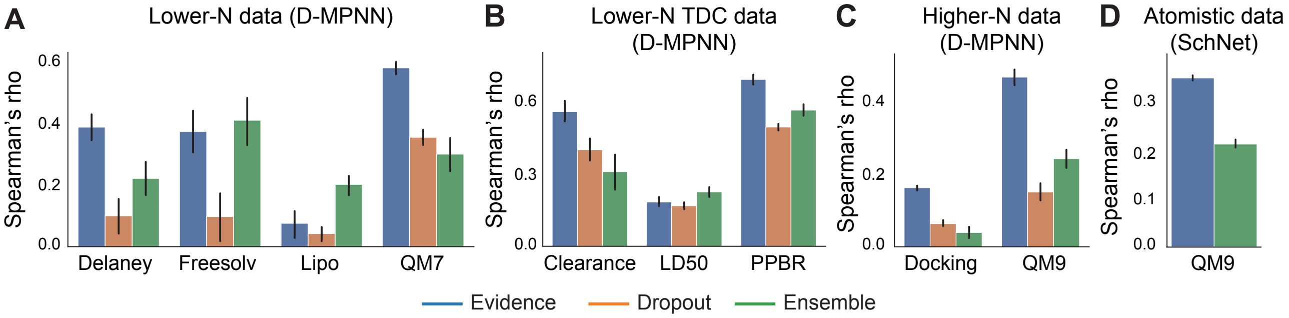

Figure S3: **Spearman rank correlation between error and uncertainty.** Spearman rank correlation coefficient between the estimated uncertainty and the absolute error for each point across lower-N **(A)**, additional lower-N Therapeutics Data Commons (TDC) **(B)**, higher-N **(C)**, and atomistic **(D)** datasets. Mean ± 95% c.i., $n = 10$ independent trials for lower-M and $n = 5$ independent trials for atomistic and higher-N datasets. For QM9 in the higher-N D-MPNN setting (C), standard error bars are computed across all tasks and independent trials ($n = 60$).

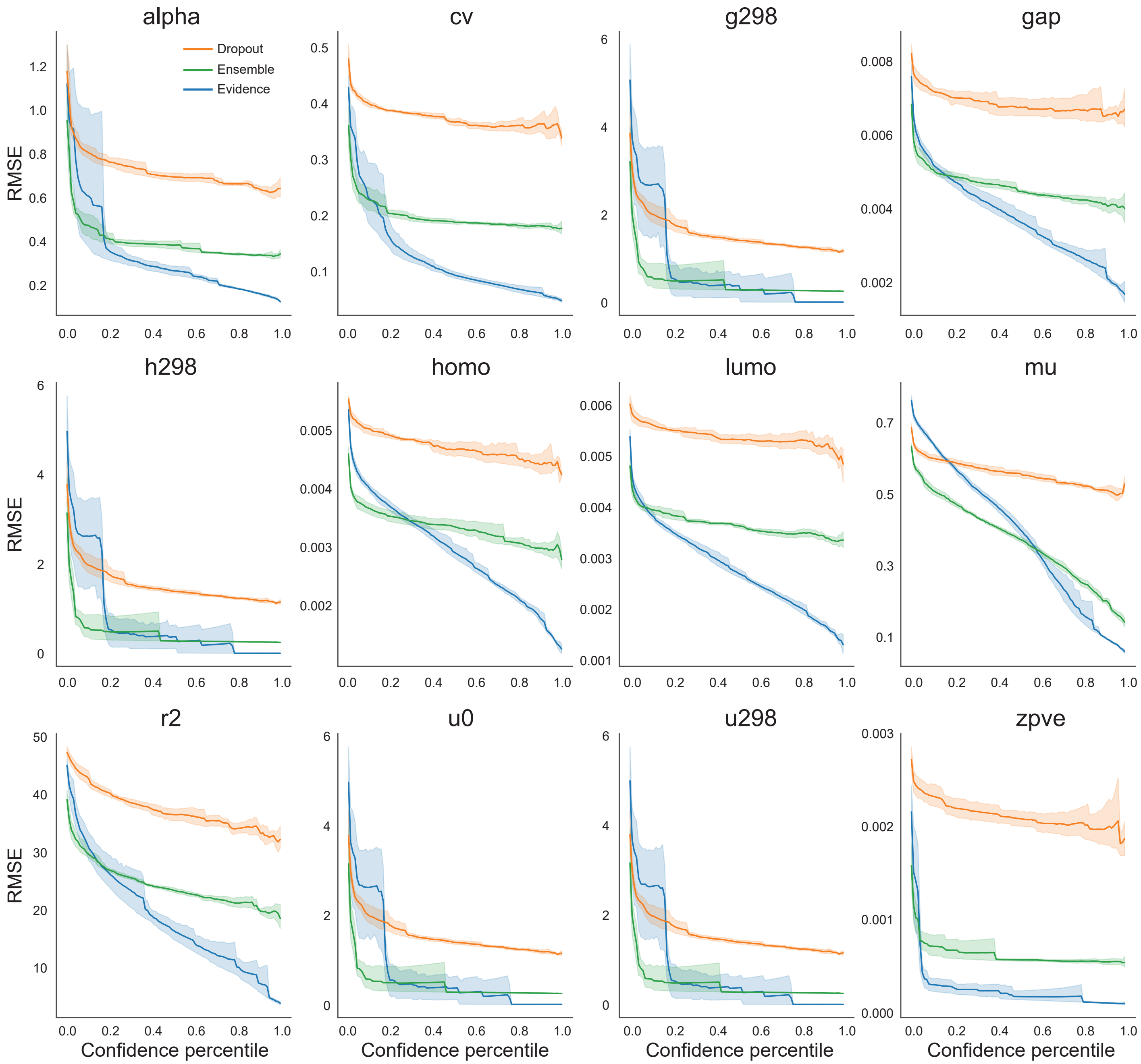

Figure S4: **Task-specific cutoffs for QM9 dataset.** RMSE at different confidence percentile cutoffs for D-MPNNs evaluated on each of the individual tasks from the QM9 dataset. Mean ± 95% c.i., $n = 5$ independent trials.

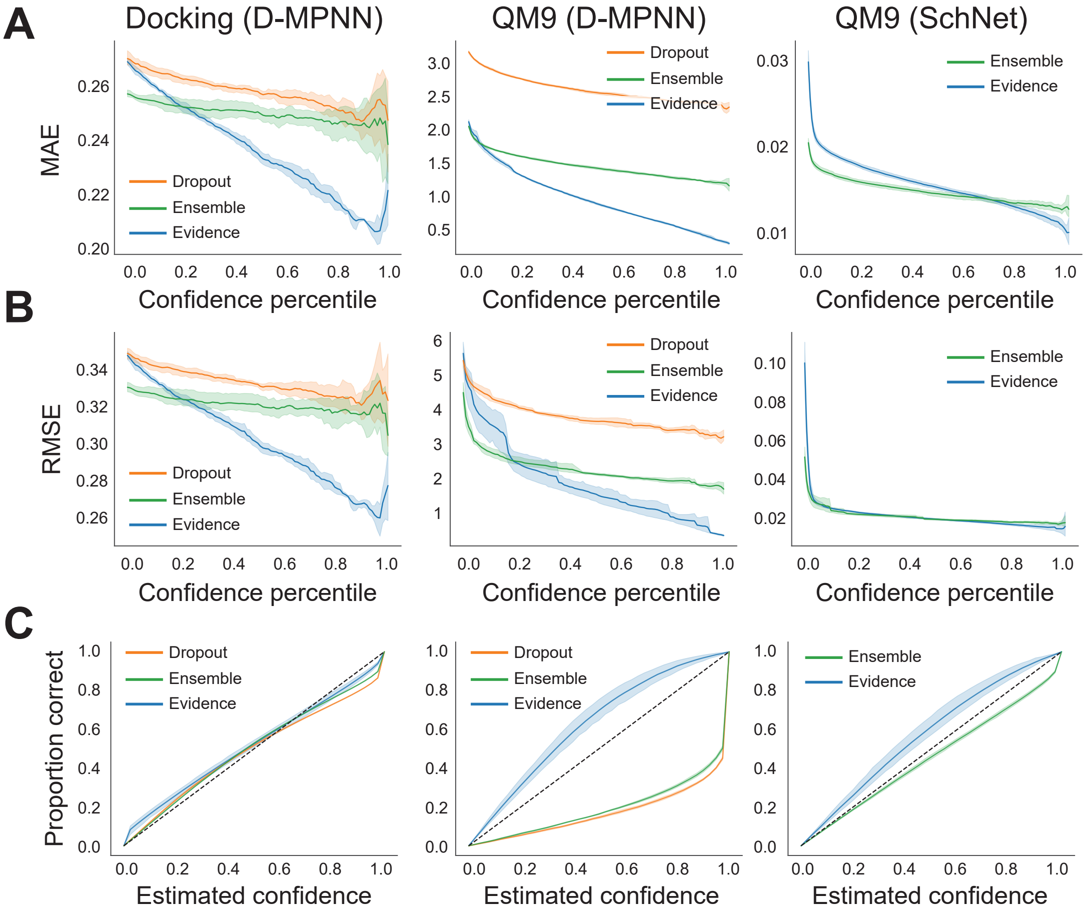

Figure S5: **Uncertainty benchmarking and calibration for higher-N 2D and 3D datasets. (A, B)** Prediction error, measured as MAE (A) or RMSE (B), at different confidence percentile cutoffs for models evaluated on the higher-N 2D and 3D datasets tested. **(C)** Estimated confidence (cumulative probability) against the observed proportion correct for the higher-N 2D and 3D datasets tested. Mean ± 95% c.i., $n = 5$ independent trials.

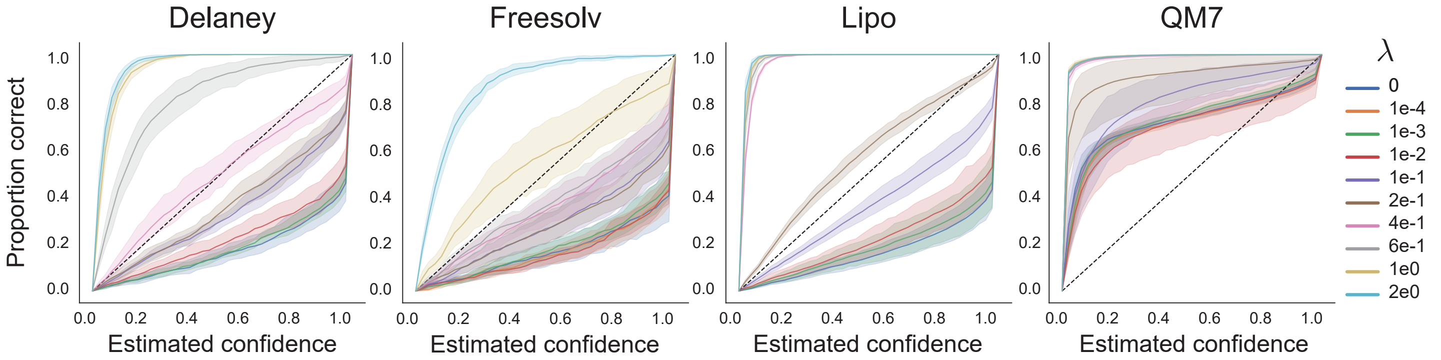

Figure S6: **Effect of λ on uncertainty calibration.** Evidential D-MPNNs are trained with different regularization coefficients λ on each of the lower-N datasets. Estimated confidence (cumulative probability) against the observed proportion correct is computed and plotted across different λ. Mean ± 95% c.i., $n = 10$ independent trials.

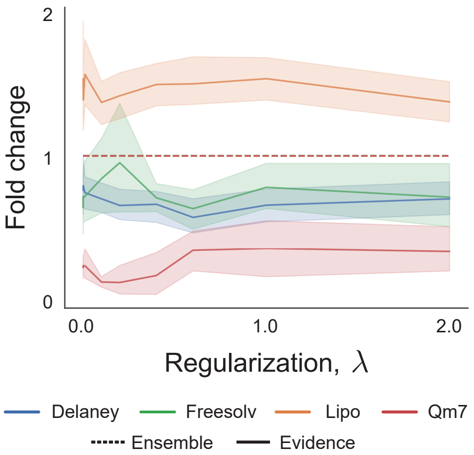

Figure S7: **Cutoff RMSE is robust to small $\boldsymbol{\lambda}$ shifts.** Evidential D-MPNNs are trained with different regularization coefficients λ on each of the lower-N datasets. RMSE is computed at the 10% confidence cutoff for each model. All computed errors are reported as the fold change of the top 10% RMSE for the evidential method relative to that of the ensemble baseline. Mean ± 95% c.i., $n = 10$ independent trials.

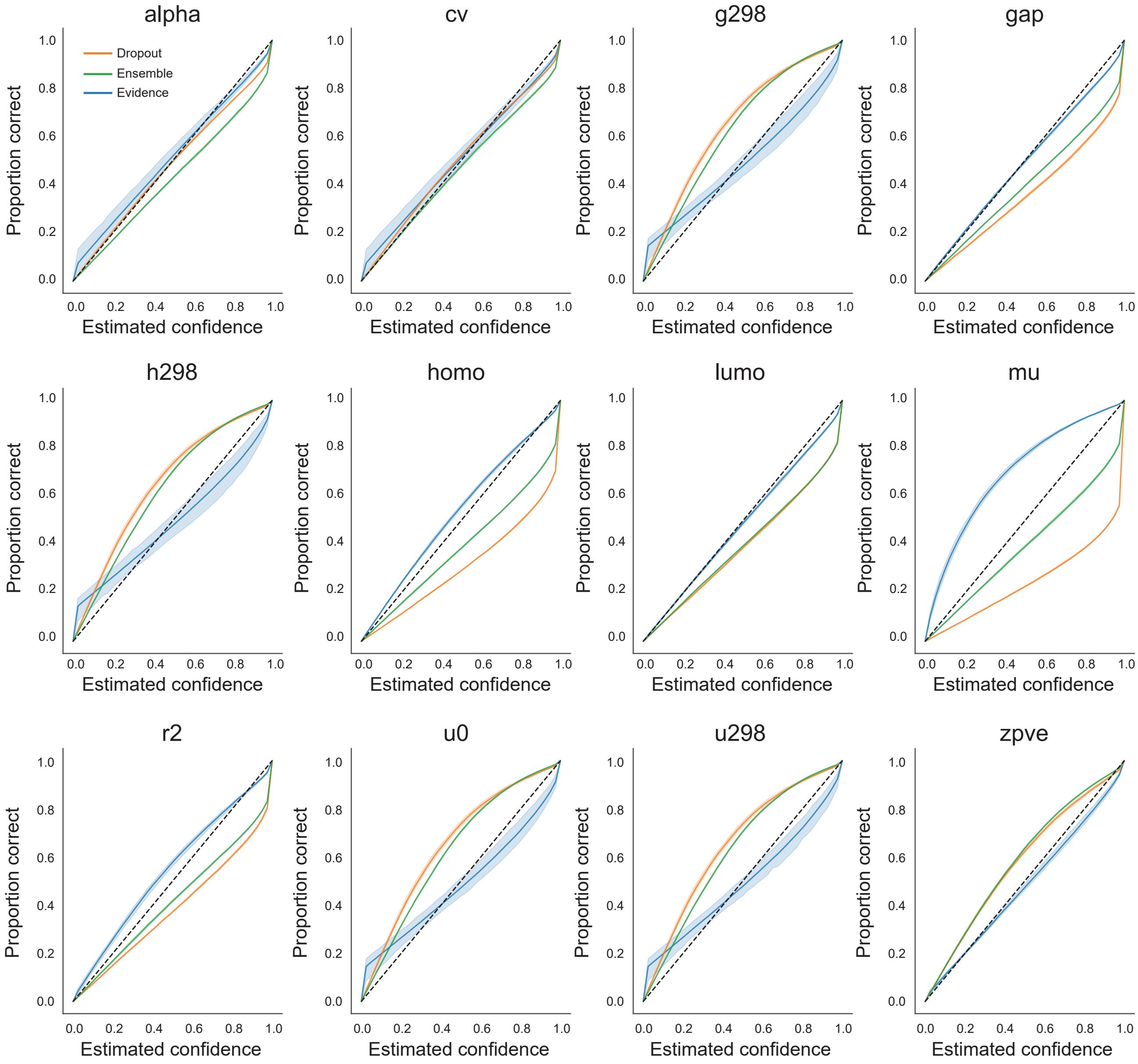

Figure S8: **Task-specific calibration for QM9 dataset.** Estimated confidence (cumulative probability) against the observed proportion correct is computed for an evidential D-MPNN evaluated on QM9 and then broken down into task-specific plots. Mean ± 95% c.i., $n = 5$ independent trials.

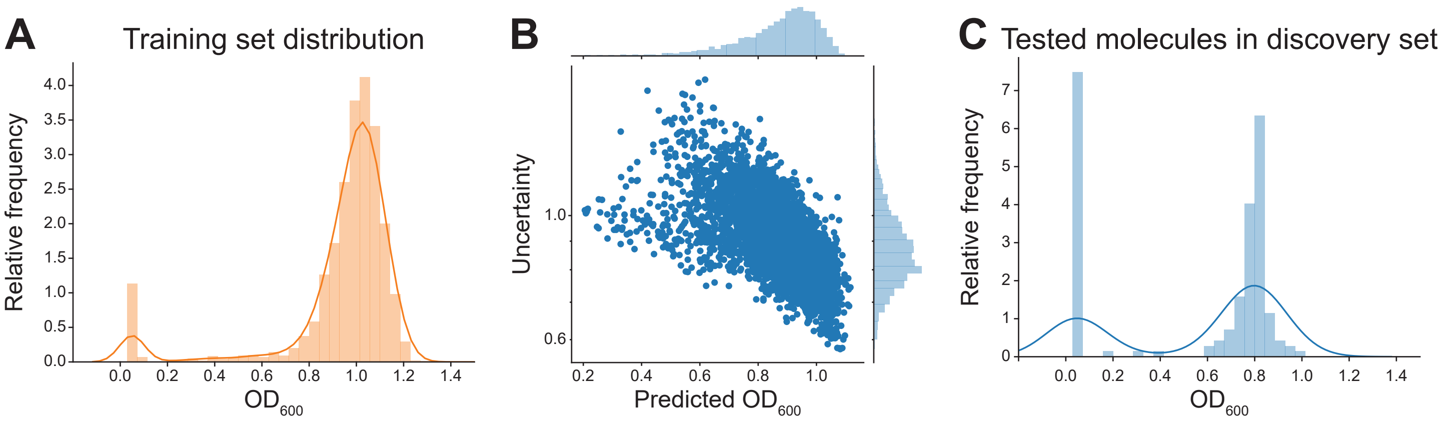

Figure S9: **Antibiotic discovery datasets and uncertainty predictions.** **(A)** Distribution of OD_{600} values for the training dataset of small molecules and their *in vitro* growth inhibitory activity against *E. coli*, as originally measured by Stokes et al.[39] Lower OD_{600} values indicate less *E. coli* growth and hence correspond to greater antibiotic activity. **(B)** Distribution of predicted OD_{600} values and evidential uncertainties for molecules in the Broad Drug Repurposing Hub discovery dataset. **(C)** Distribution of empirically determined OD_{600} for the subset of the discovery set (162 out of $6,111$ total molecules) that was experimentally tested for *in vitro* growth inhibitory activity against *E. coli.*

References

(1) Nigam, A.; Pollice, R.; Hurley, M. F.; Hickman, R. J.; Aldeghi, M.; Yoshikawa, N.; Chithrananda, S.; Voelz, V. A.; Aspuru-Guzik, A. Assigning Confidence to Molecular Property Prediction. *arXiv preprint arXiv:2102.11439* **2021**,

(2) Lamb, G.; Paige, B. Bayesian Graph Neural Networks for Molecular Property Prediction. *arXiv preprint arXiv:2012.02089* **2020**,

(3) Jiménez-Luna, J.; Grisoni, F.; Schneider, G. Drug discovery with explainable artificial intelligence. *Nature Machine Intelligence* **2020**, *2*, 573–584.

(4) Cherkasov, A.; Muratov, E. N.; Fourches, D.; Varnek, A.; Baskin, I. I.; Cronin, M.; Dearden, J.; Gramatica, P.; Martin, Y. C.; Todeschini, R., et al. QSAR modeling: where have you been? Where are you going to? *Journal of medicinal chemistry* **2014**, *57*, 4977–5010.

(5) 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. *Chemical Society Reviews* **2020**, *49*, 3525–3564.

(6) Kendall, A.; Gal, Y. What uncertainties do we need in bayesian deep learning for computer vision? *arXiv preprint arXiv:1703.04977* **2017**,

(7) Eyke, N. S.; Green, W. H.; Jensen, K. F. Iterative experimental design based on active machine learning reduces the experimental burden associated with reaction screening. *Reaction Chemistry & Engineering* **2020**,

(8) Hie, B.; Bryson, B. D.; Berger, B. Leveraging uncertainty in machine learning accelerates biological discovery and design. *Cell Systems* **2020**, *11*, 461–477.

(9) Roy, A. G.; Ren, J.; Azizi, S.; Loh, A.; Natarajan, V.; Mustafa, B.; Pawlowski, N.; Freyberg, J.; Liu, Y.; Beaver, Z., et al. Does Your Dermatology Classifier Know What It Doesn't Know? Detecting the Long-Tail of Unseen Conditions. *arXiv preprint arXiv:2104.03829* **2021**,

(10) Janet, J. P.; Duan, C.; Yang, T.; Nandy, A.; Kulik, H. J. A quantitative uncertainty metric controls error in neural network-driven chemical discovery. *Chemical science* **2019**, *10*, 7913–7922.

(11) Tran, K.; Neiswanger, W.; Yoon, J.; Zhang, Q.; Xing, E.; Ulissi, Z. W. Methods for comparing uncertainty quantifications for material property predictions. *Machine Learning: Science and Technology* **2020**,

(12) Jospin, L. V.; Buntine, W.; Boussaid, F.; Laga, H.; Bennamoun, M. Hands-on Bayesian Neural Networks–a Tutorial for Deep Learning Users. *arXiv preprint arXiv:2007.06823* **2020**,

(13) Lakshminarayanan, B.; Pritzel, A.; Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems* **2017**, 6402–6413.

(14) Gal, Y.; Ghahramani, Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. *International Conference on Machine Learning* **2016**, 1050–1059.

(15) Hirschfeld, L.; Swanson, K.; Yang, K.; Barzilay, R.; Coley, C. W. Uncertainty Quantification Using Neural Networks for Molecular Property Prediction. *Journal of Chemical Information and Modeling* **2020**, *60*, 3770–3780, DOI: 10.1021/acs.jcim.0c00502, PMID: 32702986.

(16) Smith, J. S.; Nebgen, B.; Lubbers, N.; Isayev, O.; Roitberg, A. E. Less is more: Sampling chemical space with active learning. *The Journal of Chemical Physics* **2018**, *148*, 241733, ISBN: 0021-9606 Publisher: AIP Publishing LLC.

(17) Vandermause, J.; Torrisi, S. B.; Batzner, S.; Xie, Y.; Sun, L.; Kolpak, A. M.; Kozinsky, B. On-the-fly active learning of interpretable Bayesian force fields for atomistic rare events. *npj Computational Materials* **2020**, *6*, 1–11.

(18) Klicpera, J.; Giri, S.; Margraf, J. T.; Günnemann, S. Fast and Uncertainty-Aware Directional Message Passing for Non-Equilibrium Molecules. *arXiv preprint arXiv:2011.14115* **2020**,

(19) Klicpera, J.; Groß, J.; Günnemann, S. Directional message passing for molecular graphs. *arXiv preprint arXiv:2003.03123* **2020**,

(20) Nix, D.; Weigend, A. Estimating the Mean and Variance of the Target Probability Distribution. *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)* **1994**, *1*, 55–60 vol.1, DOI: 10.1109/ICNN.1994.374138.

(21) Bishop, C. M. Mixture density networks. **1994**,

(22) Gilitschenski, I.; Sahoo, R.; Schwarting, W.; Amini, A.; Karaman, S.; Rus, D. Deep Orientation Uncertainty Learning based on a Bingham Loss. *International Conference on Learning Representations* **2019**,

(23) Amini, A.; Soleimany, A. P.; Schwarting, W.; Bhatia, S. N.; Rus, D. Uncovering and mitigating algorithmic bias through learned latent structure. Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society. 2019; pp 289–295.

(24) Gurevich, P.; Stuke, H. Gradient conjugate priors and multi-layer neural networks. *Artificial Intelligence* **2020**, *278*, 103184.

(25) 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. *Journal of Chemical Information and Modeling* **2020**, ISBN: 1549-9596 Publisher: ACS Publications.

(26) Sensoy, M.; Kaplan, L.; Kandemir, M. Evidential deep learning to quantify classification uncertainty. *Advances in Neural Information Processing Systems* **2018**, 3179–3189.

(27) Amini, A.; Schwarting, W.; Soleimany, A.; Rus, D. Deep evidential regression. *Advances in Neural Information Processing Systems* **2020**, *33*.

(28) Goodfellow, I.; Bengio, Y.; Courville, A. 6.2. 2.3 softmax units for multinoulli output distributions. *Deep learning* **2016**, 180–184.

(29) Wu, Z.; Ramsundar, B.; Feinberg, E. N.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; Pande, V. MoleculeNet: a benchmark for molecular machine learning. *Chemical science* **2018**, *9*, 513–530, Publisher: Royal Society of Chemistry.

(30) Huang, K.; Fu, T.; Gao, W.; Zhao, Y.; Roohani, Y.; Leskovec, J.; Coley, C. W.; Xiao, C.; Sun, J.; Zitnik, M. Therapeutics data Commons: machine learning datasets and tasks for therapeutics. *arXiv preprint arXiv:2102.09548* **2021**,

(31) Gorgulla, C.; Boeszoermenyi, A.; Wang, Z.-F.; Fischer, P. D.; Coote, P. W.; Das, K. M. P.; Malets, Y. S.; Radchenko, D. S.; Moroz, Y. S.; Scott, D. A., et al. An open-source drug discovery platform enables ultra-large virtual screens. *Nature* **2020**, *580*, 663–668.

(32) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; Von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data* **2014**, *1*, 1–7.

(33) Jastrzębbski, S.; Szymczak, M.; Pocha, A.; Mordalski, S.; Tabor, J.; Bojarski, A. J.; Podlewska, S. Emulating docking results using a deep neural network: a new perspective for virtual screening. *Journal of Chemical Information and Modeling* **2020**, *60*, 4246–4262.

(34) Graff, D. E.; Shakhnovich, E. I.; Coley, C. W. Accelerating high-throughput virtual screening through molecular pool-based active learning. *arXiv preprint arXiv:2012.07127* **2020**,

(35) Trott, O.; Olson, A. J. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry* **2010**, *31*, 455–461.

(36) Kuleshov, V.; Fenner, N.; Ermon, S. Accurate uncertainties for deep learning using calibrated regression. *International Conference on Machine Learning* **2018**, 2796–2804.

(37) Frazier, P. I. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811* **2018**,

(38) Hernández-Lobato, J. M.; Requeima, J.; Pyzer-Knapp, E. O.; Aspuru-Guzik, A. Parallel and distributed Thompson sampling for large-scale accelerated exploration of chemical space. *International conference on machine learning* **2017**, 1470–1479.

(39) 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.

(40) Corsello, S. M.; Bittker, J. A.; Liu, Z.; Gould, J.; McCarren, P.; Hirschman, J. E.; Johnston, S. E.; Vrcic, A.; Wong, B.; Khan, M., et al. The Drug Repurposing Hub: a next-generation drug library and information resource. *Nature Medicine* **2017**, *23*, 405–408.

(41) Blundell, C.; Cornebise, J.; Kavukcuoglu, K.; Wierstra, D. Weight uncertainty in neural network. *International Conference on Machine Learning* **2015**, 1613–1622.

(42) Imbalzano, G.; Zhuang, Y.; Kapil, V.; Rossi, K.; Engel, E. A.; Grasselli, F.; Ceriotti, M. Uncertainty estimation for molecular dynamics and sampling. *The Journal of Chemical Physics* **2021**, *154*, 074102.

(43) Rufa, D. A.; Macdonald, H. E. B.; Fass, J.; Wieder, M.; Grinaway, P. B.; Roitberg, A. E.; Isayev, O.; Chodera, J. D. Towards chemical accuracy for alchemical free energy calculations with hybrid physics-based machine learning/molecular mechanics potentials. *BioRxiv* **2020**,

(44) Schütt, K.; Kindermans, P.-J.; Felix, H. E. S.; Chmiela, S.; Tkatchenko, A.; Müller, K.-R. SchNet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in neural information processing systems* **2017**, 991–1001.

(45) Townshend, R. J.; Vögele, M.; Suriana, P.; Derry, A.; Powers, A.; Laloudakis, Y.; Balachandar, S.; Anderson, B.; Eismann, S.; Kondor, R., et al. ATOM3D: Tasks On Molecules in Three Dimensions. *arXiv preprint arXiv:2012.04035* **2020**,

(46) Kearnes, S. Pursuing a Prospective Perspective. *Trends in Chemistry* **2020**,

(47) Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703* **2019**,

(48) Landrum, G., et al. Rdkit: Open-source cheminformatics software. *GitHub and SourceForge* **2016**, *10*, 3592822.

(49) Murphy, K. P. Conjugate Bayesian analysis of the Gaussian distribution. *def* **2007**, *1*, 16.

(50) Yang, K.; Swanson, K.; Jin, W.; Coley, C.; Eiden, P.; Gao, H.; Guzman-Perez, A.; Hopper, T.; Kelley, B.; Mathea, M., et al. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling* **2019**, *59*, 3370–3388.

(51) Schütt, K. T.; Kessel, P.; Gastegger, M.; Nicoli, K. A.; Tkatchenko, A.; Müller, K.-R. SchNetPack: A Deep Learning Toolbox For Atomistic Systems. *Journal of Chemical Theory and Computation* **2019**, *15*, 448–455, DOI: `10.1021/acs.jctc.8b00908`, Publisher: American Chemical Society.

(52) Wenlock, M.; Tomkinson, N. Experimental in Vitro DMPK and Physicochemical Data on a Set of Publicly Disclosed Compounds. **2015**,

(53) Zhu, H.; Martin, T. M.; Ye, L.; Sedykh, A.; Young, D. M.; Tropsha, A. Quantitative structure-activity relationship modeling of rat acute toxicity by oral exposure. *Chemical research in toxicology* **2009**, *22*, 1913–1921.

(54) Ruddigkeit, L.; Van Deursen, R.; Blum, L. C.; Reymond, J.-L. Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17. *Journal of Chemical Information and Modeling* **2012**, *52*, 2864–2875.

(55) Virtanen, P.; Gommers, R.; Oliphant, T. E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J., et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods* **2020**, *17*, 261–272.