A Survey on Uncertainty Quantification Methods for Deep Learning

WENCHONG HE, The University of Florida, USA
ZHE JIANG*, The University of Florida, USA
TINGSONG XIAO, The University of Florida, USA
ZELIN XU, The University of Florida, USA
YUKUN LI, Tufts University, USA

Deep neural networks (DNNs) have achieved tremendous success in making accurate predictions in computer vision, natural language processing, as well as science and engineering domains. However, it is also well-recognized that DNNs sometimes make unexpected, incorrect, but overconfident predictions. This can cause serious consequences in high-stakes applications, such as autonomous driving, medical diagnosis, and disaster response. Uncertainty quantification (UQ) aims to estimate the confidence of DNN predictions in addition to prediction accuracy. In recent years, many UQ methods have been developed for DNNs. It is of great practical value to systematically categorize these UQ methods and compare their advantages and disadvantages. However, existing surveys mostly focus on categorizing UQ methodologies from the perspective of neural network architecture or Bayesian methods and ignore the source of uncertainty that each methodology can incorporate, making it difficult to select an appropriate UQ method in practice. To fill the gap, this paper presents a systematic taxonomy of UQ methods for DNNs based on the types of uncertainty sources (data uncertainty versus model uncertainty). We summarize the advantages and disadvantages of methods in each category. We show how UQ methodologies can be used in machine learning problems (e.g., active learning, robustness to out-of-distribution samples, and deep reinforcement learning). We also identify several future research directions, such as UQ for large language models (LLMs), UQ for DNNs in scientific simulations, and UQ for DNNs with structured outputs.

CCS Concepts: • **Computing methodologies → Uncertainty quantification**; **Machine learning approaches**; **Knowledge representation and reasoning**; • **Applied computing → Physical sciences and engineering**; • **Information systems → Data mining**.

Additional Key Words and Phrases: Deep learning, uncertainty quantification, data uncertainty, model uncertainty, trustworthy AI.

ACM Reference Format:
Wenchong He, Zhe Jiang, Tingsong Xiao, Zelin Xu, and Yukun Li. 2023. A Survey on Uncertainty Quantification Methods for Deep Learning. *ACM Comput. Surv.* 37, 4, Article 111 (August 2023), 35 pages. https://doi.org/XXXXXXX.XXXXXXX

*Contact author: Zhe Jiang, zhe.jiang@ufl.edu

Authors' addresses: Wenchong He, whe2@ufl.edu, The University of Florida, Gainesville, FL, USA, 32611; Zhe Jiang, zhe.jiang@ufl.edu, The University of Florida, Gainesville, FL, USA, 32611; Tingsong Xiao, xiaotingsong@ufl.edu, The University of Florida, Gainesville, FL, USA, 32611; Zelin Xu, zelin.xu@ufl.edu, The University of Florida, Gainesville, FL, USA, 32611; Yukun Li, yukun.li@tufts.edu, Tufts University, Medford, MA, USA, 02155.

0360-0300/2023/8-ART111 $15.00
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Deep neural network (DNN) models have achieved remarkable success in computer vision, natural language processing, and engineering domains [33, 91]. Most existing DNN models can be viewed as deterministic functions mapping input features to target predictions through hierarchical representation learning [13]. While these DNN models often achieve high overall accuracy, they are also known to sometimes make unexpected, incorrect, and overconfident predictions, especially in a complex real-world environment [125]. This can have serious consequences in high-stakes applications, such as autonomous driving [27], medical diagnosis [11], and disaster response [3]. In this regard, a DNN model should be aware of what it does not know. For example, in the medical domain, when a DNN-based automatic diagnosis system encounters uncertain cases, it should refer the patient to a medical expert for more in-depth analysis to avoid fatal mistakes. In an autonomous vehicle, if a DNN model knows in what scenarios it tends to make mistakes in estimating road conditions (e.g., bad weather), it can warn the driver to take over and avoid potential crashes.

Recognizing what a DNN model does not know requires assigning appropriate uncertainty scores to its predictions, also called *uncertainty quantification* (UQ). Uncertainty in DNNs may come from different types of sources, including data uncertainty and model uncertainty [169]. Data uncertainty (also aleatoric uncertainty) is an inherent property of the data, which originates from the randomness and stochasticity of the data (e.g., sensor noises) or conflicting evidence between the training labels (e.g., class confusion). Data uncertainty is often considered irreducible because we cannot reduce it by adding more training samples. On the other hand, model uncertainty (also epistemic uncertainty) comes from a lack of evidence or knowledge during model training or inference, e.g., limited training samples, sub-optimal DNN model architectures or parameter learning algorithms, and out-of-distribution (OOD) test samples.

Researchers have recently developed a growing number of UQ methods for DNN models. As shown in Fig. 1, existing surveys of UQ methods for DNNs Researchers have recently developed a growing number of UQ methods for DNN models. Specifically, [50] categorizes existing UQ methods based on their types of DNN model architectures, including Bayesian neural networks, ensemble models, and single model architecture, without discussing the connection between DNN model architectures and the types of uncertainty they address. Other surveys only focus on the Bayesian perspective. For example, [108] provides a comprehensive review of Bayesian neural networks for UQ but overlooks methods from a frequentist perspective (e.g., prediction interval, ensemble methods). [1] covers the ensemble methods and other frequentist methods, but it does not compare their advantages and disadvantages. To the best of our knowledge, existing surveys on UQ methods often overlook the types of uncertainty sources these methods address. This perspective is important for selecting the appropriate UQ methods for different applications where one type of uncertainty source dominates others.

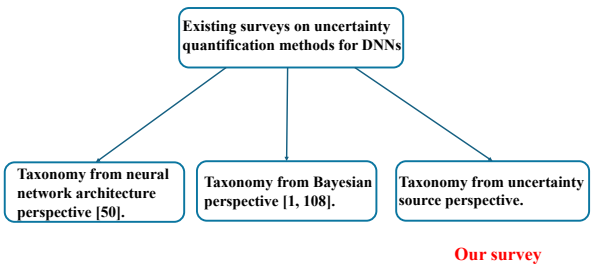


Fig. 1. Existing survey on UQ methods for DNNs.

To fill the gap, we provide the first survey of UQ methods for DNNs from the perspective of uncertainty sources. Specifically, we create a systematic taxonomy for DNN uncertainty quantification methodologies based on the types of uncertainty sources they incorporate. We summarize the characteristics of different methods in their technical approaches and compare their advantages and disadvantages in addressing different types of uncertainty sources. We also connect the taxonomy to several major deep learning topics where UQ methods are critical, including OOD detection, active learning, and deep reinforcement learning. Finally, we identify research gaps and suggest several directions for future research. The overall structure of this survey is as follows:

- Section 2 defines two different types of uncertainty (sources), i.e., data uncertainty and model uncertainty, in the supervised learning setting. This will serve as a foundation for various UQ methods in the survey.
- Section 3 highlights the practical applications of uncertainty quantification (UQ) for deep learning, focusing on how UQ applies to various real-world problems, such as medical diagnosis, geosciences, and transportation.
- Section 4 presents a taxonomy of UQ methods for deep learning based on the types of uncertainty sources they capture, including data uncertainty, model uncertainty, and both.
- Section 5 discusses several key machine learning problems (active learning, OOD detection, reinforcement learning) where UQ plays a significant role.
- Section 6 discusses several future research directions, including UQ for large language models (LLMs), UQ for DNNs in scientific simulations, UQ for DNNs with structured outputs (e.g., spatiotemporal data and graphs), and combining UQ with explainability.

2 TYPES OF UNCERTAINTY SOURCE

This section first briefly reviews the mathematical formulation of supervised learning. Based on that, we define two sources of uncertainty, i.e., data uncertainty and model uncertainty, and describe their representation.

2.1 Preliminaries of Supervised Learning

Given training data $\mathcal{D}_{tr} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Y}$, $\mathcal{X} \subseteq \mathbb{R}^d$ is the input sample feature space, and $\mathcal{Y}$ is the target variable space, whereby $\mathcal{Y} = \{\omega_1, ..., \omega_k\}$ for a classification problem with k classes, and $\mathcal{Y} \subseteq \mathbb{R}$ for a regression problem. Each training instance is assumed to be independent and identically distributed (i.i.d.) from some unknown probability distribution $p(\boldsymbol{x}, y)$ on the space $\mathcal{X} \times \mathcal{Y}$. Given a hypothesis space $\mathcal{H}$ consisting of hypotheses $h : \mathcal{X} \to \mathcal{Y}$ and a loss function l that measures the discrepancy between prediction and ground-truth, a learning problem aims to find the best hypothesis in the hypothesis space that minimizes the loss [66]:

$$h^* = \arg\min_{h \in \mathcal{H}} R(h), \quad \text{where} \quad R(h) = \int l(y, h(\boldsymbol{x}))p(\boldsymbol{x}, y)d\boldsymbol{x}dy. \tag{1}$$

In practice, the model is learned by minimizing the empirical risk [110], defined as the average loss over the training data $\mathcal{D}_{tr}$:

$$\tilde{h} = \arg\min_{h \in \mathcal{H}} R_{emp}(h), \text{ where } R_{emp}(h) = \frac{1}{|\mathcal{D}_{tr}|} \sum_{(\boldsymbol{x}_i, y_i) \in \mathcal{D}_{tr}} l(y_i, h(\boldsymbol{x}_i)). \tag{2}$$

2.2 Model uncertainty

2.2.1 Sources of model uncertainty. Model uncertainty (a.k.a. epistemic uncertainty) represents the uncertainty in a model's predictions related to the imperfect model training process. It is reducible given more training data. There are several common types of model uncertainty: uncertainty

in the choice of model family, uncertainty in model parameter learning, and uncertainty due to different sample distributions between model training and model inference (e.g., out-of-distribution samples). These types of model uncertainty are illustrated in Fig. 2, where $\mathcal{F}$ denotes the entire hypothesis space, f_1^* and f_2^* are the theoretical optimal hypotheses within $\mathcal{F}$ (based on Eq. 1) for two different sample distributions $p_1(\boldsymbol{x}, y)$ and $p_2(\boldsymbol{x}, y)$, respectively. That is, $f_i = \arg\min_{h\in\mathcal{F}} \int l(y, h(\boldsymbol{x}))p_i(\boldsymbol{x}, y)d\boldsymbol{x}dy$ for $i = 1, 2$. $\mathcal{H}$ is the sub-hypothesis space for one particular model architecture and set of hyperparameters (e.g., a specific transformer architecture). h^* is the theoretical optimal solution within $\mathcal{H}$ for a sample distribution $p_1(\boldsymbol{x}, y)$ based on Eq. 1. $\tilde{h}$ is the empirical solution within $\mathcal{H}$ that is learned by an optimizer based on a particular training data $\mathcal{D}_{tr}$ drawn from the population distribution $p_1(\boldsymbol{x}, y)$ (see Eq. 2).

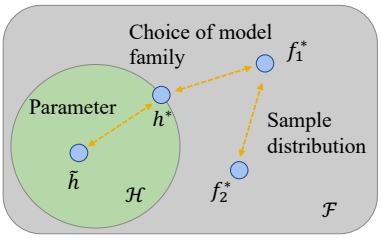


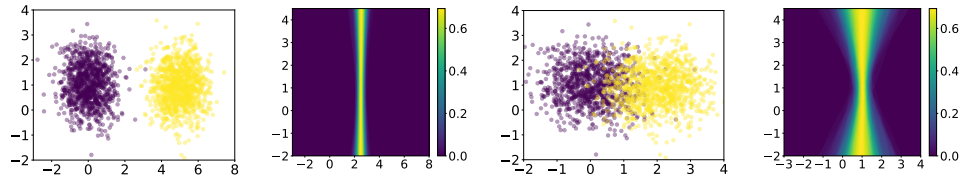
Fig. 2. Visualization on various model uncertainty sources.

Table 1 summarizes the different sources of model uncertainty in the supervised learning framework. The first type, the choice of model family, is due to the lack of knowledge of which type of model architecture is the most suitable. Because of this, the theoretical optimal solution h^* within $\mathcal{H}$ (assuming a particular model architecture) is different from the theoretical optimal f_1^*. It is related to the "bias" part in the bias-variance decomposition [55]. Second, due to limited training data or imperfect parameter learning algorithms, the learned model $\tilde{h}$ based on the empirical loss may exhibit variations and deviate from the theoretically optimal solution h^* within $\mathcal{H}$. This leads to model uncertainty related to model parameter learning. It is related to the "variance" part in the bias-variance decomposition. Third, model uncertainty can be related to the different sample distributions between model training and inference. For example, there may be two different sample distributions $p_1(\boldsymbol{x}, y)$ and $p_2(\boldsymbol{x}, y)$. Their theoretical optimal solutions f_1^* and f_2^* are different. Because of this, a model learned from training samples following $p_1(\boldsymbol{x}, y)$ will contain uncertainty in its inference on a sample drawn from $p_2(\boldsymbol{x}, y)$, i.e., out-of-distribution (OOD) samples. Another relevant scenario is that when a model makes predictions on a test sample that is far away from other training samples (or surrounded by sparse training samples) in the feature space $\mathcal{X}$ (see Fig. 6), the prediction tends to have higher uncertainty. This scenario is also relevant to the OOD case since a test sample that is far away from other training samples is more likely to be an OOD sample. Note that our definition of the three types of model uncertainty may overlap. For instance, model uncertainty due to the lack of training samples near a test sample can also be considered model parameter learning uncertainty.

Table 1. Comparison of different types of model uncertainty in the supervised learning setting

Model uncertainty sources	**Corresponding notation in supervised learning**
Choice of model family	Optimal solution h^* within $\mathcal{H}$ does not align with theoretical optimal f^* in $\mathcal{F}$
Model parameter learning	Learned solution $\tilde{h}$ does not align with optimal h^* in $\mathcal{H}$
Different sample distributions in learning and inference	Theoretical optimum f_1^* and f_2^* mismatch under different sample distribution $p(\boldsymbol{x}, y)$

2.2.2 Model uncertainty representation. As discussed above, the sources of model uncertainty can arise from different aspects: the choice of model hyper-parameters, model parameter learning, and different sample distributions in learning and inference. In general, there are various ways to



(a) Sample distribution with separable class boundary.

(b) Entropy of sample distribution in (a).

(c) Sample distribution with ambiguous class boundary.

(d) Entropy of sample distribution in (c).

Fig. 3. Data uncertainty visualization examples (Different colors represent samples in different classes).

represent model uncertainty from each type. First, uncertainty in model parameter learning arises from suboptimal parameter optimization. To account for this uncertainty source, one approach is through a Bayesian neural network (BNN) [74]. A BNN assumes a prior over the model parameters and aims to infer the posterior distribution of the model parameters to reflect the parameter uncertainty. This provides a theoretical foundation for model uncertainty. Second, uncertainty arising from the choice of model hyper-parameters is due to the inductive bias in choosing a sub-hypothesis space (e.g., a particular DNN architecture) [157, 166, 173]. It can be estimated with ensembles of different neural network architectures (deep ensembles) [89]. The intuition is to construct an ensemble of neural network architectures, each of which is trained separately. The predictions of the ensemble on an input form a distribution over the target variable. Thus, the variance of the target variable predictions can be used to estimate the prediction uncertainty. The third type of uncertainty arises from differences in sample distributions, which are caused by the mismatch between the distribution of the training dataset and that of a test sample. Capturing this type of uncertainty requires learning meaningful embeddings that reflect sample distances. More details are discussed in Section 4.

2.3 Data uncertainty

2.3.1 Source of data uncertainty. Data uncertainty (a.k.a aleatoric uncertainty) arises from inherent data randomness, noise, or class confusion (i.e., the same feature value can correspond to different classes in the sample distribution). It is irreducible even with more training data [169]. Randomness or noise in data can arise in data acquisition due to instrument errors, data transmission errors, and inappropriate data storage and formatting [58]. For example, for spatiotemporal data collected from various space and airborne platforms (e.g., CubeSat, UAVs), the data uncertainty may result from the sensor errors associated with the data acquisition devices and the fact that data are acquired in a digital format (which is discrete in nature) [26] even though the underlying process is continuous.

2.3.2 Data uncertainty representation. Consider a training dataset $\mathcal{D}_{tr}$ drawn from the distribution $p(\boldsymbol{x}, y)$. Several techniques exist for representing data uncertainty to account for the inherent randomness in the mapping from $\boldsymbol{x}$ to y. In the context of a discriminative classification task, one method to represent the uncertainty of the class variable, given a specific input $\boldsymbol{x}$ is the maximum class probability $\max_y p(y|\boldsymbol{x})$. Another approach uses the entropy of the condition class distribution $p(y|\boldsymbol{x})$, which captures the randomness of the class distribution due to class confusion. For high-dimensional structured samples, deep generative models can be employed to learn the complex underlying distribution of the data and quantify uncertainty.

Data uncertainty arises from natural variability in data (for regression) and class confusion (for classification). Consider the toy distribution in Fig. 3 as an example for classification, which consists

of two normally distributed clusters. Each cluster (color) represents a separate class. The dataset in Fig. 3 (a) has a sharper class boundary, indicating lower data uncertainty. The entropy of most samples is low except for those near the class boundary, as shown in Fig. 3 (b). In contrast, the dataset in Fig. 3 (c) exhibits more confusion between the two classes, corresponding to higher data uncertainty as shown in Fig. 3 (d).

Beyond class confusion, data uncertainty can also arise from inherent noise (variability) in the data generation or collection process. For example, in a regression problem, the observations can be represented by: $y = f(\boldsymbol{x}) + \epsilon(\boldsymbol{x})$, where $f(\boldsymbol{x})$ is the true data generation function, and $\epsilon(\boldsymbol{x})$ represents the measurement noise. There are two classes of noise: homoscedastic and heteroscedastic noise [78]. Homoscedastic noise assumes constant noise variance across all the $\boldsymbol{x}$ inputs. Heteroscedastic noise, on the other hand, models the observation noise as a function of the input $\epsilon(\boldsymbol{x}) \sim p(\epsilon|\boldsymbol{x})$ (e.g., heteroscedastic Gaussian noise). The heteroscedastic noise model is useful in the case where the noise level varies for different samples.

In summary, we have described the sources and representations of both model and data uncertainty. As Fig. 4 shows, data uncertainty arises from the inherent properties of the given data, while model uncertainty stems from issues such as misspecification of model architectures, parameters, and the differences in sample distributions. Depending on the nature of the application, the predominant source of uncertainty may vary.

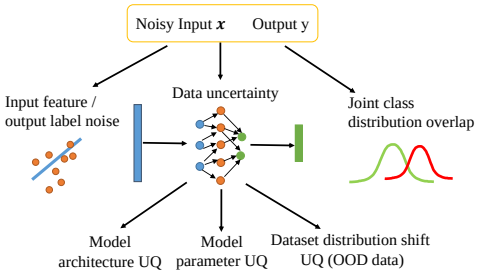


Fig. 4. Different types of uncertainty source.

3 APPLICATION DOMAINS

In this section, we discuss several application domains of uncertainty quantification for deep learning models. For each application, we discuss the motivation for developing uncertainty-aware models, the source of uncertainty, and the challenges associated with uncertainty quantification. The applications in medical diagnosis, geoscience, transportation and natural language processing are discussed below. Additional applications in Biochemistry engineering and engineering design are discussed in appendix.

Medical diagnosis: DNN models have achieved tremendous success in various medical applications, including medical imaging, clinical diagnosis support, and treatment planning [116]. However, a critical concern is that deep learning models tend to be over-confident even for a wrong prediction [101], which can lead to serious consequences. Thus, it is essential to estimate the prediction uncertainty (confidence). Both data uncertainty and model uncertainty exist in medical problems. Data uncertainty arises from noisy measurements from medical devices, ambiguous class labels (e.g., non-consensus tumor boundary annotations between different radiologists), and registration errors between medical imagery taken at different times or from different devices [53]. Model uncertainty also exists because patient cases in the test cases may not be well-represented in the training set. There are several challenges in developing UQ methods. First, medical data contains diverse sources of noise and uncertainty. Second, Interpretability in uncertainty quantification is important, but it remains an unsolved issue in medical problems. Existing uncertainty-aware deep learning models in medical domains can be categorized into those related to medical imaging and those for non-medical imaging applications [101]. In medical imaging, deep learning is often used for segmentation or classification of magnetic resonance imaging, ultrasound, and coherence tomography imagery [39]. These studies often focus on data uncertainty due to ambiguous labels [123], or image registration uncertainty [24]. Non-medical imaging applications are mostly related

to clinical diagnosis support and treatment planning from Electronic Health Records. The presence of significant variability in personalized predictions [38] requires a model to capture prediction uncertainty.

Geoscience: With advances in GPS and remote sensing technologies, a growing volume of spatiotemporal data is being collected from spaceborne, airborne, seaborne, and terrestrial platforms [136]. Emerging spatiotemporal big data, increased computational power (GPUs), and recent advances in deep learning technologies provide unique opportunities to advance our knowledge of the Earth system [136]. For example, deep learning has been used to predict river flow and temperature [68] and hurricane tracks [81]. Uncertainty quantification for deep learning is important in geoscience because of the high-stakes decision-making involved (e.g., evacuation planning based on hurricane tracking with a "cone of uncertainty"). Several challenges arise from the unique characteristics of spatiotemporal data. First, spatiotemporal data exhibit various spatial, temporal, and spectral resolutions and diverse sources of noise and errors (e.g., sensor noise, obstacles, atmospheric effects in remote sensing signals [97], and GPS errors). Second, spatial registration errors and uncertainties may arise when co-registering different layers of geospatial data into the same spatial reference system [59]. Third, spatiotemporal data are heterogeneous, i.e., the data distribution often varies across different regions or time periods [72]. As a result, a deep learning model trained in one region or time period may not generalize well to another. This issue is particularly significant when spatial observations are sparsely distributed, leading to uncertainty in inferring values at other locations in continuous space.

Transportation: Deep learning applied transportation data (e.g., ground sensors and video cameras on the road) provides unique opportunities to monitor traffic conditions, analyze traffic patterns, and improve decision-making. For instance, temporal graph neural networks are used to predict traffic flows (e.g., congestion or accidents), and incorporating physical principles into neural network modeling further enhances traffic modeling performance [67]. Deep learning plays a critical role in autonomous driving (e.g., using LiDAR sensors and optical cameras to detect road lanes, other vehicles, or pedestrians). Uncertainty quantification for AI in transportation is challenging due to temporal dynamics, the complexity of road environment, and the existence of noise and uncertainty (e.g., omission, sparse sensor coverage, errors, or inherent biases). For example, highly crowded events can disrupt normal traffic flows on road networks. Existing studies on trajectory prediction uncertainty consider the data uncertainty due to sparse or insufficient training data, and erroneous or missing measurements from signal loss [106]. Other studies consider complex environmental factors such as extreme weather conditions [154]. Uncertainty in short-term traffic status forecasting (e.g., volume, travel speeds, and occupancy) is related to the stochastic environment and model training [153], while uncertainty in long-term traffic modeling stems from exogenous factors affecting traffic flow (e.g., rainstorms and snowstorms) [96].

Natural Language Processing: The advancement of pre-trained language models (PLMs) has revolutionized language processing by addressing various tasks in a unified manner (e.g., machine translation, sentiment analysis, speech recognition) [94]. Although natural language processing (NLP) has made remarkable strides with the emergence of large language models (LLMs), these models are prone to hallucinations (i.e., generating misleading or fabricated content) [164]. Thus, uncertainty quantification plays a critical role in improving the trustworthiness of LLMs. However, quantifying uncertainty in LLMs presents significant challenges. First, uncertainty can arise from various sources related to both data and models. Data uncertainty stems from ambiguities, noise (e.g., out-of-vocabulary words or distractions), and semantic complexities in the input language. Model uncertainty occurs when the model lacks the specific knowledge required for out-of-distribution input queries, leading to arbitrary responses. Second, the large vocabulary space complicates the direct assessment of confidence through probability likelihood, and relying solely on prediction

logits can lead to overconfidence [57]. Therefore, capturing uncertainty in the semantic space is essential due to the semantic similarity among different answers. Third, aligning uncertainty with actual correctness (calibration) is crucial for trustworthy applications of LLMs.

4 A TAXONOMY OF UQ METHODOLOGIES FOR DNNS

In this section, we provide a new taxonomy (Fig. 5) of UQ methods for DNNs based on the type of uncertainty sources: model uncertainty, data uncertainty, and the combination of the two. We discuss the underlying intuitions of specific methods in each category and compare their pros and cons.

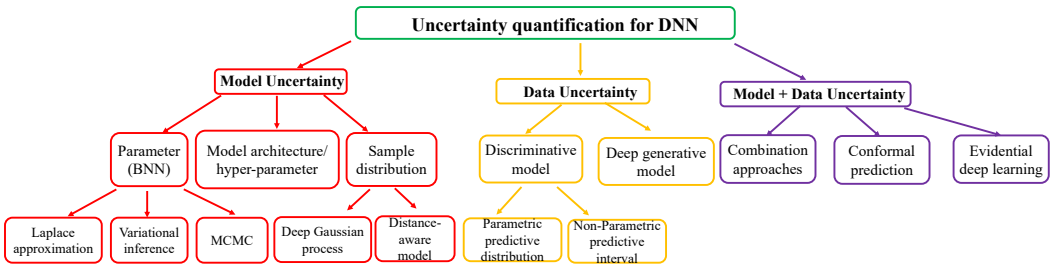


Fig. 5. A taxonomy for existing literature on UQ for DNN.

4.1 Model Uncertainty

This subsection reviews the existing methods for model uncertainty in DNNs. We categorize these methods into three subcategories: Bayesian neural network, ensemble models, and sample distribution-related models. We now introduce each subcategory.

4.1.1 Bayesian Neural Networks. From a frequentist point of view, there exists a single set of parameters $\boldsymbol{\theta}^*$ that best fit the DNN model, where $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\mathrm{Y}, f(\mathrm{X}, \boldsymbol{\theta}))$ and $\mathcal{L}$ is the loss function. However, the point estimation of DNN parameters can be over-fitting and overconfident [144]. In contrast, the Bayesian neural network (BNN) imposes a prior on the neural network parameters $p(\boldsymbol{\theta})$ and learns the posterior distribution of these parameters $p(\boldsymbol{\theta}|\mathrm{X}, \mathrm{Y}) = \frac{p(\mathrm{Y}|\mathrm{X},\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathrm{Y}|\mathrm{X})}$. This term represents the posterior distribution of model parameters conditioned on the training dataset. This distribution reflects the extent to which our model can capture patterns in the training data. Assuming a Gaussian distribution for the model parameters, a larger variance in the distribution indicates greater uncertainty in the model. Such uncertainty can be caused by a limited amount of training data. For inference on new samples $\boldsymbol{x}^*$, we can marginalize out the model parameters as follows:

$$p(y^*|x^*, \mathrm{X}, \mathrm{Y}) = \int p(y^*|\boldsymbol{x}^*, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathrm{X}, \mathrm{Y}) d\boldsymbol{\theta}. \tag{3}$$

The uncertainty of a test sample is reflected by the variance of its prediction distribution. Although BNN can theoretically quantify total uncertainty by modeling both prediction distribution $p(y^*|\boldsymbol{x}^*, \boldsymbol{\theta})$ (data uncertainty) and parameter posterior distribution $p(\boldsymbol{\theta}|\mathrm{X}, \mathrm{Y})$ (model uncertainty), most existing work simplifies the analysis by treating the prediction $p(y^*|\boldsymbol{x}^*, \boldsymbol{\theta})$ as deterministic. Therefore, these BNN methods primarily capture parameter uncertainty rather than total uncertainty.

However, the parameter posterior distribution is analytically intractable and lacks a closed-form solution. An approximation must be made. Various approaches have been proposed to estimate the

posterior of the neural network parameters in a simpler and tractable form. Some approximation methods define a parameterized class of distributions, Q, from which they select an approximation $q_\phi(\boldsymbol{\theta})$ for the posterior. For example, Q can be the set of all factorized Gaussian distributions, and ϕ is the parameters of the mean and diagonal variance. The distribution $q_\phi(\boldsymbol{\theta}) \in Q$ is selected according to some optimization criteria to approximate the posterior. Two popular methods for optimization are *variational inference* [17] and *Laplace approximation* [45]. Instead of approximating the posterior analytically, another approach is to address this problem using Monte Carlo sampling, specifically *Markov Chain Monte Carlo sampling*. More details are in the Appendix.

Monte-Carlo (MC) dropout: The MC dropout approach [46] is currently among the most popular methods for DNN uncertainty quantification due to its simplicity and ease of implementation. The main idea is that the optimization of a neural network with a dropout layer can be equivalent to approximating a BNN with variational inference on a parametric Bernoulli distribution [46]. Uncertainty estimation can be obtained by computing the variance of multiple stochastic forward predictions with different dropout masks (switching off some neurons' activations). The average predictions with various weights dropout can be interpreted as approximating the integration over the model's weights (as Eq.3) whose variational distribution follows the Bernoulli distribution. MC dropout offers several advantages. First, it requires minimal modification to the existing DNN architecture design, allowing for straightforward implementation in practice. Second, it mitigates the problem of representing uncertainty by sacrificing model accuracy as it only affects the inference. However, although there is theoretical intuition for the probabilistic interpretation of MC dropout from a variational approximation perspective, MC dropout tends to be less calibrated than other UQ methods on many uncertainty benchmark datasets [57].

In summary, we review several approximation methods for BNNs designed to reduce computational and memory demands. These methods capture model uncertainty associated with the parameters. However, the need for approximation may lead to less accurate uncertainty estimates, and in practice, these methods remain computationally intensive.

4.1.2 Ensemble models. Ensemble models combine multiple neural networks to form an output distribution, where the variability of the distribution quantifies model uncertainty. To capture model uncertainty from various sources, several strategies for constructing ensembles have been adopted. The first strategy involves bootstrapping [89, 90]. This approach involves random sampling from the original dataset with replacement. An ensemble of neural networks is then constructed, with each model trained on different bootstrapped samples. After training, inference is performed by aggregating the ensemble predictions, with uncertainty obtained from the prediction variance (for regression) or average entropy (for classification). The second strategy is to construct different neural network architectures by varying the number of layers, hidden neurons, and types of activation functions [105, 158]. This strategy can account for the uncertainty from model misspecification. Other strategies involve different parameter initializations and random shuffling of datasets. This approach is better than the bootstrap strategy since more samples can be utilized for each model. The third type is the hyperensemble approach [157]. This approach constructs ensembles with different hyper-parameters, such as learning rate, optimization strategy, and training strategy.

Although ensemble models are relatively simple to implement, they have several limitations. First, they have a high computational cost as they require training multiple independent networks and maintaining all networks in memory during inference. Second, model diversity is required to ensure accurate uncertainty estimation.

4.1.3 Sample distribution-related methods. In Section 3.1.1, we described model uncertainty related to sample distribution as a distinct category. It further includes two cases: (1) test samples and training samples follow different distributions (out of distribution); (2) a test sample is far from

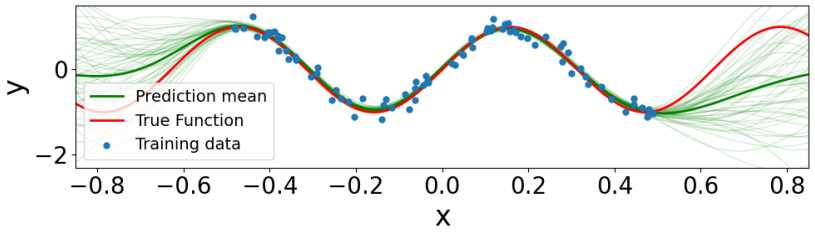


Fig. 6. Gaussian Process inference example: green lines are the prediction sample distribution.

other training samples (or is surrounded by sparse training samples) in the feature space. Here, we focus only on methods for case (2), i.e., sample density (distance)-aware neural networks, and do not specifically discuss out-of-distribution (OOD) methods, as BNNs and ensemble methods can also be used for OOD-related uncertainty. Methods for addressing OOD problems will be reviewed in Section 5. Existing methods can be grouped into two categories: Gaussian process hybrid neural network and distance-aware neural network.

Background of Gaussian Process: A Gaussian process (GP) is a type of stochastic process where any finite collection of random variables follows a multivariate Gaussian distribution [159]. Given a set of points $\{\boldsymbol{x}_i\}_{i=1}^n$, a GP defines a prior over functions $y_i = f(\boldsymbol{x}_i)$, and assumes the $p(y_1, ..., y_n)$ follows the Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}(\boldsymbol{x}), \Sigma(\boldsymbol{x}))$, where $\boldsymbol{x} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_n)$, $\boldsymbol{\mu}(\boldsymbol{x})$ is the mean function, and $\Sigma(\boldsymbol{x})$ is the covariance function based on $\Sigma_{ij} = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$. κ is a positive definite kernel function (e.g., radial basis function) that measures the similarity between pairs of input samples and controls the smoothness of the GP. For GP inference, given a new sample $\boldsymbol{x}^*$, the joint distribution between prediction for the new sample y^* and the target variable y of the training samples is shown in Eq. 4, where $\mathbf{K}_n$ is the covariance matrix between the n training samples, $\mathbf{K}_{\boldsymbol{x}}$ is the covariance between the test sample and training samples, and K^* is the prior variance of the test sample.

$$\begin{pmatrix} \mathbf{y} \\ y^* \end{pmatrix} = \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu} \\ \mu^* \end{pmatrix}, \begin{pmatrix} \mathbf{K}_n & \mathbf{K}_{\boldsymbol{x}} \\ \mathbf{K}_{\boldsymbol{x}}^T & K^* \end{pmatrix} \right). \tag{4}$$

The prediction for a new test sample $\boldsymbol{x}^*$ is obtained by computing the posterior distribution conditioned on the training data $\mathcal{D}_{\text{tr}}$, given by Eq. 5. GP inference yields lower uncertainty when test samples are in regions where training samples are abundant (higher sample density as shown in the middle part of Fig. 6), otherwise resulting in higher uncertainty (boundary part of Fig. 6). Note that although GP belongs to Bayesian methods, it considers the uncertainty source differently from the BNN methods. The GP methods capture uncertainty related to sample sparsity. Thus, we consider GP to be a separate sub-category.

$$p(y^*|x^*, \mathcal{D}_{\text{train}}, \boldsymbol{\theta}) = \mathcal{N}(y|\mathbf{K}_{\boldsymbol{x}}^T \mathbf{K}_n^{-1} \boldsymbol{y}, K_{x^*} - \mathbf{K}_{\boldsymbol{x}}^T \mathbf{K}_n^{-1} \mathbf{K}_{\boldsymbol{x}}). \tag{5}$$

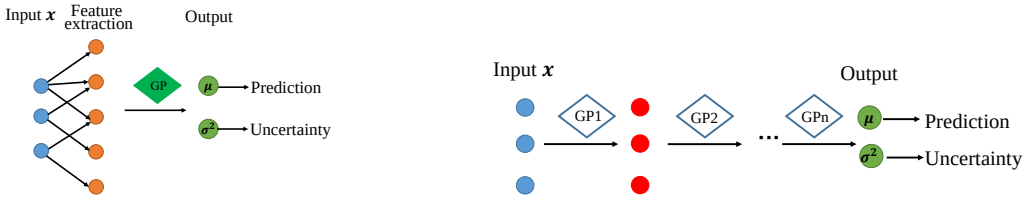
Sparse Gaussian process: Though GP has a sound theoretical framework for uncertainty estimation, it is computationally unscalable for large datasets because inverting the covariance matrix requires $O(n^3)$ time complexity (n is the total number of training samples). To mitigate this bottleneck, many methods [139, 145] attempt to make a sparse approximation to the full GP to reduce the computational complexity to $O(m^2 n)$ (m is the number of inducing variables, and $m \ll n$). The inducing variables can be anywhere in the input domain, and are not constrained to be a subset of the training data and are represented asinput-output pairs $\{\hat{\boldsymbol{x}}_i, \hat{y}_i\}_{i=1}^m$. Thus, inverting the original covariance matrix $\mathbf{K}_n$ can be replaced with a low-rank approximation from the inducing variables, which only requires the inversion of an $m \times m$ matrix $\mathbf{K}_m$. Then, the question becomes how to select

the m best-inducing variables to be representative of the training dataset. Common approaches assume that the best representative inducing variables are those that maximize the likelihood of the training dataset [139]. Subsequently, the location of inducing variables and the hyper-parameters of GP are optimized simultaneously through maximum likelihood. The training data likelihood can be obtained by marginalizing the inducing variables on the joint distribution of the training dataset and inducing variables.

Besides the computational challenge, another limitation of GP is that the joint Gaussian distribution assumption on the target variables limits the model's capability to capture diverse relationships among instances within large datasets. Additionally, GP relies heavily on the kernel function to compute the similarity between samples by transforming input features into a high-dimensional manifold. However, for high-dimensional structured data, it is challenging to construct appropriate kernel functions to extract hierarchical features for computing similarity between samples. To address these limitations, two research areas have been proposed: *deep kernel learning* and *Deep (Compositional) Gaussian Process*.

Gaussian Process Hybrid Neural Network

Deep kernel learning [160] aims to combine the structured feature learning capability of DNN with a GP to learn more flexible representations. The motivation is that DNN can automatically discover meaningful representations from high-dimensional data, which could alleviate the fixed kernel limitations of GP and improve its expressiveness. Specifically, the deep kernel learning approach transforms the kernel $\mathbf{K}_{\boldsymbol{\theta}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ to $\mathbf{K}_{\boldsymbol{\theta}}(g(\boldsymbol{x}_i; \boldsymbol{w}), g(\boldsymbol{x}_j, \boldsymbol{w}))$, where $g(\cdot; \boldsymbol{w})$ is the neural network parameterized with $\boldsymbol{w}$ and $\mathbf{K}_{\boldsymbol{\theta}}$ is the base kernel function (e.g., radial basis function) of GP. Deep learning transformation can capture the non-linear and hierarchical structure in high-dimensional data. The GP with the base kernel is applied on the final layer of DNN and makes inferences based on the learned latent features, as shown in Fig. 7 (a). The idea has been successfully applied to spatio-temporal crop yield prediction, where GP plays a role in accounting for the spatio-temporalautocorrelation between samples [171], which may not be captured by the DNN features alone.



(a) Deep kernel learning (b) Deep Gaussian process

Fig. 7. Illustration of deep kernel learning and deep (compositional) Gaussian process.

Deep (compositional) Gaussian process [29] represents another category, focusing on function composition inspired by deep neural network architecture. In this model, each layer is a GP model whose inputs are determined by the output of the preceding GP, as shown in Fig. 7 (b). The recursive composition of GPs results in a more complex distribution over the predicted target variables, which addresses the joint Gaussian distribution limitation of traditional GP. The forward propagation and joint probability distribution of the model can be expressed as follows:

$$\boldsymbol{y} = \boldsymbol{f}_L(\boldsymbol{f}_{L-1}(...\boldsymbol{f}_1(\boldsymbol{x}))) \text{ and } p(y, \boldsymbol{f}_L, ...\boldsymbol{f}_1|\boldsymbol{x}) \sim p(\boldsymbol{y}|\boldsymbol{f}_L) \prod_{i=2}^{L} p(\boldsymbol{f}_i|\boldsymbol{f}_{i-1}) p(\boldsymbol{f}_1|\boldsymbol{x}), \tag{6}$$

where each function $f_i(\cdot)$ represents a Gaussian process model. The intermediate distributions follow Gaussian distributions, but the final distribution will capture a more complex distribution over the target variable $\boldsymbol{y}$. The composition also allows uncertainty to propagate from the input through each intermediate layer. However, the challenge associated with the compositional Gaussian process lies in maximizing the data likelihood $p(\boldsymbol{y}|\boldsymbol{x})$, the direct marginalization of hidden variables $\boldsymbol{f}_i$ is intractable. To overcome this challenge, variational inference introduces inducing points on each hidden layer and by optimizing over the variational distribution $q(\boldsymbol{f}_i)$. Then, the marginal likelihood lower bound can be obtained by propagating the variational approximation at each layer [147]. The framework also allows for incorporating partial or uncertain observations into the model by placing a prior over the input variables $\boldsymbol{x}$ and propagating uncertainty layer by layer [30].

Table 2. Comparison of UQ methods for model uncertainty.

Category	Method	Pros	Cons
BNN: Capture parameter uncertainty via posterior estimation.	Variational Inference & Laplace Approximation [88, 102, 119, 127, 128]	Practically efficient for large models	Approximation is based on assumptions (e.g., Gaussian distribution)
	MCMC [111, 133, 161]	Flexible for any distribution assumption	High computational cost
	MC Dropout [46, 57]	Simple, scalable, flexible for large neural networks	Lacks theoretical grounding
Ensemble: Capture uncertainty from models, parameters, and hyperparameters	Network/Bootstrap /Hyper-Ensemble [89, 105, 157]	Capture uncertainty from architecture, learning algorithms, hyperparameters	High computational cost
Sample Distribution: Uncertainty due to OOD inputs.	Deep Gaussian Processes [29, 160, 171]	Strong theoretical grounding in GPs	Poor scalability to large datasets
	Distance-aware DNNs [99, 148, 149]	Simple and efficient	Embedding distances may not reflect input similarity

Distance-aware neural network: Although modern neural networks can extract representative features from large datasets, they do not consider how distinct new test samples may be from the training dataset. To address the uncertainty resulting from sample feature density, many approaches incorporate distance awareness between samples into neural network design, inspired by Gaussian processes [99]. Assume the input data manifold is equipped with a metric $||\cdot||_{\mathcal{X}}$, quantifying the distance between samples in the feature space. The intuition behind a distance-aware neural network is to leverage DNNs' feature extraction capability to learn a hidden representation $h(\boldsymbol{x})$ that reflects a meaningful distance in the data manifold $||\boldsymbol{x} - \boldsymbol{x}'||_{\mathcal{X}}$ as shown in Fig 8. However, one significant issue with the unconstrainted DNN model is the *feature collapse*, which means DNN feature extraction can map *in-distribution* data (training samples) and *out-of-distribution* data (lies further from the training data) to similar latent representations. Thus the Gaussian process based on the DNN extracted feature can be over-confident for those samples that lie further away from training samples. To avoid the feature collapse problem, several constraints have been proposed:

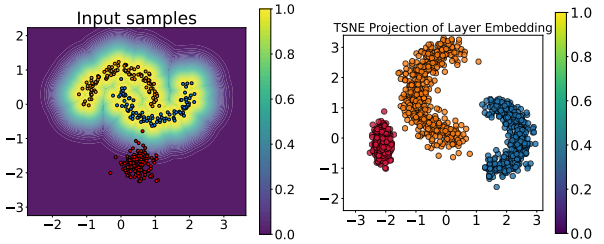


Fig. 8. Two variants of the architecture (adapted from [99]).

sensitivity and smoothness [148, 149]. Sensitivity implies that a small change in the input should result in a small change in the feature representation, which helps ensure distinct samples are mapped to different latent features. Smoothness implies that small changes in the input should not cause dramatic changes in the output. In general, these two constraints can be ensured by *bi-Lipschitz* constraints [100], which means the relative changes in the hidden feature representation $h_{\boldsymbol{\theta}}(\boldsymbol{x})$ are bounded by changes in input space as Eq. 7 shows.

$$L_1 * |\boldsymbol{x} - \boldsymbol{x}'|_{\mathcal{X}} < |h_{\boldsymbol{\theta}}(\boldsymbol{x}) - h_{\boldsymbol{\theta}}(\boldsymbol{x}')|_{\mathcal{H}} < L_2 * |\boldsymbol{x} - \boldsymbol{x}'|_{\mathcal{X}}. \tag{7}$$

To enforce bi-Lipschitz constraints to DNN, two approaches have been proposed: spectral normalization and gradient penalty. Spectral normalization [99] claims that the bi-Lipschitz constants L can be ensured to be less than one by normalizing the weights matrix in each layer with the spectral norm. This method is fast and effective for practical implementation. The other approach is called gradient penalty [149], which introduces another loss penalty: the square gradient at each input sample $\nabla^2_{\boldsymbol{x}} h_{\boldsymbol{\theta}}(\boldsymbol{x})$. This will add a soft constraint to the neural networks to constrain the Lipschitz coefficients. Gradient penalty is a soft constraint compared to spectral normalization and is computationally more intensive. Some works extend the distance-aware framework to the non-parametric estimation of the conditional label distribution, enabling more flexible modeling of the distribution [85, 155].

Summary of Model Uncertainty: We summarize and compare existing methods for quantifying model uncertainty in Table 2. BNN models can capture model uncertainty arising from parameter estimation but usually have very high computational costs, making them infeasible for practical applications. The ensemble models can capture uncertainty from multiple perspectives, such as model architecture misspecification, limited training dataset, and choices of hyperparameters. The method also has a high computational cost. On the other hand, the sample distribution-based model can capture uncertainty due to distribution shifts, but it's often hard to learn the distance-aware feature space and the method requires adding constraints to the neural network model.

Although we categorize BNN and ensemble methods under model uncertainty, they can be used to capture both model and data uncertainty simultaneously with minor modification. For BNNs, as shown in Equation 3, model uncertainty is reflected by the posterior distribution over model parameters, while data uncertainty can be captured by averaging the predictive distributions from multiple model samples. Similarly, for ensemble methods, data uncertainty can be further obtained through averaging the class probability predictions across individual models. A common approach to obtain the class probability is through the softmax of output logits. However, this method may lead to overconfident predictions [57]. To address this issue, a variety of calibration techniques, including both parametric and non-parametric approaches, have been developed to capture the predicted distribution for data uncertainty. More detailed discussions are in Section 4.3.1.

4.2 Data Uncertainty

This section discusses the existing methodologies that quantify data uncertainty in DNN models. Data uncertainty is modeled by the distribution $p(y|\boldsymbol{x}, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ represents the neural network parameters. We categorize these approaches into deep discriminative models and deep generative models for learning this distribution.

4.2.1 Deep discriminative model . To quantify data uncertainty, a discriminative model outputs a predictive distribution directly using a neural network. Specifically, the distribution can be modeled as a parametric or non-parametric model. A parametric model assumes the output follows a specified family of probability distributions with parameters (e.g., mean and variance for a Gaussian distribution) estimated by the neural network. In contrast, the non-parametric model does not

have any assumption on the underlying distributions. We will discuss existing methods for each category in detail.

Parametric model: The standard approach for quantifying data uncertainty is to directly learn a parametric model for $p(y|\boldsymbol{x}, \boldsymbol{\theta})$. From a frequentist perspective, there exists a single set of optimal parameters $\boldsymbol{\theta}^*$. For the classification problem, $p(y|\boldsymbol{x}, \boldsymbol{\theta})$ is a parameterized categorical distribution over k classes, $\boldsymbol{y} \in \mathbb{R}^p$ is a multidimensional structured output, with distribution parameters $\boldsymbol{\pi} = (\pi_1, ..., \pi_K)$ predicted by the model output as Eq. 8 shows.

$$p(y|\boldsymbol{x}, \boldsymbol{\theta}) = \text{Categorical}(y; \boldsymbol{\pi}) \ , \ \boldsymbol{\pi} = f(\boldsymbol{x}; \boldsymbol{\theta}), \ \sum_{c=1}^{K} \pi_c = 1, \ \pi_c > 0. \tag{8}$$

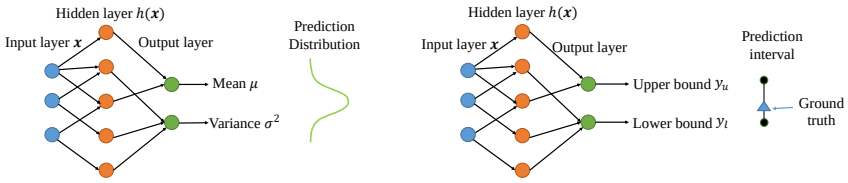
In order to obtain the categorical distribution parameters, a straightforward approach uses the softmax probability output $\pi_i = \frac{\exp(h_i(x;\boldsymbol{\theta}))}{\sum_{c=1}^{k} \exp(h_c(x;\boldsymbol{\theta}))}$ as the predicted uncertainty, but these methods tend to be over-confident because the softmax operation squeezes the prediction probability toward extreme values (zero or one) for the vast majority range of h_i [62]. Subsequent work [57] calibrate the softmax uncertainty with *temperature scaling*, which simply adds an additional hyperparameter T to the softmax calculation as $p = \frac{\exp(h_i(x)/T)}{\sum_{c=1}^{k} \exp(h_c(x)/T)}$ to overcome the overconfident outputs. This approach is straightforward to implement but may still be overconfident due to a lack of constraints and requires calibration of the parameter.

For regression problems, data uncertainty is assumed to arise from inherent noise in the training data (e.g., measurement or labeling error). In general, the training data is modeled as independent additive Gaussian noise with sample-dependent variance $\sigma(\boldsymbol{x})$, which indicates the target variable $y_i = f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) + \epsilon(\boldsymbol{x}_i)$. $\epsilon(\boldsymbol{x}_i)$ is the independent heterogeneous Gaussian noise, representing each sample's uncertainty. In this way, the output will be a parameterized continuous Gaussian distribution, as Eq. 9 and Fig. 9 (a) show. The mean and variance are predicted from the neural network [78], where the mean represents the model's prediction, and the variance represents the uncertainty of each sample's prediction. To optimize the neural network parameters $\boldsymbol{\theta}$, maximum likelihood optimization is performed jointly on the mean and variance as Eq. 9 shows. This is also known as heteroscedastic regression, which assumes the observational noise level varies with different samples. This is suitable for cases where some samples have higher noise (uncertainty), while others have lower. Besides Gaussian distribution, the neural network can also be parameterized with other types of distributions, such as mixture Gaussian distribution [56], which is implemented with mixture density network (MDN) [16], assuming multiple modes for the prediction. MDN has the advantage of accounting for the uncertainty from multiple prediction modes but consumes more computation. Choosing a suitable parameterized distribution is essential and depends on the problem.

$$p(y|\boldsymbol{x}, \boldsymbol{\theta}) = \mathcal{N}(f_{\boldsymbol{\theta}}(\boldsymbol{x}), \sigma_{\boldsymbol{\theta}}(\boldsymbol{x})) \ \text{ and } \ \mathcal{L}_{\text{NN}}(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2\sigma_{\boldsymbol{\theta}}^2(\boldsymbol{x}_i)} ||y_i - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)||^2 + \frac{1}{2} \log \sigma_{\boldsymbol{\theta}}^2(\boldsymbol{x}_i). \tag{9}$$

The advantage of using a predictive distribution is that it can be easily incorporated into existing neural network architectures and requires slight modification to the training and inference process. However, the explicit parameterization form requires choosing the appropriate distribution to capture the underlying uncertainty accurately, which can be hard if no prior information is available.

Non-parametric model: A widely popular approach for indicating data uncertainty is through *prediction interval* (PI) [118]. For regression problems, the prediction intervals output a lower and upper bound $[y_l, y_u]$, where we expect the ground truth y falls within the interval with a prescribed confidence level, $1 - \alpha$, meaning that $p(y \in [y_l, y_u]) > 1 - \alpha$ as shown in Fig. 9 (b). This approach is more flexible and does not require explicit distribution over the prediction variable. Traditional prediction intervals are typically constructed in two steps: first, to learn the point



(a) Prediction distribution example. (b) Prediction interval example.

Fig. 9. Neural network architecture for the parametric and non-parametric model.

estimation of the target variable, obtained by minimizing an error-based loss function (e.g., mean square loss), followed by estimating the prediction variance around the local optimum prediction. The strategy tries to minimize the prediction error but not optimize the interval quality. One approach [79] explicitly constructs a lower and upper bound estimation (LUBE) to improve the PI characteristics, i.e., the width and coverage probability. The basic intuition is that the PI should cover the ground truth with a certain pre-defined probability (confidence level), but should be as narrow as possible. This approach enhances the quality of the constructed PI, but the resulting cost function is non-differentiable and requires Simulated Annealing (SA) sampling to obtain the optimal NN parameters.

To address the non-differentiable limitation of LUBE, an alternative approach uses a coverage width-based loss function [118] with a goal similar to LUBE, as shown in the Eq.10. The *mean prediction interval width* (MPIW) is defined as $|y_u - y_l|$, and the *prediction interval coverage width* (PICP) measures the average probability that the PI covers the ground truth. The total loss encourages the PI to be narrow while having a higher coverage probability above the prescribed confidence level α.

$$\text{Loss} = \text{MPIW} + \lambda * \max(0, (1-\alpha) - \text{PICP})^2. \tag{10}$$

Recent approaches frame prediction interval learning as a constrained optimization problem. This optimization problem can be viewed from two perspectives: primal and dual perspectives. The primal perspective frames the objective as minimizing the PI width under the constraint that the PI attains a coverage probability larger than the confidence level [22], which is expressed as follows:

$$\min_{L,U \in \mathcal{H}, L<U} \mathbb{E}_{\boldsymbol{x} \sim \pi(\boldsymbol{x})}(U(\boldsymbol{x}) - L(\boldsymbol{x}))\ s.t.\ p_\pi(y \in [L(\boldsymbol{x}), U(\boldsymbol{x})]) > 1-\alpha. \tag{11}$$

where $\mathbb{E}_{\boldsymbol{x} \sim \pi(\boldsymbol{x})}$ denotes the expectation concerning the marginal distribution of input samples $\boldsymbol{x}$, and p_π denote the probability of the input-output pair distribution. To enforce the optimality and feasibility of the optimization problem, the tradeoff is developed through the studying of two characteristics of this approach: Lipschitz continuous model class [150] and Vapnik–Chervonenkis (VC)-subgraph class [22]. On the other hand, the dual perspective frames the objective as maximizing PI coverage probability subject to a fixed global budget constraint on average PI width in a batch setting [130].

$$\min_{f \in \mathcal{F}} \mathbb{E}_{(\boldsymbol{x},y) \sim \pi(\boldsymbol{x},y)} L(y, U(\boldsymbol{x}), L(\boldsymbol{x}))\ s.t.\ \sum_i (U(\boldsymbol{x}_i) - L(\boldsymbol{x})_i) < B. \tag{12}$$

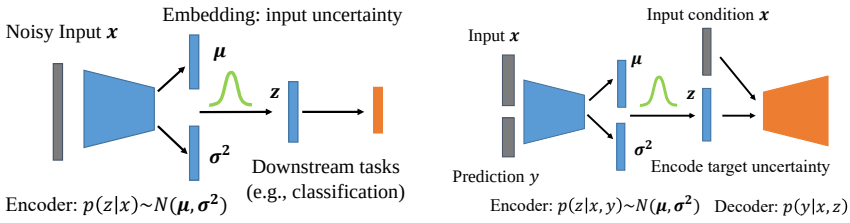
Researchers presented a discriminative learning framework that optimizes the expected error rate under a budget constraint on the interval sizes. This approach avoids single-point loss and provides a statistical guarantee of generalization for the entire population. In contrast to the primal setup, the dual perspective in batch learning constructs the prediction interval of a group of test points simultaneously, reducing the computational overhead.

4.2.2 Deep Generative Model. Deep generative models (DGMs) are a family of probabilistic models that aim to learn the complex, high-dimensional data distribution $p_{\text{data}}(\boldsymbol{x})$ with DNN. DGMs are capable of learning the intractable data distribution in the high-dimensional feature space $\mathcal{X} \subseteq \mathbb{R}^n$ from a large number of independent and identically distributed observed samples $\{\boldsymbol{x}_i\}_{i=1}^m$. Specifically, they learn a probabilistic mapping from some latent variables $\boldsymbol{z} \in \mathbb{R}^d$ that follow a tractable distribution to the data distribution, such as $\mathcal{N}(\mathbf{0}, \mathbf{I})$ to the data distribution $p_{\text{data}}(\boldsymbol{x})$. Mathematically, the generative model can be defined as the mapping function $g_{\boldsymbol{\theta}}(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^n$, where d and n are the dimensions of latent variable and original data, respectively. A deep generative model is capable of capturing probabilistic distribution for high-dimensional structured outputs (e.g., images).

The basic idea is to employ DGM to learn the *predictive distribution* $p(\boldsymbol{y}|\boldsymbol{x})$ given the supervised training data pairs $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^m$. In this subsection, we use bold $\boldsymbol{y}_i$ to denote the high-dimensional structured outputs. It should be noted that to learn the *predictive distribution* instead of the data distribution in feature space, the *conditional deep generative model* (cDGM) [140] should be employed. Generally speaking, cDGM-based uncertainty quantification models learn a conditional density over the prediction $\boldsymbol{y}$, given the input feature $\boldsymbol{x}$. This amounts to learning a model $g_{\boldsymbol{\theta}}(\boldsymbol{z}, \cdot) : \mathbb{X} \rightarrow \mathbb{Y}$ such that the generative model $g(\boldsymbol{z}, \boldsymbol{x})$ with $\boldsymbol{z} \sim p(\boldsymbol{z})$ approximates the true unknown distribution $p_{\text{true}}(\boldsymbol{y}|\boldsymbol{x})$. The variability of the prediction distribution $p(\boldsymbol{y}|\boldsymbol{x})$ is encoded into the latent variable $\boldsymbol{z}$ and the generative model. During inference, for any $\boldsymbol{x} \in \mathbb{X}$, we can generate m samples of $\boldsymbol{y}_i$ with $g_{\boldsymbol{\theta}}(\boldsymbol{z}_i, \boldsymbol{x})$ and $\boldsymbol{z}_i \sim p(\boldsymbol{z})$. By analyzing the variability of the samples $\{\boldsymbol{y}_i\}_{i=1}^m$, we can quantify prediction uncertainty.

In the following subsection, we examine three types of deep generative models: the variational autoencoder (VAE) [82], the generative adversarial network (GAN) [54], and the diffusion model [64]. The VAE, a likelihood-based generative model, is trained by maximizing the evidence lower bound (ELBO) of the likelihood function. GANs, on the other hand, are implicit generative models trained through a two-player zero-sum game framework. Lastly, the diffusion model is a probabilistic generative framework that employs a multi-step denoising process. We explore how each of these frameworks can be applied to estimate prediction uncertainty.

VAE-based model: The VAE model consists of two modules: an *encoder* and a decoder. The encoder network $q_{\phi}(\boldsymbol{z}|\boldsymbol{x})$ aims to embed the high dimensional structural output $\boldsymbol{x}$ into a low-dimensional code $\boldsymbol{z}$, that captures the inherent ambiguity or noise of the input data. The decoder $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ aims to reconstruct the input feature. VAE model has been popular for modeling structured output uncertainty, especially for tasks on image data, because of its capability to model global and local structure dependency in regular grid images. Specifically, two kinds of frameworks based on the VAE model have been proposed to account for the data uncertainty arising from *input noise* and *target output noise*: The first category aims to capture noise present in the input samples. The basic idea is to embed each sample as a Gaussian distribution instead of a deterministic embedding in the low dimensional latent space, where the mean represents the feature embedding and variance represents the uncertainty of the embedding [19]. The method accounts for varying noise levels inherent in the dataset, which is ubiquitous in many kinds of real-world datasets, for example, face image recognition [19], medical image reconstruction [39]. This probabilistic embedding framework leverages the VAE architecture to estimate the embedding and uncertainty simultaneously. The second category aims to capture the noise that lies in the target outputs, where the ground truth is imperfect, ambiguous, or corrupted. This scenario is common in the medical domain [92], where the objects in the image are ambiguous and the experts may not reach a consensus on the class of the objects (large uncertainty). Thus for segmentation or classification tasks, the model should be aware of the prediction uncertainty. To capture the prediction uncertainty in the target outputs, the



(a) Input uncertainty. (b) Target prediction uncertainty.

Fig. 10. VAE framework for uncertainty quantification.

conditional VAE (cVAE) [140] framework is adopted. Specifically, cVAE formulates the prediction distribution as an integration over the latent embedding $\boldsymbol{z}$,

$$p(\boldsymbol{y}|\boldsymbol{x}) = \int p(\boldsymbol{y}|\boldsymbol{x}, z)p(z|\boldsymbol{x})dz \approx \frac{1}{n}\sum_{j=1}^{n} p(\boldsymbol{y}|\boldsymbol{x}, z_j), \text{ where } z_j \sim p(z). \tag{13}$$

The cVAE model is trained by maximizing the evidence lower bound of the likelihood. Then, during inference, multiple latent features $\boldsymbol{z}_j$ can be drawn from the prior distribution, and the integration over latent $\boldsymbol{z}$ can be approximated with the sampling distribution [121]. Probabilistic U-Net model [84] combines the architecture of cVAE and U-Net model by treating the U-Net model as the encoder to produce a probabilistic segmentation map. The U-Net model can capture multi-scale feature representations in the low-dimensional latent space to encode the potential variability in the segmentation map. These models are illustrated in Fig. 10 (a) and (b).

In summary, the VAE-based framework can take into consideration the data uncertainty coming from the input noise or the target output noise and can integrate state-of-the-art neural network architectures into its framework, making it more flexible for many kinds of applications. The key success lies in modeling the joint probability of all samples (pixels) in the image. The approach is suitable for structured uncertainty quantification (e.g., image grid structure, graph structure) by learning the implicit joint distribution of the structure.

GAN-based generative model: GAN is a type of generative model trained with a two-player zero-game. It consists of a *generator* and a *discriminator*. In conditional GAN (cGAN), the generator takes the input x and random noise $\boldsymbol{z}$ as input and generates the target variables y: $\mathcal{G} : (\boldsymbol{x}, \boldsymbol{z}) \rightarrow \boldsymbol{y}$. The discriminator is trained to distinguish between generated samples and ground-truth samples. GAN has been adopted in many domains. For example, In the transportation domain, GAN has been used for traffic volume prediction [109]. The flow model is integrated with GAN to enable likelihood estimation and better uncertainty quantification. Another approach [49, 115] extends this method using the Wasserstein GAN [6] with gradient penalty to improve model convergence. The key advantage of deep generative modeling for uncertainty quantification is that it directly parameterizes a deep neural network to represent the prediction distribution without needing an explicit distribution format. Moreover, it can integrate a physics-informed neural network for better uncertainty estimation of physical science [31]. However, GAN-based models are harder to train, especially for GAN-based models. Model convergence is not guaranteed.

Diffusion-based generative model: Diffusion models are a family of probabilistic generative models that progressively destroy data by injecting noise in the forward diffusion process, then learn to reverse this process to generate new data samples through a backward process [165]. The forward diffusion process gradually adds noise to the data according to a variance schedule β_t: $q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\,\mathbf{x}_{t-1}, \beta_t\,\mathbf{I})$. The reverse process is defined as: $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) =$

$\mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t))$, where μ_θ and Σ_θ are learned with a neural network that aims to reconstruct (denoise) $\mathbf{x}_{t-1}$ from $\mathbf{x}_t$ at each step. The training objective supervises the model to recover data from noise accurately. Diffusion models have been applied to images [64], videos [10], time series [142], and scientific simulations [44]. Similar to conditional VAEs and GANs, conditional diffusion models can be applied to uncertainty quantification of model outputs given input features [4, 44, 95]. Latent diffusion models enhance the efficiency of diffusion models for high-resolution images or video generation by operating on lower dimensional latent embeddings through (e.g., through an autoencoder) [47, 112, 129]. As summarized in Table 3 below, the primary advantages of diffusion models are their expressiveness to model complex high-dimensional distributions and generate high-quality samples [165]. A major drawback, however, is the slow training and inference due to the iterations of hundreds to thousands of steps [165]. There is ongoing research that addresses this issue [80].

Summary on Data Uncertainty: Table 3 compares the pros and cons of existing approaches for data uncertainty quantification. Discriminative models are simple but unsuited for capturing structured output uncertainty. Generative can better quantify structured output uncertainty but requires multiple sampling of model outputs. Among deep generative models, diffusion models are currently more popular, but more efforts are needed to improve their efficiency.

The methods for data uncertainty, including deep discriminative models and deep generative models, can also quantify total uncertainty with minimal modifications. For instance, deep generative models inherently account for data uncertainty by marginalizing over latent variables. Model uncertainty can be estimated by computing the variance of predictions generated from multiple latent samples, similar to Bayesian neural networks. For deep discriminative models, model uncertainty can be assessed by measuring the variance of predictions from multiple parameterized models. A more systematic review of techniques for disentangling and quantifying multiple types of uncertainty is in Section 4.3.

Table 3. A comparison of UQ methods for data uncertainty

Model	Approach	Pros	Cons
Deep Discriminative model **Pros**: No need to modify network architecture. **Cons**: Not suitable for structured output.	Predict a parametric distribution [56, 78].	Simple model training.	Assume a parametric output distribution.
	Non-parametric: predict an interval [22, 118, 130, 162].	No rigid assumption on output distribution.	Need to design new training loss.
Deep Generative model **Pros**: Capture uncertainty for structured output data. **Cons**: Require multiple sampling for uncertainty quantification.	VAE-based model [19, 39, 84, 121].	Stability in training, probabilistic outputs.	Less expressive compared with GAN and diffusion models.
	GAN-based model [49, 109, 115].	More expressive than VAE.	Instability in training and lack of probabilistic framework.
	Diffusion-based model [15, 36, 44, 138].	More expressive than VAE, probabilistic framework.	High computational cost.

4.3 Model and data uncertainty

Besides considering the data and model uncertainty separately, many frameworks attempt to jointly consider the two kinds of uncertainty for more accurate quantification. In this part, we will review existing frameworks that aims to quantify both types of uncertainty simultaneously.

4.3.1 Approaches combining data and model uncertainty. A straightforward way to consider both data and model uncertainty is to select one of the approaches in each category and combine them

in a single framework. Below, we will introduce some major ways to combine approaches for data and model uncertainty and their potential drawbacks.

Combine BNN model with prediction distribution: The method aims to capture both data and model uncertainty within a single framework [78] by combining BNN with prediction distribution. The model uncertainty is captured with the BNN approximation approach. Specifically, MC dropout is adopted due to its simplicity for implementation. For each dropout forward pass, a sample of the weights is drawn from the weight distribution approximation $\mathbf{W}_t \sim \text{Bernoulli}(p)$, where p is the dropout rate, then one forward prediction can be made with the weight by $y_t = p(y|\boldsymbol{x}, \mathbf{W}_t)$. To obtain the data uncertainty, the output is formulated as a parameterized Gaussian distribution instead of point estimation $[y_t, \sigma_t^2] = p(y|\boldsymbol{x}; \mathbf{W}_t)$, where y_t is the target variable mean prediction and σ_t^2 is the prediction variance for a single forward prediction. With multiple dropout forward passes, we have a set of T prediction samples $\{y_t, \sigma_t^2\}_{t=1}^T$. The predictive uncertainty in the combined model can be approximated with the law of total variance expressed as $\text{Var}(y)$ in Eq. 14. The intuition behind this equation is that total uncertainty comprises two parts, the last $\frac{1}{T}\sum_{t=1}^T \sigma_t^2$ represents data uncertainty on average, and the first part $\frac{1}{T}\sum_{t=1}^T y_t^2 - (\frac{1}{T}\sum_{t=1}^T y_t)^2$ represents the disagreement across T MC-dropout models, which captures model uncertainty.

$$\text{Var}(y) \approx \frac{1}{T}\sum_{t=1}^{T} y_t^2 - (\frac{1}{T}\sum_{t=1}^{T} y_t)^2 + \frac{1}{T}\sum_{t=1}^{T} \sigma_t^2. \tag{14}$$

Combine ensemble model with prediction distribution: This approach [89] combines the ensemble method with prediction distribution. The deep ensemble method constructs an ensemble of DNN models $\mathcal{M} = \{\mathcal{M}_i\}_{i=1}^K$, where each model $\mathcal{M}_i$ can be set with different parameters, or architecture choices. The model uncertainty is expressed as the variance or "disagreement" among the ensemble models. In this way, the output of each model is modified as a parameterized distribution to capture the data uncertainty. Similar to MC-dropout, we have an ensemble of prediction distribution $\{p(y|\boldsymbol{x}, \mathcal{M}_i)\}_{i=1}^K$. In this part, we take the classification problem as an example, where the prediction distribution is a parameterized categorical distribution. The total uncertainty is captured with the entropy of average prediction distribution $\mathcal{H}(\mathbb{E}_{p(\mathcal{M}_i)} p(y|\boldsymbol{x}, \mathcal{M}_i))$, and the data uncertainty is the average entropy of each model, expressed as $\mathbb{E}_{p(\mathcal{M}_i)}\mathcal{H}(p(y|\boldsymbol{x}, \mathcal{M}_i))$. The model uncertainty can be expressed with the mutual information between the prediction and the ensemble model $y, \mathcal{M}$ as expressed in Eq. 15.

$$\text{MI}(y, \mathcal{M}) = \mathcal{H}(\mathbb{E}_{p(\mathcal{M}_i)} p(y|\boldsymbol{x}, \mathcal{M}_i)) - \mathbb{E}_{p(\mathcal{M}_i)}\mathcal{H}(p(y|\boldsymbol{x}, \mathcal{M}_i)). \tag{15}$$

Combine ensemble model with prediction interval: Since the prediction interval constructed in some approaches accounts only for the data noise variance, not the model uncertainty. To improve the total uncertainty estimation, ensemble methods are adopted to combine prediction intervals with ensemble methods to account for model uncertainty arising from model architecture misspecification, parameter initialization, etc. [118]. Specifically, Given an ensemble of models trained with different model specifications or sub-sampling of training datasets, where the model prediction intervals are denoted as $[y_l^{ij}, y_u^{ij}]$ for sample $i = \{1, ..., n\}$ and model $j = \{1, ..., m\}$, the model uncertainty can be captured by the variance of the lower bound ${\sigma_l^{(i)}}^2$ and upper bound variance ${\sigma_u^{(i)}}^2$. For example, the uncertainty of the lower bound is given by:

$$\begin{aligned} {\sigma_l^{(i)}}^2 &= \frac{1}{m-1}\sum_{j=1}^{m}(y_l^{(ij)} - \hat{y}_l^{(i)})^2, \text{ where } \hat{y}_l^{(i)} = \frac{1}{m}\sum_{j=1}^{m} y_l^{(ij)}, \\ {\sigma_u^{(i)}}^2 &= \frac{1}{m-1}\sum_{j=1}^{m}(y_u^{(ij)} - \hat{y}_u^{(i)})^2, \text{ where } \hat{y}_u^{(i)} = \frac{1}{m}\sum_{j=1}^{m} y_u^{(ij)}. \end{aligned} \tag{16}$$

Then the new prediction interval $[\tilde{y}_l, \tilde{y}_u]$ with 95% confidence level can be constructed as:

$$\tilde{y}_l = y_l - 1.96\sigma_l^{(i)^2}, \text{ and } \tilde{y}_u = y_u + 1.96\sigma_u^{(i)^2}. \tag{17}$$

The constructed interval can reflect both the data uncertainty and model uncertainty. However, this approach relies on variance from the ensemble's lower and upper bounds, which lacks theoretical justification due to the independent treatment of the two boundaries. To overcome this limitation, one recent approach proposes a split normal aggregation method to aggregate the prediction interval ensembles into final intervals [132]. Specifically, the method fits a split normal distribution (two pieces of normal distribution) over each prediction interval, and then the final prediction will become a mixture of split normal distribution. The PI can be derived from the $1-\alpha$ quantile of the cumulative distribution.

In summary, to capture both the data and model uncertainty, existing literature can combine the methodologies in the two categories. There are several limitations to the combination approaches: first, the BNN or ensemble models require multiple forward passes for the prediction, which introduces computation overhead and extra storage. Efficiency is a concern. Second, the simple combination of data and model uncertainty lacks a theoretical guarantee, which requires post hoc calibration of the model.

4.3.2 Evidential deep learning. To overcome the computational challenge for the combination approaches, evidential deep learning was proposed to use one single deterministic model to capture both the data and model uncertainty without multiple forward passes of the neural network [9, 20, 104, 134]. The intuition of evidential deep learning is to predict class-wise evidence instead of directly predicting class probabilities. In the following section, we review these methodologies, including their advantages and disadvantages.

As discussed in the aforementioned sections, for classification problems, existing deep learning based models explicitly or implicitly predict class probabilities (categorical distribution parameters) with softmax-layer parameterized by DNNs to quantify prediction uncertainty. However, softmax prediction uncertainty often tends to be overconfident [62]. Evidential deep learning is developed to overcome the limitation by introducing evidence theory [75] to neural network frameworks. The goal of evidential deep learning is to construct predictions based on evidence and predict the parameters of Dirichlet density. For example, considering the 3-class classification problem, a vanilla neural network directly predicts the categorical distribution for each class $\pi = \pi_1, \pi_2, \pi_3$ with $\sum_i \pi_i = 1$. However, this approach can only represent a point estimation of prediction distribution. On the other hand, evidential deep learning aims to predict the *evidence* for each class $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \alpha_3\}$ with the constraint $\boldsymbol{\alpha} > 0$, which can be considered as the parameters of Dirichlet distribution [134]. The framework is shown in Fig 11, where the output is the *evidence* $\boldsymbol{\alpha}$ for each class, and the prediction distribution is sampled from the Dirichlet distribution. The expected prediction distribution for each class is $p_i = \frac{\alpha_i}{\sum_{c=1}^{3} \alpha_c}$, whose entropy represents data uncertainty. On the other hand, model uncertainty is reflected by the total evidence $\sum_i \alpha_i$, which means the more evidence we collect, the more confident the model is.

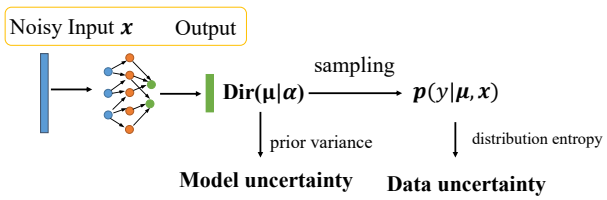
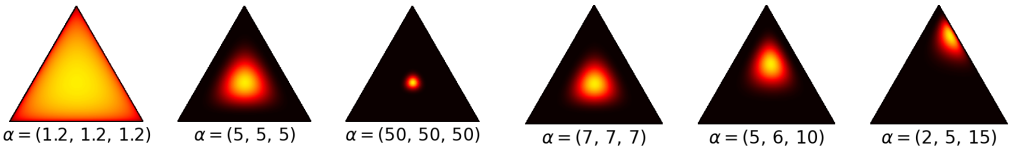


Fig. 11. Evidential deep learning architecture.

Mathematically, evidential deep learning aims to learn the prior distribution of categorical distribution parameters, which is represented by the Dirichlet distribution. The Dirichlet distribution



(a) Keep data uncertainty fixed , and model uncertainty decrease from left to right.

(b) Keep the model uncertainty fixed, and the data uncertainty decreases from left to right (the entropy of the categoric distribution decreases).

Fig. 12. Dirichlet distribution density visualization.

is parameterized by its concentration parameters $\boldsymbol{\alpha}$ (evidence) where α_0 is the sum of all α_i and is referred to as the precision of the Dirichlet distribution. A higher value of α_0 will lead to sharper distribution and lower model uncertainty. As shown in Eq.18 below, the $\text{Dir}(\boldsymbol{\mu}|\boldsymbol{\alpha})$ defines a probability density function over the k-dimensional random variable $\boldsymbol{\mu} = [\mu_1, ..., \mu_k]$, where k is the number of classes, $\boldsymbol{\mu}$ belongs to the standard $k-1$ simplex ($\mu_1 + ... + \mu_k = 1$ and $\mu_i \in [0, 1]$ for all $i \in 1, ..., k$), and can be regarded as the categorical distribution parameters.

The relationship between Dirichlet distribution and uncertainty quantification can be illustrated using a 2-simplex. The random variable $\boldsymbol{\mu} = [\mu_1, \mu_2, \mu_3]$ is represented by its Barycentric coordinates in Fig. 12 on the 2-simplex. The Barycentric coordinate is a coordinate system where points are located inside a simplex, and the value in each coordinate can be interpreted as the fraction of mass placed at each corresponding vertex of the simplex. Fig. 12 (a) shows a scenario where the evidence parameters are equal, resulting in indistinguishable classes and implying high data uncertainty (high entropy for the sampled $\boldsymbol{\mu}$). As the total evidence $\sum_i \alpha_i$ increases, the density becomes more concentrated, which means the model uncertainty decreases, while the data uncertainty remains fixed. Fig. 12 (b) shows a scenario with fixed model uncertainty, as the sum of evidence parameters remains constant. When the evidence becomes imbalanced, the density becomes more concentrated toward one class, thus decreasing the data uncertainty.

$$\text{Dir}(\boldsymbol{\mu}|\boldsymbol{\alpha}) = \frac{\Gamma(\alpha_0)}{\prod_{c=1}^{K}\Gamma(\alpha_c)}\prod_{c=1}^{K}\mu_c^{\alpha_c - 1}, \alpha_c > 0,\ \alpha_0 = \sum_{c=1}^{K}\alpha_c. \tag{18}$$

Due to the intriguing property of Dirichlet distribution, evidential deep learning directly predicts the parameters of Dirichlet density. For example, the Dirichlet prior network (DPN) [104] learns the concentration parameter $\boldsymbol{\alpha}$ for the Dirichlet distribution $\boldsymbol{\alpha} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta})$. The categorical distribution parameters are then drawn from the Dirichlet distribution as $p(\boldsymbol{\mu}|\boldsymbol{x}, \boldsymbol{\theta}) = \text{Dir}(\boldsymbol{\mu}|\boldsymbol{\alpha})$. The predicted class probability is the average over possible values of $\boldsymbol{\mu}$.

$$p(w_c|\boldsymbol{x}, \boldsymbol{\theta}) = \int p(w_c|\boldsymbol{\mu})p(\boldsymbol{\mu}|\boldsymbol{x}, \boldsymbol{\theta}) = \frac{\alpha_c}{\alpha_0}. \tag{19}$$

To measure uncertainty from the Dirichlet distribution, the total uncertainty is computed as the entropy of the average predictive distribution, while data uncertainty is determined by averaging the entropy across each realization of $\boldsymbol{\mu}$. Several approaches have extended the prior network to handle regression tasks, and a posterior network has been introduced to enhance the reliability of uncertainty estimation. Additionally, other second-order methods have been proposed to jointly quantify data and model uncertainty within a single deterministic framework [14, 65, 86].

The advantage of evidential deep learning is the approach requires only a single forward pass during inference and is much more computationally efficient. The approach also explicitly distinguishes data and model uncertainty in a principled way. The disadvantage is that the training

stage is more complex and does not guarantee the same prediction accuracy as a vanilla network and does not leverage existing advances in DNNs. Furthermore, the training stage requires OOD samples to learn effective representations and increase the amount of dataset.

Table 4. A comparison of UQ methods for both model and data uncertainty

Model	Approaches	Pros	Cons
Combination of existing approaches	Combine BNN with prediction distribution [78].	Capture uncertainty from both data noise and model parameters.	Require multiple forward pass during inference.
	Combine ensemble with prediction distribution [89].	Capture uncertainty from both data noise and mode architectures.	Require more computation and storage requirement.
	Combine ensemble with prediction interval [118, 132].	Capture both data and model uncertainty and do not need explicit parametric distribution form.	Require modification on existing DNN training process.
Evidential deep learning	Evidential deep learning [9, 20, 104, 134].	Computationally efficient relative to combination approaches.	Strong assumption on prior distribution. Difficult to train.
Conformal Prediction	Prediction set or interval [18, 60, 76, 103].	No rigid assumption in distribution.	Require exchangeability on sample distribution.

4.3.3 Conformal Prediction. Though existing UQ approaches can quantify total uncertainty from various sources, a major limitation is the lack of formal guarantees. To address this issue, conformal prediction (CP) is a post-processing approach that constructs a finite prediction set with a statistical guarantee to cover the true label [5]. Conformal prediction, which belongs to distribution-free uncertainty quantification is model-agnostic, and provides formal results for marginal coverage, defined as the average probability that the true class is contained in the prediction set [5]. Formally, this coverage guarantee states:

$$\mathbb{P}(y \in C(x)) \geq 1 - \alpha. \tag{20}$$

where $\boldsymbol{x}$ is a sample instance, and y is the ground truth. Unlike traditional confidence intervals which typically assume a specific distribution of data or residuals, conformal prediction provides a non-parametric and distribution-free approach [76]. This is advantageous for deep learning, where traditional assumptions are often violated due to model complexity and non-linearity.

There are several approaches to developing conformal predictions for deep learning models. One approach is to apply conformal methods directly to pre-trained deep learning models [103]. Second, credal Bayesian deep learning trains an (uncountable) infinite ensemble of BNNs using only finitely many elements and outputs a prediction set for total uncertainty estimation [18]. Third, density-based deep conformal prediction models uncertainty from out-of-distribution samples by considering distances between samples explicitly [60]. Common conformal prediction methods assume that data instances are independently and identically distributed (i.i.d.). Recent efforts extend conformal prediction beyond this assumption to graph data by introducing a permutation invariance condition [163, 174] and to time series data through adaptive conformal prediction [172].

Summary on UQ for Both Model and Data Uncertainty: Table 4 compares the pros and cons of existing approaches for both data and model uncertainty. The combination of data and model uncertainty estimation methods is simple but computationally expensive. On the other hand, evidential deep learning and conformal prediction are more efficient but require more assumptions on the data distributions.

5 UNCERTAINTY ESTIMATION IN VARIOUS MACHINE LEARNING PROBLEMS

In this section, we discuss several major ML problems where UQ can play a critical role.

5.1 Out-of-distribution detection

A fundamental assumption in deep neural networks is that the test data distribution closely resembles the training data distribution $p_{\text{train}}(x) \approx p_{\text{test}}(x)$. However, in complex real-world scenarios, a DNN can encounter out-of-distribution (OOD) samples that differ from the training data distribution. This can lead to significant drops in prediction performance. A DNN model should be able to recognize these situations.

Given a training data distribution $p(x)$, OOD data includes samples that are either unlikely under the training data distribution or outside the support of $p(x)$. Accurate detection of OOD samples is important for safety-critical applications, e.g., autonomous driving [135], medical image analysis [52]. Since OOD samples lie further away from the training samples, the model may not generalize well and could produce unstable predictions. The primary uncertainty for OOD data is concerned with model uncertainty because the model trained with in-distribution may not generalize well to other domains.

Existing approaches: Existing approaches leveraging the model uncertainty framework are much more popular for OOD detection. For example, drop-out-based BNN approaches have been applied to OOD detection and improved the performance using randomized embeddings from intermediate layers of a dropout BNN [113] and node-based BNN [146]. Deep ensembles are simple and well-performing on OOD detection [89, 173]. Recent advances have developed distance-aware DNN for more accurate OOD detection by imposing constraints on the feature extracting process [93, 99]. The evidential deep learning framework has also demonstrated its capability on OOD detection in many benchmark datasets because of the explicit distinction between two types of uncertainty in the framework [20].

5.2 Active Learning

Obtaining labeled data for deep learning models can be laborious and time-consuming. Active learning [126] aims to solve the data labeling issue by learning from a small amount of data and choosing by the model what data it requires the user to label and retrain the model iteratively. The goal is to reduce the number of labeled examples needed by using a strategy to prioritize samples worth labeling. A popular strategy is to use predictive uncertainty, prioritizing instances where predictions are most uncertain.

The key goal in active learning is to choose observations $\boldsymbol{x}$ where obtaining labels y would improve learning performance. As discussed in the background, adding samples with high data uncertainty may not improve the trained model because its inherent randomness is irreducible while more samples with model uncertainty can improve the model's performance. In this regard, model uncertainty is more important for active learning [114]. The critical challenge for active learning is to distinguish between the data and model uncertainty and utilize model uncertainty for selecting new samples.

Existing approaches: Similar to OOD detection, approaches for model uncertainty detection can be adapted for active learning by considering uncertainty coming from the parameter, model architecture, and sample density sparsity. For example, the BNN framework considers the samples that decrease the entropy of $p(\boldsymbol{\theta}|\mathcal{D}_{\text{tr}}, \{\boldsymbol{x}, y\})$ the most will be the most useful [34]. The deep ensemble and MC-dropout approach can also be a straightforward way for quantifying the model uncertainty in active learning [61]. Recent approaches propose margin-based uncertainty sampling schemes and provide convergence analysis [124].

5.3 Deep Reinforcement Learning

Deep reinforcement learning (DRL) aims to train an agent interacting with the environment to maximize its total rewards [141]. DRL can be regarded as learning via the Markov Decision Process, defined by the tuple $\{S, A, R, P\}$, where S is the set of states (environment conditions), A is the set of actions (agent), R is the function mapping state-action pairs to rewards, and P is the transition probability (the probability of next state after performing actions on current state). The goal of DRL is to learn the policy π (a function mapping given state to an action) that maximizes the sum of discounted future rewards $J(\pi) = \mathbb{E}[\sum_i \gamma^i R(s_i, a_i)]$, where γ is the discount factor on the future reward (indicating that future rewards are less significant) [141]. In Deep Q-learning, DNN models are used to learn value functions (expected rewards for each state-action pair) given an input state.

Due to the complex agent and environment conditions and limited training states, two types of uncertainty sources exist: data uncertainty and model uncertainty. Data uncertainty arises from the intrinsic randomness in the interactions between the agent and environment, affecting the reward R, the transition functions P, and the next state value distribution. To characterize the data uncertainty arising from those sources, the distributional RL [12] takes a probabilistic perspective on learning the rewards functions instead of approximating the expectation of the value. Thus the approach can be used to implement risk-aware behavior in the agent. A similar approach is proposed to quantify the data uncertainty in DRL aiming for curiosity-based learning in the face of unpredictable transitions [107]. The following work [28] extends the parametric distribution to non-parametric prediction interval methods to quantify the data uncertainty and avoid the explicit parametric format. The approach corresponds to the literature we discuss in section 5.1. On the other hand, given the limited training state space, model uncertainty also exists. The DNN model may not learn the optimum policy function and miss the unexplored state spaces, potentially giving higher rewards. This means the DRL model faces a trade-off between exploitation and exploration. Exploitation means utilizing the model's knowledge and choosing the best policy to maximize future rewards. Exploration involves selecting unexplored states to learn about potential high-reward state-action pairs. The challenge of effective exploration is connected to model uncertainty. The higher model uncertainty means the model is not learned well in the given state and requires more exploration of that sample. For example, the deep ensemble Q-network [25] is proposed to inject model uncertainty into Q learning for more efficient exploration sampling. To reduce computational overhead, the Dropout Q-functions [63] method uses MC-dropout for model uncertainty quantification. The following work [117] demonstrates the previous ensemble and dropout methods may produce a poor approximation to the model uncertainty in cases where state density does not correlate with the true uncertainty. To overcome the shortcoming, they suggest adding a random prior to the ensemble DQNs.

In summary, UQ plays a critical role across various machine-learning problems. In out-of-distribution detection, uncertainty due to domain shift is categorized as model uncertainty rather than data uncertainty. BNN, ensemble, and distance-based methods are well-suited in these cases by capturing model uncertainty through weight distributions or sample embeddings. In active learning, identifying samples with high model uncertainty is essential for improving model training, while samples with high data uncertainty are generally less important for sample selection. Therefore, BNN and ensemble-based approaches play a larger role in active learning. In reinforcement learning, data and model uncertainty are important since both help in efficient policy learning and exploration. Therefore, the combined approach is useful in this context.

6 FUTURE DIRECTION

This section identifies several future directions, including UQ for large language models, UQ for deep learning in scientific simulations, combining UQ and explainability, and UQ for DNN with structured outputs.

6.1 UQ for Large Language Models

In recent years, large language models (LLMs) such as OpenAI's GPT-4 [2] have revolutionized deep learning across diverse tasks such as text summarization, machine translation, complex problem-solving, and even creative writing. However, it is found that an LLM sometimes generates over-confident outputs that are plausible-sounding but factually incorrect, nonsensical, or unsupported by their training data, also called *hallucinations* [21, 41]. Designing UQ methods for LLMs is essential for improving trustworthiness, especially in high-stakes applications. However, several unique challenges exist. Unlike simpler DNN models, LLMs operate in a high-dimensional output space of a long sequence of tokens, making traditional measures like common class entropy insufficient [98]. Furthermore, LLM-generated outputs may vary lexically (different token sequences) but convey the same semantic meaning, requiring UQ that can assess semantic similarity [42].

Table 5. Categorization of Existing Methods for Uncertainty Quantification in LLMs

Strategy	Black-box Methods	White-box Methods
Token probability-based UQ	N/A	Reweight token entropy based on token importance [37], Claim-conditioned token uncertainty for factual claim detection [40]
Self-knowledge-based UQ	Use self-evaluation prompting to elicit a confidence score from the model [23]	Train a separate module using latent representations to predict uncertainty [7]
Sampling-based UQ	Generate multiple outputs and measure response similarity [98]	Analyze response covariance in latent space (e.g., eigenvalues) [21]

Existing methods for UQ in LLMs can be categorized based on their underlying strategies, including token probability-based, self-knowledge-based, and sampling-based UQ. Within each strategy, specific methods can also be divided into black-box-based methods and white-box-based methods, according to whether a method requires access to model internal details [51]. The categorization is summarized in Table 5. The first category, i.e., token-probability-based UQ, is only applicable in white-box settings, as it requires access to token-logit level outputs. One approach focuses on reweighting token class entropy based on the importance of each token [37]. Another method, Claim Conditioned Probability (CCP), quantifies token-level uncertainty specifically for factual claims, filtering out noise from uncertainty about claim formulation [40]. The second category is self-knowledge-based UQ. For black-box methods, self-evaluation is often used to prompt an LLM to produce a confidence score [23]. In white-box scenarios, a separate module can be trained to predict uncertainty scores based on the LLM's latent representations [7]. The third category is sampling-based UQ. In black-box methods, multiple samples are generated to assess response similarity [98]. For white-box approaches, semantic consistency can be examined based on eigenvalues of the response covariance matrix in latent embeddings [21]. Additionally, the conformal prediction has also been used to produce a prediction set of possible outputs that include the correct answer with a specified error rate [170]. Some semantic consistency-based methods are general for both white-box and black-box scenarios. For instance, [42] proposes computing semantic entropy through hidden embeddings or prompt outputs from an LLM.

The field is still in its infancy, and further research is needed to advance UQ for LLM. One research direction is the development of new techniques for calibrating LLMs to produce more accurate confidence estimates. New benchmarking datasets and evaluation metrics for UQ in LLMs are needed. Another direction involves integrating uncertainty estimation directly into LLM's reasoning and decision-making process, e.g., extending chain-of-thought [156] and tree-of-thought [168] with uncertainty quantification. Finally, it is also important to develop uncertainty-aware LLM agents (agentic AI) in complicated and collaborative tasks.

6.2 UQ for deep learning in scientific simulations

Effective and efficient simulation of scientific phenomena, such as extreme weather events, climate change, and tsunamis, often require running physical models [95]. Traditionally, these physical models are based on numerical Partial Differential Equations (PDEs), which are computationally intensive. In recent years, scientific machine (deep) learning has emerged as a new paradigm since data-driven techniques can learn complex patterns from vast amounts of data and make fast predictions with GPUs [83]. UQ for deep learning in scientific simulation is crucial in high-stake decision-making applications (e.g., disaster response). The uncertainty in scientific simulation and modeling can come from different sources. First, the initial and boundary conditions of the physical system are non-deterministic, and the system may be chaotic [152]. Second, the inherent physical principle may not be perfectly known, or the parameter of the governing equation may be stochastic, i.e., in an imperfect physical system, the conservation law of heat may be violated in a non-closed system [167]. Compared to traditional physics-based numerical simulations, diffusion models can generate ensembles of predictions more quickly for uncertainty estimation through probabilistic sampling [95].

Physics-informed neural networks (PINNs) [77] and neural operators [87] are currently two major deep learning techniques for solving PDEs in scientific simulations. PINNs incorporate physical constraints as soft regularization within the loss function, ensuring adherence to governing physical laws. In contrast, neural operators aim to train a neural network surrogate for a family of PDE instances. To enable uncertainty quantification, PINNs, and neural operators are often combined with UQ methods such as Bayesian neural networks and ensemble approaches [122, 131]. However, most existing works often focus on synthetic data instead of complex real-world applications (e.g., physical oceanographers). In recent years, deep generative models, especially diffusion models [64], are increasingly used for real-world scientific simulations such as weather forecasting [48, 95, 120].

There are several potential future research directions. One direction is to decompose different sources of uncertainty, including those from model misspecification, stochasticity, incomplete knowledge of the underlying physical processes, and uncertainties tied to initial conditions, boundary conditions, and external forcings. Second, more efforts are needed for UQ for AI in simulating and forecasting extreme events, such as storm surges [143]. These events are rare (less observational data are available for training) but their societal impacts are very high. Moreover, model outputs can be highly sensitive to inputs (e.g., a small change in the input wind field and air pressures from a hurricane track will make a dramatic difference in output surge levels). Addressing this challenge requires the incorporation of physical knowledge in the UQ framework of the AI surrogate. Another direction is to improve the computational efficiency of AI models such as diffusion models, which are slow for both training and inference due to a large number of iterations [64]. This is of particular importance for high-resolution spatiotemporal simulations. Finally, it is important to design UQ methods for AI in long-term temporal forecasting as error and uncertainty can accumulate over extended time horizons [83].

6.3 Combine UQ with DNN explainability

The explanation for DNN model predictions has been increasingly crucial because it provides tools for understanding and diagnosing the model's prediction. Recently, many explainability methods, termed explainers [151], have been introduced in the category of local feature attribution methods. That is, methods that return a real-valued score for each feature of a given data sample, representing the feature's relative importance for the sample prediction. These explanations are local in that each data sample may have a different explanation. Using local feature attribution methods, therefore, helps users better understand nonlinear and complex black-box models. Both uncertainty quantification and explanation are important for a robust, trustworthy AI model. Current methodologies consider two directions separately, and we believe it could enable a more trustworthy AI system if combined. Though many methodologies have been proposed for more precise uncertainty quantification, very few techniques attempt to explain why uncertainty exists in the predictions.

There are two possible directions to combine the power of explanations and uncertainty quantification: First, existing explanation methods could be potentially improved after considering the prediction uncertainty since those uncertain samples' explanations may not be trustworthy and can be omitted [175]. Second, from another perspective, after obtaining the uncertainty quantification, we can leverage the existing post hoc explanation methods to understand the reason that the model is uncertain [69]. For example, it is intriguing to ask the question of why the prediction is uncertain and which set of input features are uncertain, or due to which layer of the model is imperfect.

6.4 UQ for DNNs with structured outputs

Structured data are samples that are interdependent with each other, violating the common i.i.d assumption [8]. Examples are imaging data, spatiotemporal data, and graphs. Deep learning has been widely used to model structured data, but the uncertainty of its prediction is not often quantified. Here, we list future research directions for the three different types of structured data.

6.4.1 Imaging and inverse problem. The goal of the imaging process is to reconstruct an unknown image from measurements, which is an inverse problem commonly used in medical imaging (e.g., magnetic resonance imaging and X-ray computed tomography) and geophysical inversion (e.g., seismic inversion) [39]. However, this process is challenging due to the limited and noisy information used to determine the original image, leading to structured uncertainty and correlations between nearby pixels in the reconstructed image [78]. To overcome this issue, current research in uncertainty quantification of inverse problems employs conditional deep generative models, such as cVAE, cGAN, and conditional normalizing flow models [35]. These methods utilize a low-dimensional latent space for image generation but may overlook unique data characteristics, such as structural constraints from domain physics in certain types of image data, such as remote sensing images, MRI images, or geological subsurface images [73, 137]. The use of physics-informed models may improve uncertainty quantification in these cases. It's promising to incorporate the physics constraints for quantifying the uncertainty associated with the imaging process.

6.4.2 Spatiotemporal data. Spatiotemporal data are special due to the violation of the common assumption that samples follow an identical and independent distribution [70, 136]. Uncertainty quantification of spatiotemporal deep learning poses several unique challenges. First, the analysis of spatiotemporal data requires the co-registration of different maps (e.g., points, lines, polygons, geo-raster) into the same spatial reference system. The process is subject to registration uncertainty due to GPS errors or annotation mistakes in map generation [71]. Such registration uncertainty causes troubles when training deep neural networks [59]. Second, implicit dependency structures exist

in continuous space and time (e.g., spatial and temporal autocorrelation, and temporal dynamics). Thus, the uncertainty quantification process should be aware of such a dependency structure. Third, spatiotemporal non-stationary requires characterizing uncertainty due to out-of-distribution samples [70, 136]. In addition, a different level of uncertainty exists based on the nearby training sample density. Traditionally, the Gaussian process has been widely used to quantify spatial uncertainty. However, for deep neural network models, new techniques are needed that consider sample density both in the non-spatial feature space and in the geographic space.

6.4.3 Graph data. Graph data is a general type of structured data with nodes and edge connections. Graph neural networks (GNNs) have been widely used for graph applications related to node classification and edge (link) prediction. However, UQ for GNN models has been less explored. Some work utilizes existing UQ techniques for GNN models [43] without considering their unique characteristics. First, predictions on a graph are structured, so the UQ module needs to consider such structural dependency. Second, many GNN models assume a fixed graph topology from training and test instances (e.g., spectral-based methods [32]). Uncertainty in GNN predictions arises from shifts in graph topology between training and test graphs. Similarly, uncertainty exists when the graph is perturbed by removing nodes and edges. Finally, many real-world graph problems are spatiotemporal at the same time (e.g., traffic flow prediction on road networks). Thus, challenges related to UQ for spatiotemporal deep learning also apply to graphs.

7 CONCLUSION

This paper presents a systematic survey on uncertainty quantification for DNNs based on the types of uncertainty sources. We categorize the existing literature into three groups: model uncertainty, data uncertainty, and their combination. Additionally, we analyze the strengths and weaknesses of each approach based on the specific type of uncertainty it addresses. We also summarize the sources of uncertainty and the unique challenges faced across various applications, and ML problems, and propose several future research directions.

REFERENCES

[1] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. 2021. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion* 76 (2021), 243–297.

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[3] Firoj Alam, Muhammad Imran, and Ferda Ofli. 2017. Image4act: Online social media image processing for disaster response. In *Proceedings of the 2017 IEEE/ACM international conference on advances in social networks analysis and mining 2017*. 601–604.

[4] Martin Andrae, Tomas Landelius, Joel Oskarsson, and Fredrik Lindsten. 2024. Continuous Ensemble Weather Forecasting with Diffusion models. *arXiv preprint arXiv:2410.05431* (2024).

[5] Anastasios N Angelopoulos, Stephen Bates, et al. 2023. Conformal prediction: A gentle introduction. *Foundations and Trends® in Machine Learning* 16, 4 (2023), 494–591.

[6] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*. PMLR, 214–223.

[7] Amos Azaria and Tom Mitchell. 2023. The Internal State of an LLM Knows When It's Lying. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

[8] Gökhan BakIr, Thomas Hofmann, Alexander J Smola, Bernhard Schölkopf, and Ben Taskar. 2007. *Predicting structured data*. MIT press.

[9] Wentao Bao, Qi Yu, and Yu Kong. 2021. Evidential deep learning for open set action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 13349–13358.

[10] Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Guanghui Liu, Amit Raj, et al. 2024. Lumiere: A space-time diffusion model for video generation. *arXiv preprint arXiv:2401.12945*

(2024).
[11] Edmon Begoli, Tanmoy Bhattacharya, and Dimitri Kusnezov. 2019. The need for uncertainty quantification in machine-assisted medical decision making. *Nature Machine Intelligence* 1, 1 (2019), 20–23.
[12] Marc G Bellemare, Will Dabney, and Rémi Munos. 2017. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*. PMLR, 449–458.
[13] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* 35, 8 (2013), 1798–1828.
[14] Viktor Bengs, Eyke Hüllermeier, and Willem Waegeman. 2023. On second-order scoring rules for epistemic uncertainty quantification. In *International Conference on Machine Learning*. PMLR, 2078–2091.
[15] Lucas Berry, Axel Brando, and David Meger. 2024. Shedding Light on Large Generative Networks: Estimating Epistemic Uncertainty in Diffusion Models. In *The 40th Conference on Uncertainty in Artificial Intelligence*.
[16] Christopher M Bishop. 1994. Mixture density networks. (1994).
[17] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. 2017. Variational inference: A review for statisticians. *Journal of the American statistical Association* 112, 518 (2017), 859–877.
[18] Michele Caprio, Souradeep Dutta, Kuk Jin Jang, Vivian Lin, Radoslav Ivanov, Oleg Sokolsky, and Insup Lee. 2023. Imprecise Bayesian neural networks. *arXiv preprint arXiv:2302.09656* (2023).
[19] Jie Chang, Zhonghao Lan, Changmao Cheng, and Yichen Wei. 2020. Data uncertainty learning in face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5710–5719.
[20] Bertrand Charpentier, Daniel Zügner, and Stephan Günnemann. 2020. Posterior network: Uncertainty estimation without ood samples via density-based pseudo-counts. *Advances in Neural Information Processing Systems* 33 (2020), 1356–1367.
[21] Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. 2024. INSIDE: LLMs' Internal States Retain the Power of Hallucination Detection. In *The Twelfth International Conference on Learning Representations*.
[22] Haoxian Chen, Ziyi Huang, Henry Lam, Huajie Qian, and Haofeng Zhang. 2021. Learning prediction intervals for regression: Generalization and calibration. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 820–828.
[23] Jiuhai Chen and Jonas Mueller. 2024. Quantifying uncertainty in answers from any language model and enhancing their trustworthiness. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 5186–5200.
[24] Xiang Chen, Andres Diaz-Pinto, Nishant Ravikumar, and Alejandro F Frangi. 2021. Deep learning in medical image registration. *Progress in Biomedical Engineering* 3, 1 (2021), 012003.
[25] Xinyue Chen, Che Wang, Zijian Zhou, and Keith W Ross. 2021. Randomized Ensembled Double Q-Learning: Learning Fast Without a Model. In *International Conference on Learning Representations*.
[26] Reynold Cheng, Tobias Emrich, Hans-Peter Kriegel, Nikos Mamoulis, Matthias Renz, Goce Trajcevski, and Andreas Züfle. 2014. Managing uncertainty in spatial and spatio-temporal data. In *2014 IEEE 30th International Conference on Data Engineering*. IEEE, 1302–1305.
[27] Jiwoong Choi, Dayoung Chun, Hyun Kim, and Hyuk-Jae Lee. 2019. Gaussian yolov3: An accurate and fast object detector using localization uncertainty for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 502–511.
[28] Will Dabney, Mark Rowland, Marc Bellemare, and Rémi Munos. 2018. Distributional reinforcement learning with quantile regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.
[29] Andreas Damianou. 2015. *Deep Gaussian processes and variational propagation of uncertainty*. Ph. D. Dissertation. University of Sheffield.
[30] Andreas C Damianou, Michalis K Titsias, and Neil Lawrence. 2016. Variational inference for latent variables and uncertain inputs in Gaussian processes. (2016).
[31] Arka Daw, M Maruf, and Anuj Karpatne. 2021. PID-GAN: A GAN Framework based on a Physics-informed Discriminator for Uncertainty Quantification with Physics. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 237–247.
[32] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems* 29 (2016).
[33] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.
[34] Stefan Depeweg. 2019. *Modeling epistemic and aleatoric uncertainty with bayesian neural networks and latent variables*. Ph. D. Dissertation. Technische Universität München.
[35] Garoe Dorta, Sara Vicente, Lourdes Agapito, Neill DF Campbell, and Ivor Simpson. 2018. Structured uncertainty prediction networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5477–5485.

[36] Zhekai Du and Jingjing Li. 2023. Diffusion-based probabilistic uncertainty estimation for active domain adaptation. *Advances in Neural Information Processing Systems* 36 (2023), 17129–17155.

[37] Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. 2024. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 5050–5063.

[38] Michael W Dusenberry, Dustin Tran, Edward Choi, Jonas Kemp, Jeremy Nixon, Ghassen Jerfel, Katherine Heller, and Andrew M Dai. 2020. Analyzing the role of model uncertainty for electronic health records. In *Proceedings of the ACM Conference on Health, Inference, and Learning*. 204–213.

[39] Vineet Edupuganti, Morteza Mardani, Shreyas Vasanawala, and John Pauly. 2020. Uncertainty quantification in deep MRI reconstruction. *IEEE Transactions on Medical Imaging* 40, 1 (2020), 239–250.

[40] Ekaterina Fadeeva, Aleksandr Rubashevskii, Artem Shelmanov, Sergey Petrakov, Haonan Li, Hamdy Mubarak, Evgenii Tsymbalov, Gleb Kuzmin, Alexander Panchenko, Timothy Baldwin, et al. 2024. Fact-checking the output of large language models via token-level uncertainty quantification. *arXiv preprint arXiv:2403.04696* (2024).

[41] Ekaterina Fadeeva, Roman Vashurin, Akim Tsvigun, Artem Vazhentsev, Sergey Petrakov, Kirill Fedyanin, Daniil Vasilev, Elizaveta Goncharova, Alexander Panchenko, Maxim Panov, et al. 2023. LM-Polygraph: Uncertainty Estimation for Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 446–461.

[42] Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature* 630, 8017 (2024), 625–630.

[43] Boyuan Feng, Yuke Wang, and Yufei Ding. 2021. Uag: Uncertainty-aware attention graph neural network for defending adversarial attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 7404–7412.

[44] Marc Anton Finzi, Anudhyan Boral, Andrew Gordon Wilson, Fei Sha, and Leonardo Zepeda-Núñez. 2023. User-defined event sampling and uncertainty quantification in diffusion models for physical dynamical systems. In *International Conference on Machine Learning*. PMLR, 10136–10152.

[45] Karl Friston, Jérémie Mattout, Nelson Trujillo-Barreto, John Ashburner, and Will Penny. 2007. Variational free energy and the Laplace approximation. *Neuroimage* 34, 1 (2007), 220–234.

[46] Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*. PMLR, 1050–1059.

[47] Han Gao, Xu Han, Xiantao Fan, Luning Sun, Li-Ping Liu, Lian Duan, and Jian-Xun Wang. 2024. Bayesian conditional diffusion models for versatile spatiotemporal turbulence generation. *Computer Methods in Applied Mechanics and Engineering* 427 (2024), 117023.

[48] Han Gao, Sebastian Kaltenbach, and Petros Koumoutsakos. 2024. Generative learning for forecasting the dynamics of high-dimensional complex systems. *Nature Communications* 15, 1 (2024), 8904.

[49] Yihang Gao and Michael K Ng. 2022. Wasserstein generative adversarial uncertainty quantification in physics-informed neural networks. *J. Comput. Phys.* 463 (2022), 111270.

[50] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. 2023. A survey of uncertainty in deep neural networks. *Artificial Intelligence Review* 56, Suppl 1 (2023), 1513–1589.

[51] Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. 2024. A Survey of Confidence Estimation and Calibration in Large Language Models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Kevin Duh, Helena Gomez, and Steven Bethard (Eds.). Association for Computational Linguistics, Mexico City, Mexico, 6577–6595. https://doi.org/10.18653/v1/2024.naacl-long.366

[52] Biraja Ghoshal, Allan Tucker, Bal Sanghera, and Wai Lup Wong. 2021. Estimating uncertainty in deep learning for reporting confidence to clinicians in medical image segmentation and diseases detection. *Computational Intelligence* 37, 2 (2021), 701–734.

[53] Xuan Gong, Luckyson Khaidem, Wentao Zhu, Baochang Zhang, and David Doermann. 2022. Uncertainty learning towards unsupervised deformable medical image registration. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2484–2493.

[54] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.

[55] Cornelia Gruber, Patrick Oliver Schenk, Malte Schierholz, Frauke Kreuter, and Göran Kauermann. 2023. Sources of Uncertainty in Machine Learning–A Statisticians' View. *arXiv preprint arXiv:2305.16703* (2023).

[56] Axel Brando Guillaumes. 2017. *Mixture density networks for distribution and uncertainty estimation.* Ph. D. Dissertation. Universitat Politècnica de Catalunya. Facultat d'Informàtica de Barcelona.

[57] C Guo, G Pleiss, Y Sun, and KQ Weinberger. 2017. On calibration of modern neural networks, international conference on machine learning. *PMLR* (2017), 1321–1330.

[58] Reihaneh H Hariri, Erik M Fredericks, and Kate M Bowers. 2019. Uncertainty in big data analytics: survey, opportunities, and challenges. *Journal of Big Data* 6, 1 (2019), 1–16.

[59] Wenchong He, Zhe Jiang, Marcus Kriby, Yiqun Xie, Xiaowei Jia, Da Yan, and Yang Zhou. 2022. Quantifying and Reducing Registration Uncertainty of Spatial Vector Labels on Earth Imagery. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 554–564.

[60] Yotam Hechtlinger, Barnabás Póczos, and Larry Wasserman. 2018. Cautious deep learning. *arXiv preprint arXiv:1805.09460* (2018).

[61] Alice Hein, Stefan Röhrl, Thea Grobel, Manuel Lengl, Nawal Hafez, Martin Knopp, Christian Klenk, Dominik Heim, Oliver Hayden, and Klaus Diepold. 2022. A Comparison of Uncertainty Quantification Methods for Active Learning in Image Classification. In *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

[62] Dan Hendrycks and Kevin Gimpel. 2022. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. In *International Conference on Learning Representations*.

[63] Takuya Hiraoka, Takahisa Imagawa, Taisei Hashimoto, Takashi Onishi, and Yoshimasa Tsuruoka. 2022. Dropout Q-Functions for Doubly Efficient Reinforcement Learning. In *International Conference on Learning Representations*.

[64] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.

[65] Paul Hofman, Yusuf Sale, and Eyke Hüllermeier. 2024. Quantifying Aleatoric and Epistemic Uncertainty with Proper Scoring Rules. *arXiv preprint arXiv:2404.12215* (2024).

[66] Eyke Hüllermeier and Willem Waegeman. 2021. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning* 110, 3 (2021), 457–506.

[67] Jiahao Ji, Jingyuan Wang, Zhe Jiang, Jiawei Jiang, and Hu Zhang. 2022. STDEN: Towards physics-guided neural networks for traffic flow prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 4048–4056.

[68] Xiaowei Jia, Jacob Zwart, Jeffrey Sadler, Alison Appling, Samantha Oliver, Steven Markstrom, Jared Willard, Shaoming Xu, Michael Steinbach, Jordan Read, et al. 2021. Physics-guided recurrent graph model for predicting flow and temperature in river networks. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*. SIAM, 612–620.

[69] Junji Jiang, Chen Ling, Hongyi Li, Guangji Bai, Xujiang Zhao, and Liang Zhao. 2024. Quantifying uncertainty in graph neural network explanations. *Frontiers in big Data* 7 (2024), 1392662.

[70] Zhe Jiang. 2018. A survey on spatial prediction methods. *IEEE transactions on knowledge and Data Engineering* 31, 9 (2018), 1645–1664.

[71] Zhe Jiang, Wenchong He, Marcus Kirby, Sultan Asiri, and Da Yan. 2021. Weakly Supervised Spatial Deep Learning based on Imperfect Vector Labels with Registration Errors. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 767–775.

[72] Zhe Jiang, Arpan Man Sainju, Yan Li, Shashi Shekhar, and Joseph Knight. 2019. Spatial ensemble learning for heterogeneous geographic data with class ambiguity. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10, 4 (2019), 1–25.

[73] Zhe Jiang and Shashi Shekhar. 2017. Spatial big data science. *Schweiz: Springer International Publishing AG* (2017).

[74] Laurent Valentin Jospin, Hamid Laga, Farid Boussaid, Wray Buntine, and Mohammed Bennamoun. 2022. Hands-on Bayesian neural networks—A tutorial for deep learning users. *IEEE Computational Intelligence Magazine* 17, 2 (2022), 29–48.

[75] AUDUN. JSANG. 2018. *Subjective Logic: A formalism for reasoning under uncertainty*. Springer.

[76] Hamed Karimi and Reza Samavi. 2023. Quantifying deep learning model uncertainty in conformal prediction. In *Proceedings of the AAAI Symposium Series*, Vol. 1. 142–148.

[77] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. 2021. Physics-informed machine learning. *Nature Reviews Physics* 3, 6 (2021), 422–440.

[78] Alex Kendall and Yarin Gal. 2017. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems* 30 (2017).

[79] Abbas Khosravi, Saeid Nahavandi, Doug Creighton, and Amir F Atiya. 2010. Lower upper bound estimation method for construction of neural network-based prediction intervals. *IEEE transactions on neural networks* 22, 3 (2010), 337–346.

[80] Hyunsu Kim, Jongmin Yoon, and Juho Lee. 2024. Fast Ensembling with Diffusion Schrödinger Bridge. In *The Twelfth International Conference on Learning Representations*.

[81] Sookyung Kim, Hyojin Kim, Joonseok Lee, Sangwoong Yoon, Samira Ebrahimi Kahou, Karthik Kashinath, and Mr Prabhat. 2019. Deep-hurricane-tracker: Tracking and forecasting extreme climate events. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 1761–1769.

[82] Diederik P Kingma and Max Welling. 2014. Auto-encoding variational bayes. (2014).
[83] Dmitrii Kochkov, Janni Yuval, Ian Langmore, Peter Norgaard, Jamie Smith, Griffin Mooers, Milan Klöwer, James Lottes, Stephan Rasp, Peter Düben, et al. 2024. Neural general circulation models for weather and climate. *Nature* 632, 8027 (2024), 1060–1066.
[84] Simon Kohl, Bernardino Romera-Paredes, Clemens Meyer, Jeffrey De Fauw, Joseph R Ledsam, Klaus Maier-Hein, SM Eslami, Danilo Jimenez Rezende, and Olaf Ronneberger. 2018. A probabilistic u-net for segmentation of ambiguous images. *Advances in neural information processing systems* 31 (2018).
[85] Nikita Kotelevskii, Aleksandr Artemenkov, Kirill Fedyanin, Fedor Noskov, Alexander Fishkov, Artem Shelmanov, Artem Vazhentsev, Aleksandr Petiushko, and Maxim Panov. 2022. Nonparametric uncertainty quantification for single deterministic neural network. *Advances in Neural Information Processing Systems* 35 (2022), 36308–36323.
[86] Nikita Kotelevskii and Maxim Panov. 2024. Predictive Uncertainty Quantification via Risk Decompositions for Strictly Proper Scoring Rules. *arXiv preprint arXiv:2402.10727* (2024).
[87] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. 2023. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research* 24, 89 (2023), 1–97.
[88] Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. 2021. Learnable uncertainty under laplace approximations. In *Uncertainty in Artificial Intelligence*. PMLR, 344–353.
[89] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems* 30 (2017).
[90] Olivier Laurent, Adrien Lafage, Enzo Tartaglione, Geoffrey Daniel, Jean-marc Martinez, Andrei Bursuc, and Gianni Franchi. 2023. Packed Ensembles for efficient uncertainty estimation. In *The Eleventh International Conference on Learning Representations*.
[91] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436–444.
[92] Hong Joo Lee, Jung Uk Kim, Sangmin Lee, Hak Gu Kim, and Yong Man Ro. 2020. Structure boundary preserving segmentation for medical image with ambiguous boundary. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4817–4826.
[93] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems* 31 (2018).
[94] Junyi Li, Tianyi Tang, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2021. Pre-trained Language Models for Text Generation: A Survey. *Comput. Surveys* (2021).
[95] Lizao Li, Robert Carver, Ignacio Lopez-Gomez, Fei Sha, and John Anderson. 2024. Generative emulation of weather forecast ensembles with diffusion models. *Science Advances* 10, 13 (2024), eadk4489.
[96] Yiqun Li, Songjian Chai, Guibin Wang, Xian Zhang, and Jing Qiu. 2022. Quantifying the Uncertainty in Long-Term Traffic Prediction Based on PI-ConvLSTM Network. *IEEE Transactions on Intelligent Transportation Systems* 23, 11 (2022), 20429–20441.
[97] Richard J Licata and Piyush M Mehta. 2022. Uncertainty quantification techniques for data-driven space weather modeling: thermospheric density application. *Scientific Reports* 12, 1 (2022), 7256.
[98] Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2024. Generating with Confidence: Uncertainty Quantification for Black-box Large Language Models. *Transactions on Machine Learning Research* (2024). https://openreview.net/forum?id=DWkJCSxKU5
[99] Jeremiah Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax Weiss, and Balaji Lakshminarayanan. 2020. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in Neural Information Processing Systems* 33 (2020), 7498–7512.
[100] Jeremiah Zhe Liu, Shreyas Padhy, Jie Ren, Zi Lin, Yeming Wen, Ghassen Jerfel, Zachary Nado, Jasper Snoek, Dustin Tran, and Balaji Lakshminarayanan. 2022. A simple approach to improve single-model deep uncertainty via distance-awareness. *Journal of Machine Learning Research* 23 (2022), 1–63.
[101] Tyler J Loftus, Benjamin Shickel, Matthew M Ruppert, Jeremy A Balch, Tezcan Ozrazgat-Baslanti, Patrick J Tighe, Philip A Efron, William R Hogan, Parisa Rashidi, Gilbert R Upchurch Jr, et al. 2022. Uncertainty-aware deep learning in healthcare: a scoping review. *PLOS digital health* 1, 8 (2022), e0000085.
[102] Christos Louizos and Max Welling. 2017. Multiplicative normalizing flows for variational bayesian neural networks. In *International Conference on Machine Learning*. PMLR, 2218–2227.
[103] Charles Lu, Andréanne Lemay, Ken Chang, Katharina Höbel, and Jayashree Kalpathy-Cramer. 2022. Fair conformal predictors for applications in medical imaging. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 12008–12016.
[104] Andrey Malinin and Mark Gales. 2018. Predictive uncertainty estimation via prior networks. *Advances in neural information processing systems* 31 (2018).

[105] Tanwi Mallick, Prasanna Balaprakash, and Jane Macfarlane. 2022. Deep-Ensemble-Based Uncertainty Quantification in Spatiotemporal Graph Neural Networks for Traffic Forecasting. *arXiv preprint arXiv:2204.01618* (2022).
[106] Christos Markos, JQ James, and Richard Yi Da Xu. 2021. Capturing uncertainty in unsupervised GPS trajectory segmentation using Bayesian deep learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 390–398.
[107] Augustine Mavor-Parker, Kimberly Young, Caswell Barry, and Lewis Griffin. 2022. How to stay curious while avoiding noisy tvs using aleatoric uncertainty estimation. In *International Conference on Machine Learning*. PMLR, 15220–15240.
[108] José Mena, Oriol Pujol, and Jordi Vitrià. 2021. A survey on uncertainty estimation in deep learning classification systems from a bayesian perspective. *ACM Computing Surveys (CSUR)* 54, 9 (2021), 1–35.
[109] Zhaobin Mo and Yongjie Fu. 2022. TrafficFlowGAN: Physics-informed Flow based Generative Adversarial Network for Uncertainty Quantification. In *European Conference on Machine Learning and Data Mining (ECML PKDD)*.
[110] Kevin P Murphy. 2012. *Machine learning: a probabilistic perspective*. MIT press.
[111] Radford M Neal. 2012. *Bayesian learning for neural networks*. Vol. 118. Springer Science & Business Media.
[112] Marion Neumeier, Sebastian Dorn, Michael Botsch, and Wolfgang Utschick. 2024. Reliable Trajectory Prediction and Uncertainty Quantification with Conditioned Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3461–3470.
[113] Andre T Nguyen, Fred Lu, Gary Lopez Munoz, Edward Raff, Charles Nicholas, and James Holt. 2022. Out of distribution data detection using dropout bayesian neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 7877–7885.
[114] Vu-Linh Nguyen, Mohammad Hossein Shaker, and Eyke Hüllermeier. 2022. How to measure uncertainty in uncertainty sampling for active learning. *Machine Learning* 111, 1 (2022), 89–122.
[115] Philipp Oberdiek, Gernot Fink, and Matthias Rottmann. 2022. Uqgan: A unified model for uncertainty quantification of deep classifiers trained via conditional gans. *Advances in Neural Information Processing Systems* 35 (2022), 21371–21385.
[116] Shu Lih Oh, Yuki Hagiwara, U Raghavendra, Rajamanickam Yuvaraj, N Arunkumar, M Murugappan, and U Rajendra Acharya. 2020. A deep learning approach for Parkinson's disease diagnosis from EEG signals. *Neural Computing and Applications* 32, 15 (2020), 10927–10933.
[117] Ian Osband, John Aslanides, and Albin Cassirer. 2018. Randomized prior functions for deep reinforcement learning. *Advances in Neural Information Processing Systems* 31 (2018).
[118] Tim Pearce, Alexandra Brintrup, Mohamed Zaki, and Andy Neely. 2018. High-quality prediction intervals for deep learning: A distribution-free, ensembled approach. In *International conference on machine learning*. PMLR, 4075–4084.
[119] Konstantin Posch, Jan Steinbrener, and Jürgen Pilz. 2019. Variational inference to measure model uncertainty in deep neural networks. *arXiv preprint arXiv:1902.10189* (2019).
[120] Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Tom R Andersson, Andrew El-Kadi, Dominic Masters, Timo Ewalds, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, et al. 2023. Gencast: Diffusion-based ensemble forecasting for medium-range weather. *arXiv preprint arXiv:2312.15796* (2023).
[121] Sergey Prokudin, Peter Gehler, and Sebastian Nowozin. 2018. Deep directional statistics: Pose estimation with uncertainty quantification. In *Proceedings of the European conference on computer vision (ECCV)*. 534–551.
[122] Apostolos F Psaros, Xuhui Meng, Zongren Zou, Ling Guo, and George Em Karniadakis. 2023. Uncertainty quantification in scientific machine learning: Methods, metrics, and comparisons. *J. Comput. Phys.* 477 (2023), 111902.
[123] Yu Qin, Zhiwen Liu, Chenghao Liu, Yuxing Li, Xiangzhu Zeng, and Chuyang Ye. 2021. Super-Resolved q-Space deep learning with uncertainty quantification. *Medical Image Analysis* 67 (2021), 101885.
[124] Anant Raj and Francis Bach. 2022. Convergence of uncertainty sampling for active learning. In *International Conference on Machine Learning*. PMLR, 18310–18331.
[125] Markus Reichstein, Gustau Camps-Valls, Bjorn Stevens, Martin Jung, Joachim Denzler, and Nuno Carvalhais. 2019. Deep learning and process understanding for data-driven Earth system science. *Nature* 566, 7743 (2019), 195–204.
[126] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. 2021. A survey of deep active learning. *ACM computing surveys (CSUR)* 54, 9 (2021), 1–40.
[127] Hippolyt Ritter, Aleksandar Botev, and David Barber. 2018. Online structured laplace approximations for overcoming catastrophic forgetting. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. 3742–3752.
[128] Hippolyt Ritter, Aleksandar Botev, and David Barber. 2018. A scalable laplace approximation for neural networks. In *6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings*, Vol. 6. International Conference on Representation Learning.
[129] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10684–10695.

[130] Nir Rosenfeld, Yishay Mansour, and Elad Yom-Tov. 2018. Discriminative learning of prediction intervals. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 347–355.
[131] Izzet Sahin, Christian Moya, Amirhossein Mollaali, Guang Lin, and Guillermo Paniagua. 2024. Deep operator learning-based surrogate models with uncertainty quantification for optimizing internal cooling channel rib profiles. *International Journal of Heat and Mass Transfer* 219 (2024), 124813.
[132] Tárik S Salem, Helge Langseth, and Heri Ramampiaro. 2020. Prediction intervals: Split normal mixture from quality-driven deep ensembles. In *Conference on Uncertainty in Artificial Intelligence*. PMLR, 1179–1187.
[133] Tim Salimans, Diederik Kingma, and Max Welling. 2015. Markov chain monte carlo and variational inference: Bridging the gap. In *International conference on machine learning*. PMLR, 1218–1226.
[134] Murat Sensoy, Lance Kaplan, and Melih Kandemir. 2018. Evidential deep learning to quantify classification uncertainty. *Advances in neural information processing systems* 31 (2018).
[135] Sina Shafaei, Stefan Kugele, Mohd Hafeez Osman, and Alois Knoll. 2018. Uncertainty in machine learning: A safety perspective on autonomous driving. In *Computer Safety, Reliability, and Security: SAFECOMP 2018 Workshops, ASSURE, DECSoS, SASSUR, STRIVE, and WAISE, Västerås, Sweden, September 18, 2018, Proceedings 37*. Springer, 458–464.
[136] Shashi Shekhar, Zhe Jiang, Reem Y Ali, Emre Eftelioglu, Xun Tang, Venkata MV Gunturi, and Xun Zhou. 2015. Spatiotemporal data mining: A computational perspective. *ISPRS International Journal of Geo-Information* 4, 4 (2015), 2306–2338.
[137] Shu-Fu Shih, Sevgi Gokce Kafali, Kara L Calkins, and Holden H Wu. 2022. Uncertainty-aware physics-driven deep learning network for free-breathing liver fat and R2* quantification using self-gated stack-of-radial MRI. *Magnetic Resonance in Medicine* (2022).
[138] Dule Shu and Amir Barati Farimani. 2024. Zero-Shot Uncertainty Quantification using Diffusion Probabilistic Models. *arXiv preprint arXiv:2408.04718* (2024).
[139] Edward Snelson and Zoubin Ghahramani. 2005. Sparse Gaussian processes using pseudo-inputs. *Advances in neural information processing systems* 18 (2005).
[140] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems* 28 (2015).
[141] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.
[142] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. 2021. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems* 34 (2021), 24804–24816.
[143] Claudia Tebaldi, Katharinec Hayhoe, Julie M Arblaster, and Gerald A Meehl. 2006. Going to the extremes: an intercomparison of model-simulated historical and future changes in extreme events. *Climatic change* 79 (2006), 185–211.
[144] Sunil Thulasidasan, Gopinath Chennupati, Jeff A Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. 2019. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. *Advances in Neural Information Processing Systems* 32 (2019).
[145] Michalis Titsias. 2009. Variational learning of inducing variables in sparse Gaussian processes. In *Artificial intelligence and statistics*. PMLR, 567–574.
[146] Trung Q Trinh, Markus Heinonen, Luigi Acerbi, and Samuel Kaski. 2022. Tackling covariate shift with node-based Bayesian neural networks. In *International Conference on Machine Learning*. PMLR, 21751–21775.
[147] Ivan Ustyuzhaninov, Ieva Kazlauskaite, Markus Kaiser, Erik Bodin, Neill Campbell, and Carl Henrik Ek. 2020. Compositional uncertainty in deep Gaussian processes. In *Conference on Uncertainty in Artificial Intelligence*. PMLR, 480–489.
[148] Joost van Amersfoort, Lewis Smith, Andrew Jesson, Oscar Key, and Yarin Gal. 2021. On Feature Collapse and Deep Kernel Learning for Single Forward Pass Uncertainty. *arXiv preprint arXiv:2102.11409* (2021).
[149] Joost Van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. 2020. Uncertainty estimation using a single deep deterministic neural network. In *International conference on machine learning*. PMLR, 9690–9700.
[150] Aladin Virmaux and Kevin Scaman. 2018. Lipschitz regularity of deep neural networks: analysis and efficient estimation. *Advances in Neural Information Processing Systems* 31 (2018).
[151] Minh Vu and My T Thai. 2020. Pgm-explainer: Probabilistic graphical model explanations for graph neural networks. *Advances in neural information processing systems* 33 (2020), 12225–12235.
[152] Bin Wang, Jie Lu, Zheng Yan, Huaishao Luo, Tianrui Li, Yu Zheng, and Guangquan Zhang. 2019. Deep uncertainty quantification: A machine learning approach for weather forecasting. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2087–2095.
[153] Jian Wang, Wei Deng, and Yuntao Guo. 2014. New Bayesian combination method for short-term traffic flow forecasting. *Transportation Research Part C: Emerging Technologies* 43 (2014), 79–94.
[154] Pengyue Wang, Yan Li, Shashi Shekhar, and William F Northrop. 2019. Uncertainty estimation with distributional reinforcement learning for applications in intelligent transportation systems: A case study. In *2019 IEEE Intelligent*

Transportation Systems Conference (ITSC). IEEE, 3822–3827.

[155] Zi Wang, Alexander Ku, Jason Baldridge, Tom Griffiths, and Been Kim. 2023. Gaussian process probes (GPP) for uncertainty-aware probing. *Advances in neural information processing systems* 36 (2023), 63573–63594.

[156] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

[157] Florian Wenzel, Jasper Snoek, Dustin Tran, and Rodolphe Jenatton. 2020. Hyperparameter ensembles for robustness and uncertainty quantification. *Advances in Neural Information Processing Systems* 33 (2020), 6514–6527.

[158] Veit David Wild, Sahra Ghalebikesabi, Dino Sejdinovic, and Jeremias Knoblauch. 2024. A rigorous link between deep ensembles and (variational) Bayesian methods. *Advances in Neural Information Processing Systems* 36 (2024).

[159] Christopher KI Williams and Carl Edward Rasmussen. 2006. *Gaussian processes for machine learning*. Vol. 2. MIT press Cambridge, MA.

[160] Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. 2016. Deep kernel learning. In *Artificial intelligence and statistics*. PMLR, 370–378.

[161] Christopher Wolf, Maximilian Karl, and Patrick van der Smagt. 2016. Variational inference with hamiltonian monte carlo. *arXiv preprint arXiv:1609.08203* (2016).

[162] Dongxia Wu, Liyao Gao, Matteo Chinazzi, Xinyue Xiong, Alessandro Vespignani, Yi-An Ma, and Rose Yu. 2021. Quantifying Uncertainty in Deep Spatiotemporal Forecasting. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1841–1851.

[163] Chen Xu and Yao Xie. 2023. Conformal prediction for time series. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 10 (2023), 11575–11587.

[164] Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. Hallucination is inevitable: An innate limitation of large language models. *arXiv preprint arXiv:2401.11817* (2024).

[165] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. 2023. Diffusion models: A comprehensive survey of methods and applications. *Comput. Surveys* 56, 4 (2023), 1–39.

[166] Xueying Yang, Jiamian Wang, Xujiang Zhao, Sheng Li, and Zhiqiang Tao. 2022. Calibrate automated graph neural network via hyperparameter uncertainty. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 4640–4644.

[167] Yibo Yang and Paris Perdikaris. 2019. Adversarial uncertainty quantification in physics-informed neural networks. *J. Comput. Phys.* 394 (2019), 136–152.

[168] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems* 36 (2024).

[169] Gal Yarin. 2016. Uncertainty in deep learning. *University of Cambridge, Cambridge* (2016).

[170] Fanghua Ye, Mingming Yang, Jianhui Pang, Longyue Wang, Derek F Wong, Emine Yilmaz, Shuming Shi, and Zhaopeng Tu. 2024. Benchmarking LLMs via Uncertainty Quantification. *arXiv preprint arXiv:2401.12794* (2024).

[171] Jiaxuan You, Xiaocheng Li, Melvin Low, David Lobell, and Stefano Ermon. 2017. Deep gaussian process for crop yield prediction based on remote sensing data. In *Thirty-First AAAI conference on artificial intelligence*.

[172] Margaux Zaffran, Olivier Féron, Yannig Goude, Julie Josse, and Aymeric Dieuleveut. 2022. Adaptive conformal predictions for time series. In *International Conference on Machine Learning*. PMLR, 25834–25866.

[173] Sheheryar Zaidi, Arber Zela, Thomas Elsken, Chris C Holmes, Frank Hutter, and Yee Teh. 2021. Neural ensemble search for uncertainty estimation and dataset shift. *Advances in Neural Information Processing Systems* 34 (2021), 7898–7911.

[174] Soroush H Zargarbashi, Simone Antonelli, and Aleksandar Bojchevski. 2023. Conformal prediction sets for graph neural networks. In *International Conference on Machine Learning*. PMLR, 12292–12318.

[175] Xiaoge Zhang, Felix TS Chan, and Sankaran Mahadevan. 2022. Explainable machine learning in image classification models: An uncertainty quantification perspective. *Knowledge-Based Systems* 243 (2022), 108418.