Introduction

Undesirable pharmacokinetics and toxicity are major contributors to drug development failures. It is widely recognized that evaluating the absorption, distribution, metabolism, excretion, and toxicity (ADMET) of chemicals early on is crucial. To evaluate the ADMET comprehensively and accurately and physicochemical properties of molecules, alongside their pharmaceutical chemical friendliness, the ADMETlab platform undergoes continuous upgrades from ADMETlab 1.0 to ADMETlab 2.0 and now to the current ADMETlab 3.0. With enhancements in ADMET training data, the utilization of more robust model frameworks, the integration of specific API functionalities, and the provision of uncertainty assessments, ADMETlab 3.0 has significantly expanded its capabilities, aiding medicinal chemists in accelerating the drug development process.

图片包含 图形用户界面

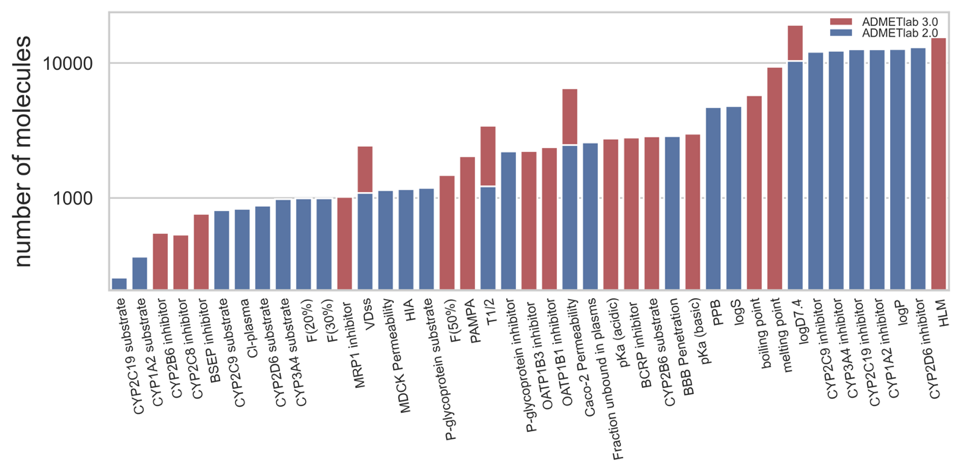
描述已自动生成

What’s New

1. Comprehensive coverage of ADMET endpoint data

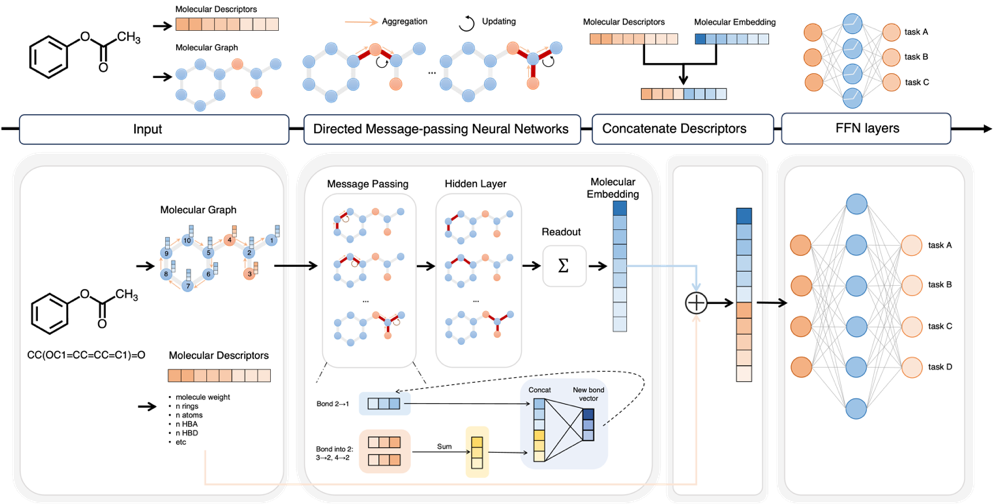
ADMETlab 3.0 marks a significant upgrade, expanding the dataset to over 400000 data points covering 119 endpoints, including 30 new endpoints and updates to 4 existing ones. The new endpoints cover absorption, distribution, metabolism, toxicity, physicochemical properties, and drug chemistry. Compiled from scholarly sources and databases like ChEMBL and PubChem, these additions offer a more comprehensive understanding of molecular behavior, enhancing analysis across in vitro and in vivo stages. The updates and additions to the data are illustrated in Figures 1 and 2. Red indicates newly added data, while blue represents existing data.

**Figure 1**. The bar chart illustrates the dataset sizes for absorption, distribution, metabolism, and physicochemical properties. The red bars represent the volume of newly added data, while the blue bars indicate the existing data volume.



**Figure 2**. The bar chart illustrates the dataset sizes for toxicity properties. The red bars represent the volume of newly added data, while the blue bars indicate the existing data volume.

2. Robust and accurate multi-task DMPNN models



**Figure 3**. DMPNN-Des Model Framework

At its core, the DMPNN is a Graph Convolutional Neural Network (GCNN) specifically designed for molecular graphs. This model has two notable features: (1) operating on mixed representations that combine convolution and descriptors, allowing flexibility in task-specific encoding while providing strong priors with fixed descriptors, and (2) learning to construct molecular encodings through bond-centered convolutions instead of atom-centered convolutions, thus avoiding unnecessary loops during the message passing phase.

This integrated framework aims to enhance the model’s capability in accurately predicting molecular properties, especially when global features play a crucial role. By amalgamating these additional features with the message passing architecture of DMPNN, it creates a more robust and nuanced representation of molecular structures, thereby bolstering the model’s predictive power across various chemical properties.

3. API integration and architecture upgrades

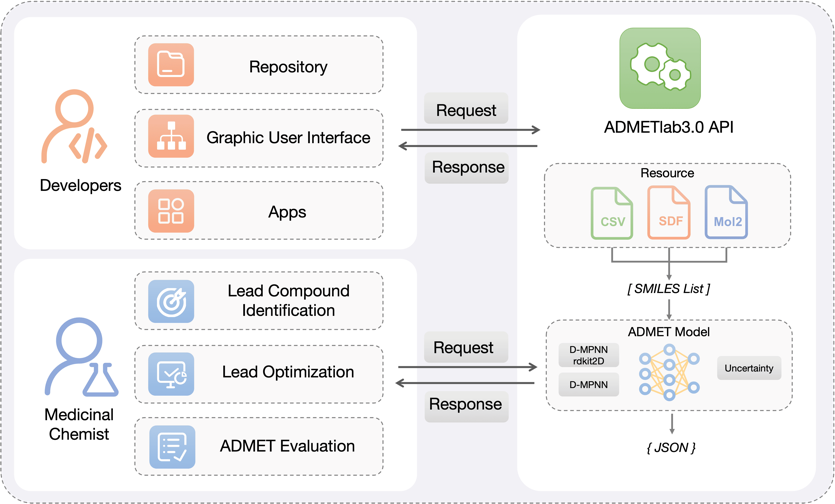


Figure 4. ADMETlab 3.0 API

The Application Programming Interface (API) offers researchers command-line options, facilitating access to ADMETlab3.0 and ensuring comprehensive exploration of ADMET-related research on datasets. This API operates using standard protocols and user-friendly programming languages such as Python, R, and bash, enabling effortless access to ADMETlab3.0’s computational models via simple scripts. For a tutorial with sample code, please visit the following link: http://121.40.210.46:8097/devDoc/dev-affix. The conceptual framework of ADMETlab 3.0 is primarily composed of three-part request: 1) input—cleaning individual molecules or batches of molecules; 2) operation—calculating the ADMET properties of these molecules using deep learning models; and 3) output—determining which result files to return. The beauty of this design is that all the underlying functionalities are modular, allowing for flexible combinations.

3．Uncertainty evaluation

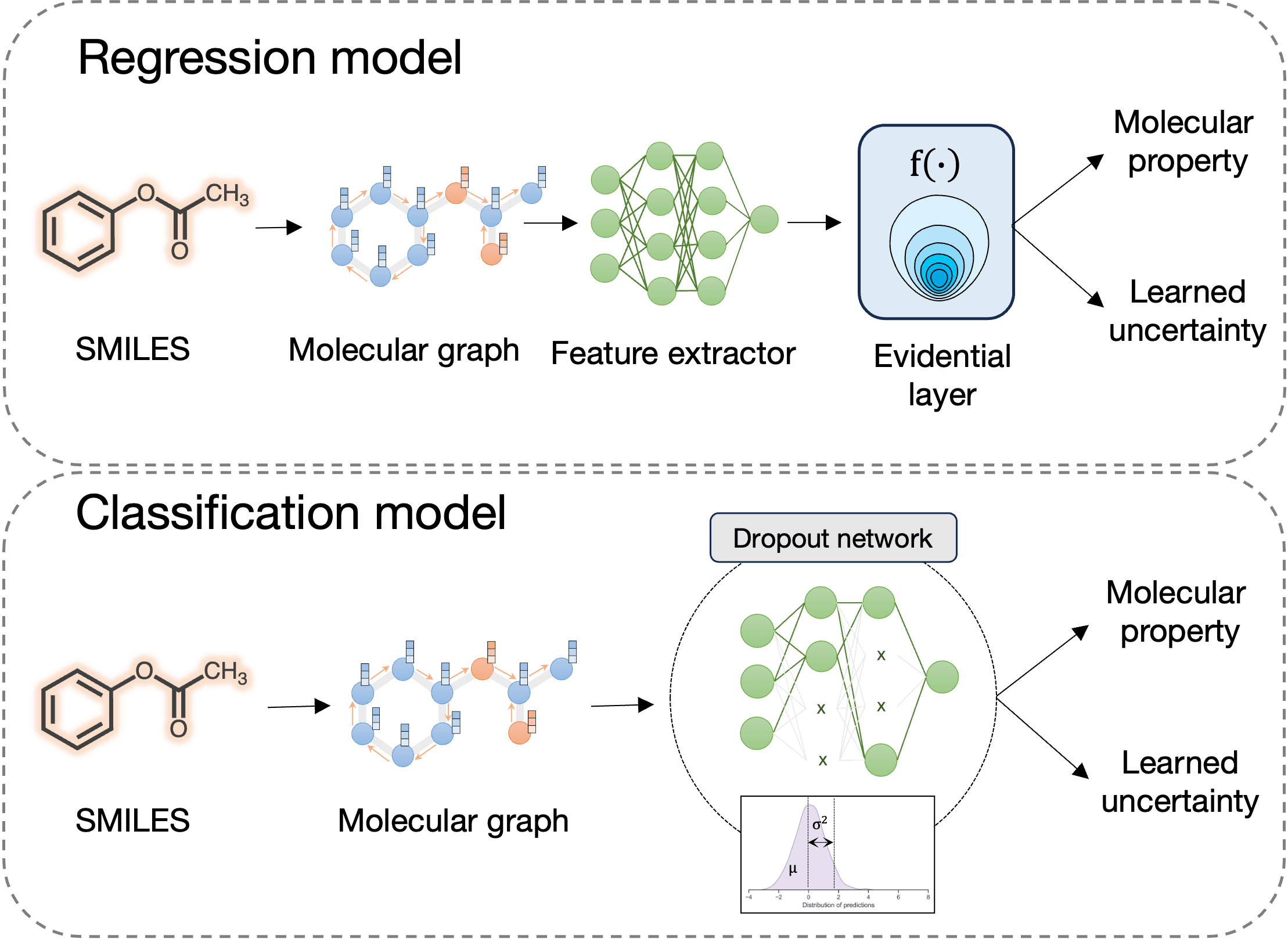


Figure 5. Schematic Representation of Uncertainty Estimation in Regression and Classification Models.

In the regression model, an evidence-based deep learning approach is utilized to forecast target properties and deduce parameters of the underlying evidential distribution. This method captures the support for each prediction, enabling the estimation of uncertainty.

In the classification model, Monte Carlo dropout is employed to assess uncertainty across different properties. This technique generates a distribution of prediction outcomes, presenting both the prediction mean and variance for each property. The mean of this distribution signifies the predicted value, while the variance σ² serves as a measure of uncertainty.

1. DMPNN framework

Yang et al(1). introduced an open-source Python package called Chemprop designed for implementing DMPNN models. This package offers a robust and efficient solution tailored for molecular property prediction tasks and has garnered extensive utilization in fields such as drug discovery and materials science(2). Within the DMPNN framework (Figure 3), there exist two distinct stages: message passing and readout. Here, a graph G serves as an illustrative example, representing node (atom) features as and edge (bond) features as .Initially, the edge hidden state is initialized using eq 1. This equation concatenate atom and bond features by passing them through the learned matrix and applying the rectified linear unit (RELU) activation function. This initialization defines the edge hidden state, which undergoes subsequent updates during the message passing process. In eq 1, represent the RELU activation function, denotes a learned matrix, and signifies a basic concatenation operation.

(1)

The first phase of message passing computes interactions between atom and atom . This is achieved by summing the hidden states of all bonds connected to atom while excluding the hidden state of bonds from atom , as described by in eq 2. This step captures information about the neighboring atoms and their connections to individual atoms within . In eq 2, represents bond features at layer *，*  denotes the set of nodes connected to excluding .

(2)

Following this, a new hidden message at depth 1 is created by summing the product of the initial hidden state and the learned matrix with the message. This resultant output undergoes further processing using the activation function ，denoted as in eq 3.

(3)

In the final message passing layer (at t=T), the updated hidden states are summed to create the ultimate message for each atom, as described in eq 4. This action aggregates information about all neighboring atoms and their relationships into the final message for each atom.

(4)

The hidden state for each atom is derived by concatenating the initial atom features with the message vector, as indicated in eq 5.

(5)

Finally, employing eq 6, the hidden states ​ of each atom are summed to generate a molecular feature vector. This step aggregate information from all atoms in the molecule into a unified molecular feature vector, facilitating property prediction. It encompasses both structural and attribute information of the entire molecule, offering a comprehensive representation for further property prediction.

(6)

In the DMPNN-Des model, preceding the readout phase, vector is concatenated with descriptor vectors and collectively processed using a fully connected feedforward neural network to predict properties. The algorithm is implemented using the open-source Chemprop package(2). Regarding the raw datasets, we trained each dataset using Chemprop, employing random segmentation ratios [0.8, 0.1, 0.1] for training, testing, and evaluation. The batch process is iterated five times, and the average RMSE value and variance from these iterations are calculated to evaluate the robustness of the model.

1. Uncertainty estimation

In pharmaceutical research, especially in ADMET assessment, unreliable predictions might lead to misjudging drug efficacy, causing missed opportunities in drug development. Therefore, within AI-assisted drug development, quantifying predictive reliability is crucial for guiding subsequent research decisions by medicinal chemists (3).

**Regression model**

Within the regression model, ADMETlab 3.0 employs the evidence-based deep learning technique proposed by Amini et al (4). Evidential deep learning extends the concept of learning probability distribution parameters, further predicting higher-order distributions of the original likelihood parameters themselves. These higher-order parameters define the evidence distribution, thereby capturing the model's predictions and the degree of evidence associated with those predictions. In contrast to Bayesian neural networks that set priors on neural network weights, the evidence learning approach estimates uncertainty by directly learning the parameters defining this evidence distribution. It encompasses both epistemic and aleatoric uncertainties, eliminating the need for sampling and thus obviating the necessity of sampling procedures.

In a regression setting, the training samples comprise , where the target values comprise an i.i.d. Gaussian distribution defined by mean and variance . Within the context of an evidential depth model, these parameters are presumed unknown and replaced by probabilistic estimates, achieved by placing a Gaussian distribution on the unknown mean μ and an Inverse-Gamma prior on the unknown variance . We obtain a higher-order distribution (also termed the evidential distribution), depicted as , represented by a Normal Inverse-Gamma distribution. This evidential distribution is determined through four parameters, m =. The model can capture predictive uncertainty by learning these parameters, as demonstrated in Figure 5. In this work, the uncertainty estimation of the regression model is achieved by setting the chemprop package(2) uncertainty\_method to evidential\_total.

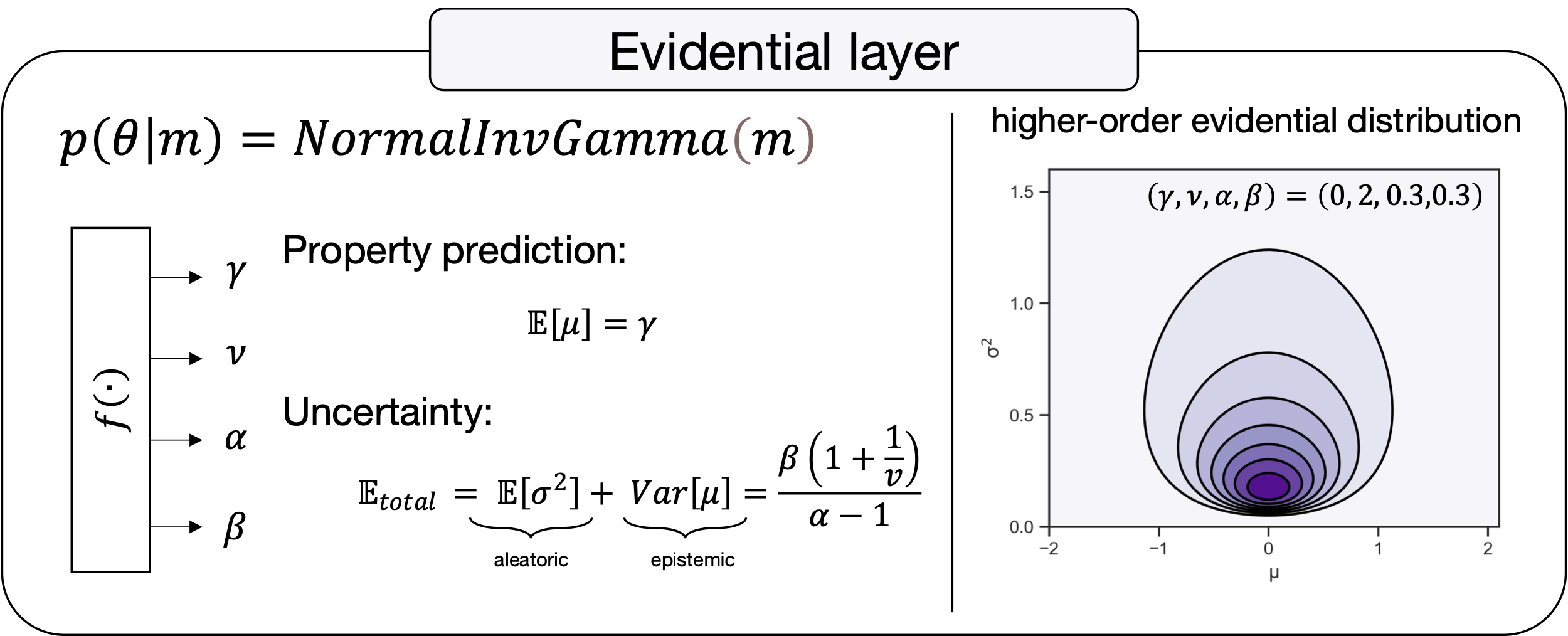


Figure 5. For continuous regression learning problems, the evidential distribution p(θ|m) has the flexibility to adopt a Normal-Inverse-Gamma distribution, defined by m = . This adaptation facilitates the simultaneous achievement of property prediction and uncertainty estimation. The variation in color signifies the likelihood density, with darker shades indicating higher density.

Moreover, to better assist users in assessing the reliability of model predictions based on model uncertainty, furthermore, we furnish the range of model RMSE within different uncertainty intervals. This aids users in assessing the reliability of model predictions based on varying levels of uncertainty.

**Classification model**

In the classification model, uncertainty is estimated using the Monte Carlo dropout t approach(5). This method considers dropout in deep neural networks as an approximate Bayesian inference of deep Gaussian processes. Specifically, the dropout technique involves applying dropout before each layer during training and maintaining dropout activation during the inference process. This allows for the generation of prediction distributions using different random masks, approximating the posterior of deep Gaussian processes (figure 5). The variance of this distribution serves as an estimation of predictive uncertainty(6)(7)(8).

During experiments with uncertainty quantification, an ensemble of models generates a prediction distribution for each molecule using a dropout-enabled network with a sample size of 10. The probability of 0.1 to use for Monte Carlo dropout uncertainty estimation. Let represent the prediction from a single model within the ensemble, which contains T = 10 models. For a query sample , the prediction is represented as the means of all predictions and the uncertainty of this sample can be provided by the variance  of the prediction distribution.

(7)

We employed the method proposed by Dolezal et al(9). to determine the optimal uncertainty threshold value, θ. This method establishes an uncertainty threshold, where predictions below this threshold are more likely to be correct than those with higher levels of uncertainty. To find the uncertainty threshold that optimally separates predictions into likely-correct (high-confidence) and likely-incorrect (low-confidence), we calculated the sensitivity and specificity for misprediction for all possible uncertainty thresholds. The corresponding Youden’s index (J) for each uncertainty threshold is the calculated as

(8)

The optimal uncertainty threshold is the defined as the threshold which maximized the Youden’s index:

(9)

The single threshold is than used for all predicted made by the model. We take a binary a binary approach to confidence using the uncertainty threshold, with confidence of the classification model defined as

(10)

In other words, prediction uncertainty exceeding this value designates the model's prediction as low confidence, while prediction uncertainty below this threshold indicates high confidence in the model's prediction. This threshold will be used to assess the reliability of prediction in classification tasks within the ADMET models. The uncertainty thresholds for different tasks and their corresponding maximum Youden’s index are shown in Table 5.

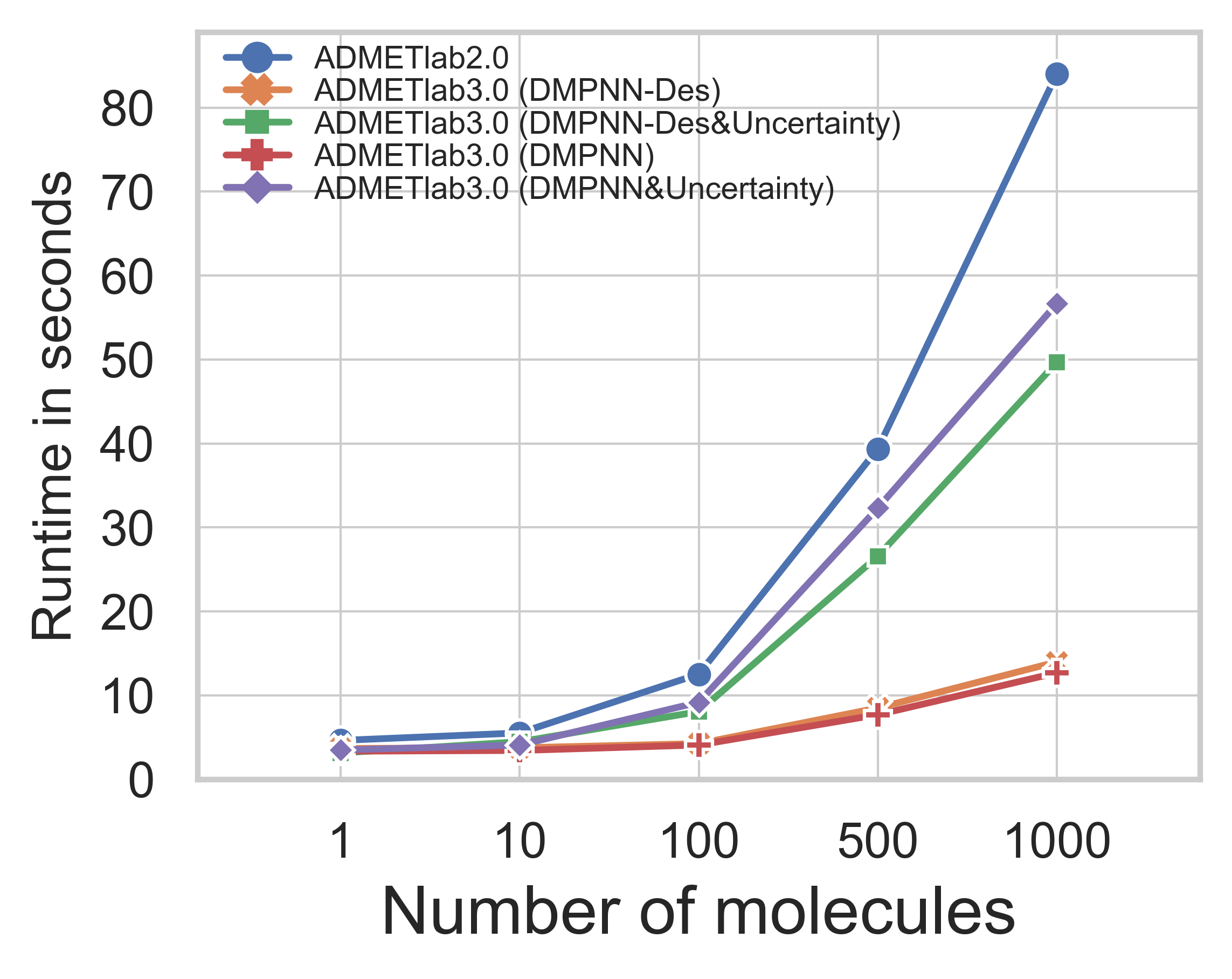


Figure 6. Runtime analysis in seconds for submissions of 1 to 1000 molecules for ADMETlab2.0, ADMETlab3.0 (DMPNN-Des), ADMETlab3.0 (DMPNN-Des & Uncertainty), ADMETlab3.0 (DMPNN), ADMETlab3.0 (DMPN & Uncertainty).

Reference:  
1. Yang,K., Swanson,K., Jin,W., Coley,C., Eiden,P., Gao,H., Guzman-Perez,A., Hopper,T., Kelley,B., Mathea,M., *et al.* (2019) Analyzing Learned Molecular Representations for Property Prediction. *J. Chem. Inf. Model.*, **59**, 3370–3388.  
  
https://doi.org/10.1021/acs.jcim.9b00237

2. Heid,E., Greenman,K.P., Chung,Y., Li,S.-C., Graff,D.E., Vermeire,F.H., Wu,H., Green,W.H. and McGill,C.J. (2023) Chemprop: Machine Learning Package for Chemical Property Prediction Chemistry.  
  
https://doi.org/10.26434/chemrxiv-2023-3zcfl

3. Begoli,E., Bhattacharya,T. and Kusnezov,D. (2019) The need for uncertainty quantification in machine-assisted medical decision making. *Nat. Mach. Intell.*, **1**, 20–23.  
  
https://doi.org/10.1038/s42256-018-0004-1

4. Amini,A., Schwarting,W., Soleimany,A. and Rus,D. (2020) Deep evidential regression. *Adv. Neural Inf. Process. Syst.*, **33**, 14927–14937.

5. Gal,Y. and Ghahramani,Z. (2016) Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of The 33rd International Conference on Machine Learning*. PMLR, pp. 1050–1059.

6. Wen,M. and Tadmor,E.B. (2020) Uncertainty quantification in molecular simulations with dropout neural network potentials. *Npj Comput. Mater.*, **6**, 1–10.  
  
https://doi.org/10.1038/s41524-020-00390-8

7. Tong,X., Wang,D., Ding,X., Tan,X., Ren,Q., Chen,G., Rong,Y., Xu,T., Huang,J., Jiang,H., *et al.* (2022) Blood–brain barrier penetration prediction enhanced by uncertainty estimation. *J. Cheminformatics*, **14**, 44.  
  
https://doi.org/10.1186/s13321-022-00619-2

8. Wang,S. and Manning,C. (2013) Fast dropout training. In *Proceedings of the 30th International Conference on Machine Learning*. PMLR, pp. 118–126.

9. Dolezal,J.M., Srisuwananukorn,A., Karpeyev,D., Ramesh,S., Kochanny,S., Cody,B., Mansfield,A.S., Rakshit,S., Bansal,R., Bois,M.C., *et al.* (2022) Uncertainty-informed deep learning models enable high-confidence predictions for digital histopathology. *Nat. Commun.*, **13**, 6572.  
  
https://doi.org/10.1038/s41467-022-34025-x