# Lightning UQ Box: A Comprehensive Framework for Uncertainty Quantification in Deep Learning

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

Uncertainty quantification (UQ) is an essential tool for applying deep neural networks (DNNs) to real world tasks, as it attaches a degree of confidence to DNN outputs. However, despite its benefits, UQ is often left out of the standard DNN workflow due to the additional technical knowledge required to apply and evaluate existing UQ procedures. Hence there is a need for a comprehensive toolbox that allows the user to integrate UQ into their modelling workflow, without significant overhead. We introduce `Lightning UQ Box`: a unified interface for applying and evaluating various approaches to UQ. In this paper, we provide a theoretical and quantitative comparison of the wide range of state-of-the-art UQ methods implemented in our toolbox. We focus on two challenging vision tasks: (i) estimating tropical cyclone wind speeds from infrared satellite imagery and (ii) estimating the power output of solar panels from RGB images of the sky. By highlighting the differences between methods our results demonstrate the need for a broad and approachable experimental framework for UQ, that can be used for benchmarking UQ methods. The toolbox, example implementations, and further information are available at: https://github.com/lightning-uq-box/lightning-uq-box.

## 1 Introduction

In real world applications, deep learning (DL) models are often deployed in safety-critical domains such as healthcare [45], robotics [56], and Earth observation [55, 60], with relevant areas including flood monitoring [7], wildfire mapping and forecasting [58], and weather forecasting [59]. In these fields, an incorrect prediction can cause significant damage and corresponding consequences. Uncertainty quantification (UQ) aims to provide a measure of confidence about a neural network's prediction and to support practitioners in identifying potentially false predictions to better guide analyses and decision-making processes [21]. Besides this, UQ can even improve predictive performance via regularization [16, 39].

The direct application of UQ to DL is often not straightforward for practitioners. Besides the implementation challenges associated with probabilistic modelling and stochastic training algorithms, the performance of UQ methods can fluctuate, depending on the data and the task [50]. Moreover, there is a lack of clear guidance on which methods are promising for specific tasks, given the ever-increasing zoo of UQ methods for DL [1, 21]. These challenges are particularly prominent for data modalities of higher dimensions, such as vision, where uncertainty modelling adds a further layer of complexity. Therefore, various approaches need to be considered, which usually involve different loss functions, training procedures, and model architectures. The need of accessible and open-source UQ frameworks is also called upon in a recent position paper on Bayesian Deep Learning (BDL) by leading experts in this field [53]: "Software development is key to encouraging DL practitioners to use Bayesian methods. More generally, there is a need for software that would make it easier for

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

practitioners to try BDL in their projects. The use of BDL must become competitive in human effort with standard deep learning." [53].

`Lightning UQ Box` provides users with all the tools needed to equip deep neural networks (DNNs) with UQ. We created `Lightning UQ Box` to tackle the gap between theoretical researchers and actual practitioners in the field of UQ in DL. The toolbox offers a comprehensive framework, building on top of `PyTorch` [54] and `Lightning` [18], as an accessible end-to-end solution. The toolbox is particularly suited for vision applications (see Section 3): it offers flexible layer configurations like Bayesian convolution layers that can be modularly placed in backbone architectures, which streamline UQ.

We underline the usefulness of the presented toolbox with two example applications: estimating the maximum sustained wind speed of tropical cyclones from satellite imagery and predicting the power voltage output of solar panels from a time series of sky images. These applications contain different sources and types of uncertainties in the input and target variables and illustrate the stochastic nature of real world phenomena and measurement systems practitioners are confronted with. Simultaneously, these applications carry an associated inherent risk that demands reliable predictive uncertainties. The central contributions of our work all aim to equip practitioners with the necessary tools to apply UQ methods for DL on their specific (real world) problem:

- **Comprehensive End-to-End UQ Toolbox:** `Lightning UQ Box` enables practitioners to efficiently iterate over ideas without having to re-implement the provided UQ methods. To do so, it provides backbone architecture- and dataset-agnostic implementations of a wide array of UQ methods and corresponding evaluation schemes for DL, covering regression, classification, semantic segmentation, and pixel-wise regression tasks.
- **Adaptability and Expandability:** The modular implementation using `Lightning` encourages practitioners and the community to an individual adaptation and a continuous expansion and growth of the toolbox. Additionally, the implementation is adapted to vector or vision data. Specifically, partial stochasticity [65] is supported when applicable. This supports any larger architectures used for vision, and the "frozen" functionality enables retraining only a few layers.
- **Practical and Theoretical Introductions:**The toolbox contains comprehensive practical and theoretical introductions to the field of UQ and the application of the toolbox. UQ Tutorials and case studies on designing downstream tasks to compare various UQ methods are provided. A comprehensive theory guide provides methodological backgrounds on the implemented methods.

**Related Work** Frameworks for UQ in DL already exist in the `PyTorch` [54] ecosystem. However, they are limited to either a handful of UQ methods or a specific class of approaches, such as BDL. Several libraries exist for BDL, most notably `TorchBNN` [38], `BLiTZ` [17], and `Bayesian-Torch` [35]. Yet BNNs are only one approach to UQ and require choosing a prior distribution. When an abundance of data is available, frequentist procedures, such as conformal prediction, can be a more attractive alternative. The library `Fortuna` [14] supports several approaches to conformal prediction (CP), of which we currently support a subset (with plans to incorporate more). The primary difference between our work and `Fortuna` is that `Fortuna` is primarily compatible with `JAX` [9] and only supports post-hoc calibration of `PyTorch` models. `TorchCP` [71] is another framework that implements conformal prediction [4], but it does not support other UQ methods (such as BDL). The most closely related package to ours is `torch-uncertainty` [36], which implements both frequentist and Bayesian UQ methods in addition to common benchmarks. Yet our `Lightning UQ Box`, to date, implements the largest number of UQ methods across different theoretical frameworks, such as BDL and CP, while including cutting-edge techniques as partially stochastic networks [65], and additionally supports UQ methods for semantic segmentation tasks. Table 1 gives a comparison with previous libraries.

## 2 Benchmarking UQ Methods: the Lightning UQ Box

The underlying design of `Lightning UQ Box` is based on three pillars:

- provide a comprehensive set of reference implementations of state-of-the-art UQ methods,
- optimally fit in the wide open-source landscape for DL based on `PyTorch`, and
- enhance automation, scalability, and reproducibility of experiments with `Lightning`.

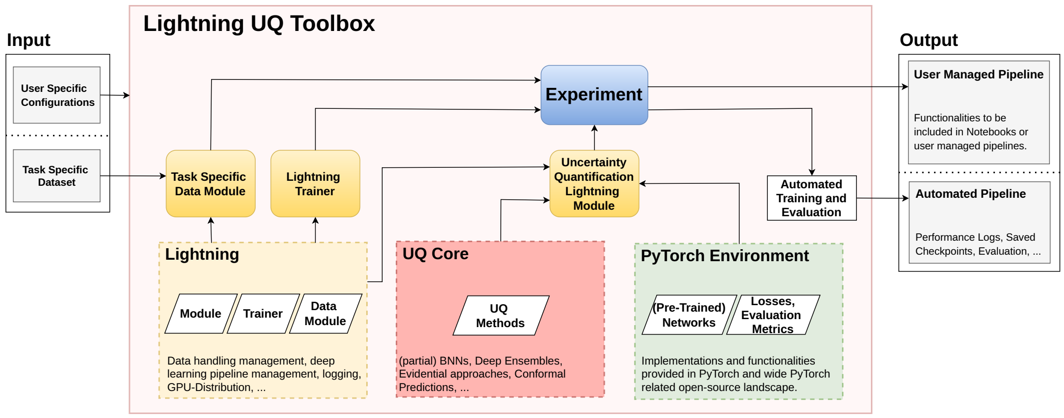



Figure 1: The structure of `Lightning UQ Box`. The experiments can be built and evaluated at scale or manually tailored to specific use cases. For large experiments at scale, only a dataset and a configuration file have to be provided.

These design goals are reflected in the structure of the toolbox, as visualized in Figure 1, and build up on the three core components of the available DL functionalities provided within the `Lightning` framework for structuring and pipeline managing, the UQ Core, which contains the UQ method implementations, and the `PyTorch` ecosystem.

The UQ Core contains a comprehensive collection of UQ methods for DL with different theoretical underpinnings consolidated and implemented for this toolbox. The theoretical backgrounds are very diverse and cover, for example, mean-field estimation and various Bayesian-motivated approaches, including kernel-based approaches and partially stochastic networks, ensemble methods, and evidence-motivated approaches (cmp. Section 2.1). Besides the diversity in methodological approaches, the toolbox provides unified interfaces and configuration patterns, thereby improving accessibility and, importantly, enabling comparability between the methods.

The toolbox is compatible with common DL libraries and frameworks from the `PyTorch` ecosystem. The provided UQ methods can be combined with user-specific architectures and implementations provided in the `PyTorch` ecosystem, including pre-trained networks and foundation models. This is especially useful as our framework can build upon or be included in existing code and pipelines based on `PyTorch`-based libraries, such as `timm` [72]. In order to scale BDL to modern-sized architectures, we offer functionality to convert existing deterministic architectures, or specified components thereof, automatically to a Bayesian framework. As a result, the collection of UQ methods goes beyond mere method compilation, offering not only comprehensiveness but also removing time-consuming implementation overhead. This enables users to use the UQ toolbox as a simple extension of their existing DL pipelines.

The toolbox utilizes the `Lightning` framework to enhance experiment automation, scalability, and reproducibility. `Lightning` offers a flexible and user-friendly interface for the automated management of complex pipelines. It is specifically designed to support practitioners in managing experiments by providing functionalities to enhance their scalability and reproducibility. These include managing configurations, training loops, evaluation steps, and logging processes. To this end, each UQ method is implemented as a `LightningModule` that can be used with a `LightningDataModule` and a `Trainer` to execute training, evaluation, and inference for a desired task. The toolbox also utilizes the `Lightning` command line interface (CLI) for better experiment reproducibility and for setting up experiments at scale. This provides an end-to-end configuration, such that a full pipeline of experiments can be built with minimal overhead. Many optional configurations and user-specific objects, such as logging functionalities or models, can be included but are not mandatory. The general concept of the toolbox is illustrated in Figure 1.

### 2.1 Provided Types of UQ Methods

`Lightning UQ Box` provides the most comprehensive collection of the extensive and versatile landscape of UQ methods for DL. The following section gives an overview of these different UQ

Table 1: The methods provided with `Lightning UQ Box` and other available frameworks and reviews. The table represents the status at the time of publication and will be extended in the future. All currently available methods can be found in the provided repository.

| Publication | [26] | [63] | [15] | [31] | [64] | [50] | [46] | [36] | `Lightning UQ Box` |
|---:|---|---|---|---|---|---|---|---|---|
| **Deterministic Methods** | | | | | | | | | |
| Gaussian (MVE) | ✓ | | | | | | | ✓ | ✓ |
| Deep Evidential Networks (DER) | | | | | | | | ✓ | ✓ |
| **Neural Network Ensembles** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| **Bayesian Neural Networks** | | | | | | | | | |
| MC Dropout (GMM) | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| BNN with VI ELBO | | | | ✓ | ✓ | ✓ | ✓ | | ✓ |
| BNN with VI (alpha divergence) | | | | | | | | | ✓ |
| VBLL | | | | | | | | | ✓ |
| Laplace Approximation | | | | | ✓ | | | | ✓ |
| SWAG | | | | ✓ | ✓ | | | | ✓ |
| DVI, SI | | | | ✓ | | | | | |
| HMC | | | | ✓ | | | | | |
| Radial BNN | | | | | | | ✓ | | |
| Rank-1 BNN | | | | | | | ✓ | | |
| **Gaussian Process based** | | | | | | | | | |
| Deep Kernel Learning (DKL) | | | | | | | | | ✓ |
| Det. Unc. Estimation (DUE) | | | | | | | | | ✓ |
| Spectral Normalized GPs (SNGP) | | | | | ✓ | | ✓ | | ✓ |
| **Quantile based** | | | | | | | | | |
| Quantile Regression (QR) | ✓ | | ✓ | | | | | | ✓ |
| Conformal Prediction (CQR) | ✓ | | ✓ | | | | | | ✓ |
| **Diffusion Model** | | | | | | | | | |
| CARD | | | | | | | | | ✓ |
| **Post-hoc Calibration** | | | | | | | | | |
| RAPS | | | | | | | | | ✓ |
| TempScaling | | | | | | ✓ | | ✓ | ✓ |

methods, which are listed in Table 1. For comprehensive explanations, we refer to the theory guide in the supplement and to existing reviews [1, 21]. For **regression tasks** NNs predict a continuous target $y^\star$. Currently, the toolbox supports six classes of UQ methods for regression: deterministic, quantile, ensemble, Bayesian, Gaussian Process, and diffusion-based methods.

1. Deterministic methods: use a DNN, $f_\theta : X \to \mathcal{P}(Y)$, that map inputs $x$ to the parameters of a probability distribution $f_\theta(x^\star) = p_\theta(x^\star) \in \mathcal{P}(Y)$, and include methods like Deep Evidential Regression (**DER**) [2] and Mean Variance Estimation (**MVE**) [49]. The latter, for example, outputs the mean and standard deviation of a Gaussian distribution $f_\theta^{\mathrm{MVE}}(x^\star) = (\mu_\theta(x^\star), \sigma_\theta(x^\star))$.
2. Quantile based models: use a DNN, $f_\theta : X \to Y^n$, that map to $n$ quantiles, $f_\theta(x^\star) = (q_1(x^\star), ..., q_n(x^\star)) \in Y^n$, and include Quantile Regression [33] (**Quantile Regression**) and the conformalized version thereof (**ConformalQR**) [62].
3. Ensembles: Deep Ensembles [37], which utilize an ensemble over MVE networks.
4. Bayesian methods: model the network parameters as random variables. Multiple principles and techniques to approximate BNNs have been introduced. We include BNNs with Variational Inference (VI) (**BNN VI ELBO**) [8], BNNs with VI and alpha divergence (**BNN VI**) [13], Variational Bayesian Last Layers (**VBLL**) [28], MC-Dropout (**MCDropout**) [20], the Laplace Approximation (**Laplace**) [61][12] and **SWAG** [43] with partially stochastic variants [65].
5. Gaussian Process-based methods: these model a joint distribution over a set of functions in a data-driven manner that approximates the first and second moment of the marginalized distribution. These include Deep Kernel Learning (**DKL**) [73], an extension thereof Deterministic Uncertainty Estimation (**DUE**) [69, 70], and Spectral Normalized Gaussian Process (**SNGP**) [40].
6. Conditional Generative model: Classification and Regression Diffusion (**CARD**) [27].

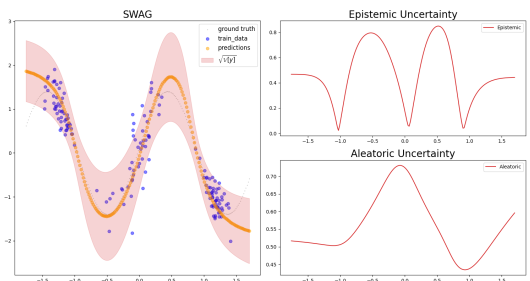
For **classification**, the toolbox currently supports six classes of UQ methods. Vanilla softmax probabilities can be directly used to obtain predictive uncertainties. However, they are often miscalibrated and have lead to post-hoc recalibration methods being proposed [25].

1. Deep Ensembles (**DeepEnsembles**) [37]: utilize an ensemble over independent standard classification networks.
2. Bayesian methods: **BNN VI ELBO** [8], **VBLL** [28], **MCDropout** [20], **Laplace** [61][12], **SWAG** [43].
3. Gaussian Process based methods: **DKL** [73], **DUE** [69] and Spectral-normalized Neural Gaussian Processes (**SNGP**) [40].
4. Conformal Prediction: [62], Regularized Adaptive Prediction Sets (**RAPS**) [3].
5. Other: Test-time Augmentation (**TTA**) [41], Temperature Scaling [25].

Additionally to the general purpose tasks of regression and classification, `Lightning UQ Box` supports UQ methods for vision-specific tasks. These include segmentation and pixel-wise regression, where an extensive overview of supported UQ methods can be found on our documentation page.

```python
dm = ToyDataModule()
swag_model = SWAGRegression(
    deterministic_model.model,
    num_mc_samples=50,
    swag_lr=1e-3,
    loss_fn=NLL(),
)
swag_trainer = Trainer(
    max_epochs=20,
)
swag_trainer.fit(swag_model, datamodule=dm)
preds = swag_model.predict_step(X_test)
```

(a) Example code to fit SWAG method.




(b) SWAG regression toy example.

Figure 2: Example code and visualization on toy regression dataset.

**Quantifying Predictive Uncertainty:** By default, we quantify predictive uncertainty via the standard deviation for regression and via the entropy of the predictive distribution for classification. In general, for UQ in DL, two main types of uncertainties can be considered: aleatoric and epistemic [13, 21]. Aleatoric uncertainty originates from random, or partially observable, effects in the data itself and is not reducible: for instance, the Earth covered with clouds does not contain enough information to surely assign the land cover type to one of multiple given options. In contrast, epistemic uncertainty quantifies the model's predictive uncertainty originating from uncertainty over its parameters: it will typically shrink as more data becomes available [30]. See Figure 2b for an example decomposition. Depending on the underlying theoretical assumptions, UQ methods model these types of uncertainties individually or mutually [30]. From a statistical perspective, Gruber et al. [24] allude that such a distinction is often not possible. Thus, in the examples given here, we focus on the approximate predictive distributions of the UQ methods $p_\theta(y_\star|x_\star)$, from which we derive the aforementioned uncertainty measures. However, where applicable, the toolbox also enables researchers to decompose these two types of uncertainties.

**Limitations:** Despite the robustness and versatility of the `Lightning UQ Box`, it is tightly integrated within the `PyTorch` ecosystem, limiting its applicability to other existing DL frameworks like `Tensorflow` and `JAX`. Furthermore, merely using UQ methods does not guarantee complete reliability, and applications nevertheless require proper experimental design and evaluation.

## 3 Experimental Setup for Validation

We now showcase `Lightning UQ Box` as a valuable tool for conducting experimental studies including benchmarking. We exemplify this by comparing UQ methods on three challenging computer vision datasets from two different domains. More concretely, we evaluate the methods on selected downstream tasks that highlight the efficacy of UQ and the usefulness of a unified framework[1]. Each experiment was completed using the UQ toolbox in less than 10 hours (6 hours on average) on a single A100 40GB GPU.

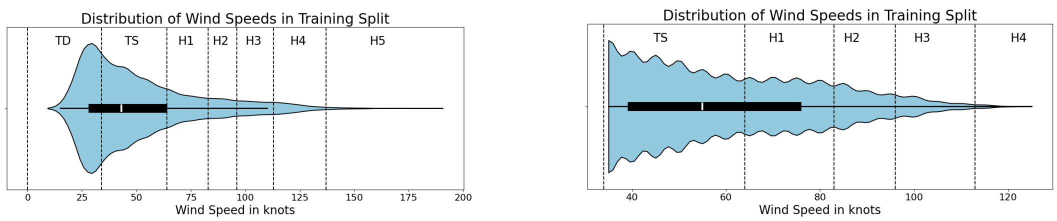
[1]Code for all experiments available at this link: Github Repo.

### 3.1 Datasets

For our experiments, we consider three datasets: the Tropical Cyclone Driven Data Challenge dataset (TC) [44], the Digital Typhoon (DT) dataset [32], and the SKy Images and Photovoltaic Power Generation Dataset (SKIPP'D) [47]. An overview of the datasets is given in Table 2. For a detailed explanations of the datasets see supplementary section 1.

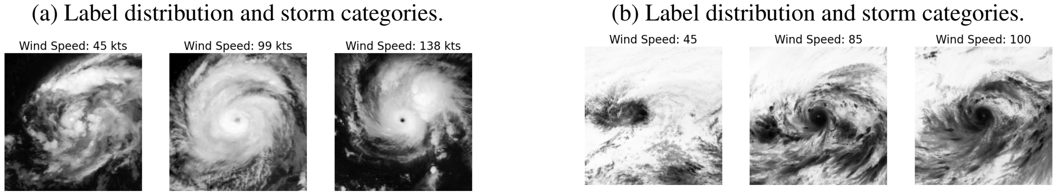
Table 2: Dataset Overview.

| Dataset | Image source/Satellite | Spatial Res. | Temporal Res. | Train Samples | Val. Samples | Test Samples |
|---|---|---|---|---|---|---|
| Tropical Cyclone | GOES | 2 km | 15 min | 53k | 11k | 43k |
| Digital Typhoon | Himawari | 5 km | 60 min | 64.5k | 14k | 20k |
| SKIPP'D | Fisheye camera | - | 1 min | 280k | 63k | 14k |




(a) Label distribution and storm categories.

(b) Label distribution and storm categories.

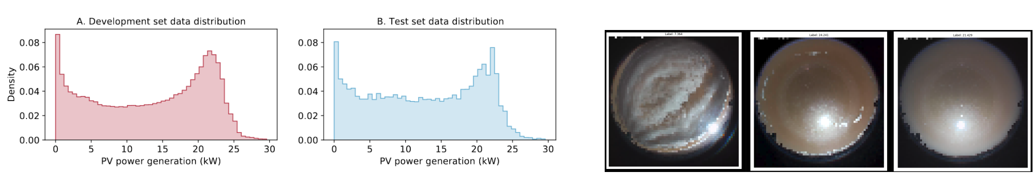



(c) Samples from the Tropical Cyclon Dataset.

(d) Samples from the Digital Typhoon Dataset.

Figure 3: Visualization of the Tropical Cyclon (left) and the Digital Typhoon Dataset (right).

**Cyclone and Typhoon Dataset:** The TC and DT datasets consist of infrared measurements that capture the spatial structure of storms. Corresponding wind speed targets are matched based on hurricane databases. There are varying sources of uncertainty in the inputs, such as missing pixels due to the swath of the satellites, and in the targets, such as measurement uncertainties and interpolations over time with respect to non-uniform time steps. As such, these datasets exemplify real world stochastic phenomena, where predictive uncertainties are essential for decision-making due to the inherent risk associated with these potentially extreme events. The magnitude of rapid intensification events has been increasing [6], thus causing more damage if not properly detected and predicted. One such recent example is Hurricane Otis in October 2023, where existing models had to disproportionally rely on satellite data, due to limited in-situ data, which lead to erroneous forecasts [34]. Given the extensive availability of satellite imagery, research efforts using this modality are a promising avenue to enhance existing forecasts.




(a) Statistics of SKIPP'D test and train set [47].

(b) Example Image of the SKIPP'D dataset.

Figure 4: Visualization of SKIPP'D Dataset.

**Photovoltaic Dataset:** The SKIPP'D dataset consists of ground-based fish-eye RGB images over a 3-year period (2017–2019), where associated targets are power output measurements from a 30 kilowatt (kW) rooftop photovoltaic array [47]. Given the urgent necessity to transform the world's energy sector to more sustainable solutions [5], this dataset aims to support research efforts of large-scale integration of power voltage into electricity grids, where the main problem is to manage the non-constant and intermittent power source [47].

### 3.2 Methodological Setup

**Cyclone and Typhoon Dataset:** Various works have framed the wind speed estimation of tropical cyclones from a satellite image as both a regression [10, 42, 75] and classification [57, 74] task. We apply all UQ methods provided by the toolbox to the regression and classification task. For all wind speed experiments, we use the same ResNet-18 [29] pre-trained on ImageNet[2] as the backbone architecture of compared UQ methods. For the TC and DT datasets, the chosen task is selective prediction, as introduced by Geifman et al. [22]. Here, samples with a predictive uncertainty above a given threshold are omitted and can be referred to domain experts or further decision-making pipelines. If the corresponding UQ method has higher uncertainties for inaccurate predictions, leaving out the predictions for these samples should increase the overall accuracy, indicating a correlation between predictive uncertainty and model error. This can be beneficial in a deployment setting where automated analysis systems are paired with human expertise. Examples are visualized in Figure 6.

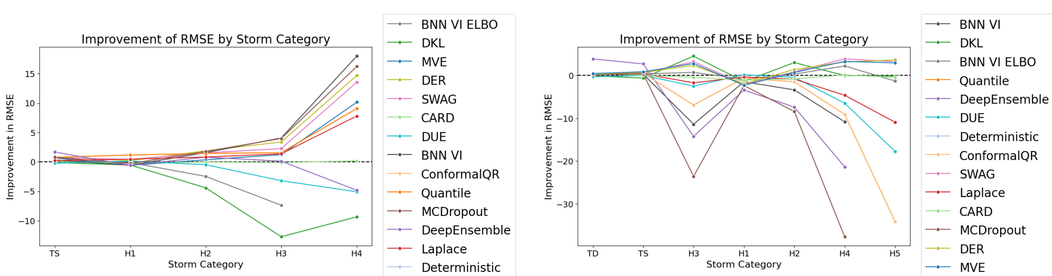
**Photovoltaic Dataset:** Previous work have demonstrated promising results of such image data for photovoltaic power generation estimation modeled as a regression task [67, 76, 68, 48, 19, 51, 52]. We apply all UQ methods provided by the toolbox (see Table 1) to this regression task. Here, we use the proposed CNN architecture of Nie et al. [47], which requires only a single line code change in experiment configuration for each respective UQ method.[3] Given the central problem of photovoltaics being a non-constant power source, we analyze the additional benefits of UQ by evaluating predictive uncertainty on annotated sunny and cloudy days. From a reliable model, we expect that both the predictive error as well as the predictive uncertainty is larger on the cloudy samples because the partial occlusions make it more difficult to estimate the corresponding power voltage output.

**Evaluation Metrics:** As evaluation metrics, we use the root mean squared error (RMSE), as well as proper scoring rules such as the negative log-likelihood (NLL) [23]. Furthermore, we also consider the mean absolute calibration error (MACE) and correlation between the predictive uncertainties and mean absolute error (MAE).[4] A detailed description of the employed metrics is in the supplementary.

## 4 Results: Examples of UQ Method Analysis

The following section provides a quantitative performance comparison of different UQ methods under a possible benchmark setting, easily enabled by our proposed framework.

### 4.1 Selective Prediction for Wind Speed Estimation




(a) (b)

Figure 5: Selective Prediction RMSE improvement per category on the Digital Typhoon Dataset (left) and Tropical Cyclone Dataset (right).

Table 3 shows that most UQ methods improve model accuracy when applying selective prediction with respect to a deterministic baseline, which cannot express any uncertainty. Compared to Table 3, Figure 5 demonstrates a different ranking of the UQ methods, with respect to the accuracy improvement due to selective prediction, when evaluated per category, according to the Saffir-Simpson scale [66]. This ranking also differs on the DT and TC dataset, as Figure 5 shows. The skewed data distribution of both datasets, 3a and 3b, and the different uncertainty sources in the

[2] As available in the `timm` library [72]

[3] More examples are shown in the Github Repo for these experiments.

[4] Metrics computed with the library provided by [11]

TC and DT datasets 3.1 may contribute to these observations of aggregation pathologies. For the classification task the ranking of methods varies with Gaussian Process based methods performing better, see supplementary section 2.

Table 3: Evaluation of Regression Results on the test set. Note that [64] observe a similar ranking in terms of accuracy, also with respect to Deep Ensembles. RMSE $\Delta$ shows the improvement after selective prediction, while Coverage denotes the fraction of remaining samples that were not omitted. Selective prediction is based on the 0.8 quantile of predictive uncertainties on a validation set.

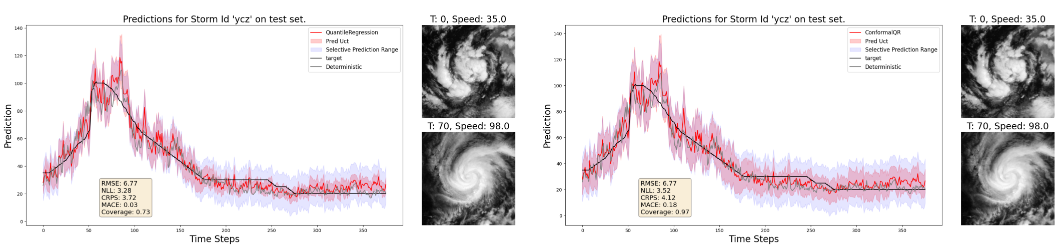
| UQ group | Method | RMSE ↓ | RMSE Δ ↑ | NLL ↓ | MACE ↓ |
|---|---|---|---|---|---|
| None | Deterministic | 9.64 | 0.00 | NaN | NaN |
| Deterministic | MVE | 10.10 | 0.64 | 3.74 | 0.06 |
| | DER | 9.59 | **1.07** | 4.32 | 0.30 |
| Quantile | QR | 9.54 | **1.03** | **3.64** | 0.05 |
| | CQR | 9.54 | **1.03** | 3.72 | 0.10 |
| Ensemble | Deep Ensemble | 14.37 | 0.77 | 4.05 | **0.01** |
| Bayesian | MC Dropout | 9.77 | **1.03** | 3.75 | 0.10 |
| | SWAG | **9.10** | 0.97 | 3.67 | 0.12 |
| | Laplace | 9.64 | 0.44 | 3.69 | 0.03 |
| | BNN VI ELBO | **9.15** | 0.17 | 15.82 | 0.35 |
| | BNN VI | 10.74 | 0.94 | 3.76 | 0.03 |
| | SNGP | 9.33 | -0.05 | 14.00 | 0.36 |
| | VBLL | 9.72 | 0.06 | 3.70 | 0.03 |
| | DKL | 10.35 | -0.31 | 3.77 | **0.01** |
| | DUE | 9.46 | -0.10 | 3.68 | **0.01** |
| Diffusion | CARD | 9.57 | 0.09 | 9.35 | 0.30 |

(a) Digital Typhoon Dataset.

| UQ group | Method | RMSE ↓ | RMSE Δ ↑ | NLL ↓ | MACE ↓ |
|---|---|---|---|---|---|
| None | Deterministic | 10.50 | 0.00 | NaN | NaN |
| Deterministic | MVE | 9.95 | 1.15 | **3.64** | 0.04 |
| | DER | 10.14 | 1.17 | 4.60 | 0.35 |
| Quantile | QR | 10.95 | **1.05** | 3.73 | **0.01** |
| | CQR | 10.95 | **1.05** | 3.79 | 0.10 |
| Ensemble | Deep Ensemble | 16.19 | **3.30** | 4.15 | 0.05 |
| Bayesian | MC Dropout | 10.23 | 0.87 | 3.81 | 0.16 |
| | SWAG | 9.78 | 1.13 | 3.71 | 0.13 |
| | Laplace | 10.53 | 0.60 | 4.31 | 0.28 |
| | BNN VI ELBO | 11.82 | 1.56 | 5.57 | 0.23 |
| | BNN VI | 11.20 | 1.45 | 3.74 | 0.02 |
| | SNGP | 12.01 | 0.28 | 5.53 | 0.18 |
| | VBLL | 10.79 | 0.51 | 3.80 | 0.07 |
| | DKL | 12.59 | 0.21 | 3.95 | 0.06 |
| | DUE | 9.95 | -0.21 | 3.73 | 0.08 |
| Diffusion | CARD | 10.86 | 0.45 | 3.92 | 0.05 |

(b) Tropical Cyclone Dataset.

Figure 6 gives a visual intuition of the selective prediction scheme. If the predictive uncertainty (red-shaded region) exceeds the established threshold (blue-shaded region), individual predictions are deferred to an expert. The models provide a reasonable mean estimate of a storm track, even though predictions are made for single image instances and the regression task is modeled by ResNet-18 without a notion of time. Figure 6 additionally showcases the effect of conformalizing the predictive uncertainty of an underlying model. Conformal prediction aims to calibrate prediction sets while providing theoretical coverage guarantees; it can be particularly interesting in the case of a high-risk task such as wind speed estimation. Figure 6b demonstrates the effectiveness of the procedure, as the coverage has increased from 0.73 to 0.97, which is also reflected in the wider prediction intervals that cover the targets without sacrificing any accuracy.




(a) Quantile Regression.

(b) Conformalized Quantile Regression.

Figure 6: Individual nowcasting predictions are stitched together to recover a time series. Areas where the red-shaded regions exceed the blue denote samples that *would* be omitted during selective prediction. The example showcases the effect of the conformal procedure, where conformalized prediction intervals increase the desired empirical coverage.

### 4.2 Photovoltaic Power Output Estimation Under Cloudy and Sunny Conditions

Figure 7 demonstrates that model performance differs under cloudy or sunny conditions. Across methods the NLL demonstrates differences in the model performance and related calibration between cloudy and sunny days. The consideration of uncertainty improved the accuracy of models compared to the deterministic baseline, as shown in the supplementary material. The correlation between the model error (in terms of MAE) and the predictive uncertainty shows a clear positive correlation (>0.45) across all methods. However, there are differences in the magnitude between methods and cloud conditions. Stakeholders might prefer good UQ estimates on more complex days, i.e., the cloudy ones, than for sunny days, where the output is much easier to predict. Exhaustive results can be found in the supplementary material.

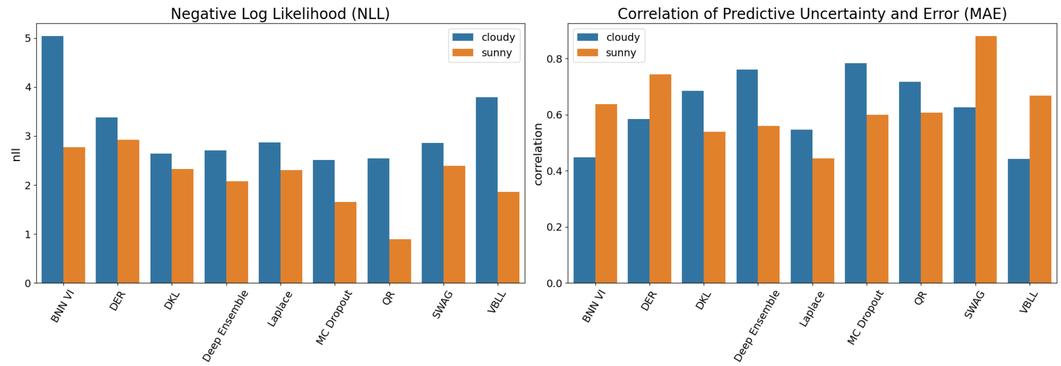
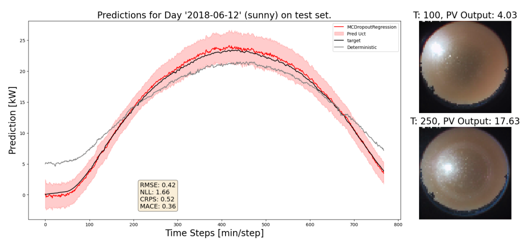
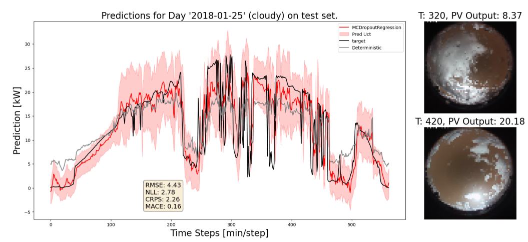



Figure 7: Negative Log Likelihood (left) and correlation between model error (measured by MAE) and predictive uncertainty for different methods on cloudy and sunny test examples.

Figure 8 showcases concrete examples with power voltage estimates plotted over the duration of a cloudy and a non-cloudy day. Compared to the smooth and consistent power output on a sunny day 8a, the predictive uncertainty is larger under cloudy conditions. This may reflect the uncertainty in the input images due to cloudiness changing faster than the time step resolution.




(a) MC Dropout prediction: sunny day example.




(b) MC Dropout prediction: cloudy day example.

Figure 8: Individual nowcasting predictions stitched together to recover a time series. The plot shows qualitative and quantitative differences between the two methods for the same set of predictions.

# 5 Conclusion

We have introduced `Lightning UQ Box`, a comprehensive framework for enhancing neural networks with uncertainty estimates. Additionally, we have showcased its usefulness for comparing a broad range of methods from different theoretical foundations on three relevant tasks with various sources of uncertainty. Our framework not only makes it easier for practitioners to use Bayesian methods for DL as demanded by [53] but goes beyond this by supporting UQ methods stemming from various theoretical frameworks and assumptions. Our experimental results demonstrate the differences and variability between UQ methods and, therefore, the benefit of this benchmarking framework. In conclusion, our open-source framework and the accompanying resources can be both an entry point for researchers to the field of UQ and also aid the development of new methods that address the shortcomings of existing ones [50].

# 6 Ethics and Broader Impact Statement

Including UQ in DL applied to real world and safety critical applications is of significant importance. UQ can provide the means to reduce risks, yet practitioners should not succumb to a false sense of security provided by such methods. The performance and reliability of UQ methods may be dataset and task dependent. Exactly for that reason we provide our framework under the open-source Apache-2.0 license to support open science, transparency, and collaborative research efforts.

## 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. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information fusion*, 76:243–297, 2021.

[2] Alexander Amini, Wilko Schwarting, Ava Soleimany, and Daniela Rus. Deep evidential regression. *Advances in Neural Information Processing Systems*, 33:14927–14937, 2020.

[3] Anastasios Angelopoulos, Stephen Bates, Jitendra Malik, and Michael I Jordan. Uncertainty sets for image classifiers using conformal prediction. *arXiv preprint arXiv:2009.14193*, 2020.

[4] Anastasios N Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2021.

[5] Svante Arrhenius. Xxxi. on the influence of carbonic acid in the air upon the temperature of the ground. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 41(251):237–276, 1896.

[6] Karthik Balaguru, Gregory R Foltz, and L Ruby Leung. Increasing magnitude of hurricane rapid intensification in the central and eastern tropical atlantic. *Geophysical Research Letters*, 45(9):4238–4247, 2018.

[7] Roberto Bentivoglio, Elvin Isufi, Sebastian Nicolaas Jonkman, and Riccardo Taormina. Deep learning methods for flood mapping: a review of existing applications and future research directions. *Hydrology and earth system sciences*, 26(16):4345–4378, 2022.

[8] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015.

[9] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: Composable transformations of Python+NumPy programs, 2018.

[10] Buo-Fu Chen, Boyo Chen, Hsuan-Tien Lin, and Russell L Elsberry. Estimating tropical cyclone intensity by satellite imagery utilizing convolutional neural networks. *Weather and Forecasting*, 34(2):447–465, 2019.

[11] Youngseog Chung, Ian Char, Han Guo, Jeff Schneider, and Willie Neiswanger. Uncertainty toolbox: an open-source library for assessing, visualizing, and improving uncertainty quantification. *arXiv preprint arXiv:2109.10254*, 2021.

[12] Erik Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. Laplace redux-effortless bayesian deep learning. *Advances in Neural Information Processing Systems*, 34:20089–20103, 2021.

[13] Stefan Depeweg, Jose-Miguel Hernandez-Lobato, Finale Doshi-Velez, and Steffen Udluft. Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning. In *International Conference on Machine Learning*, pages 1184–1193. PMLR, 2018.

[14] Gianluca Detommaso, Alberto Gasparin, Michele Donini, Matthias Seeger, Andrew Gordon Wilson, and Cedric Archambeau. Fortuna: A library for uncertainty quantification in deep learning. *arXiv preprint arXiv:2302.04019*, 2023.

[15] Nicolas Dewolf, Bernard De Baets, and Willem Waegeman. Valid prediction intervals for regression problems. *Artificial Intelligence Review*, pages 1–37, 2022.

[16] Codruț-Andrei Diaconu and Nina Maria Gottschling. Uncertainty-aware learning with label noise for glacier mass balance modelling. *IEEE Geoscience and Remote Sensing Letters*, 2024.

[17] Piero Esposito. BLiTZ - Bayesian Layers in Torch Zoo (a Bayesian deep learing library for Torch). `https://github.com/piEsposito/blitz-bayesian-deep-learning/`, 2020.

[18] William A. Falcon. PyTorch Lightning. `https://github.com/Lightning-AI/pytorch-lightning`, 2019.

[19] Cong Feng and Jie Zhang. Solarnet: A sky image-based deep convolutional neural network for intra-hour solar forecasting. *Solar Energy*, 204:71–78, 2020.

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

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

[22] Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. *Advances in neural information processing systems*, 30, 2017.

[23] Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

[24] Cornelia Gruber, Patrick Oliver Schenk, Malte Schierholz, Frauke Kreuter, and Göran Kauermann. Sources of uncertainty in machine learning–a statisticians' view. *arXiv preprint arXiv:2305.16703*, 2023.

[25] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.

[26] Fredrik K. Gustafsson, Martin Danelljan, and Thomas B. Schön. How reliable is your regression model's uncertainty under real-world distribution shifts?, 2023.

[27] Xizewen Han, Huangjie Zheng, and Mingyuan Zhou. Card: Classification and regression diffusion models. *Advances in Neural Information Processing Systems*, 35:18100–18115, 2022.

[28] James Harrison, John Willes, and Jasper Snoek. Variational bayesian last layers. In *International Conference on Learning Representations (ICLR)*, 2024.

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

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

[31] Pavel Izmailov, Sharad Vikram, Matthew D Hoffman, and Andrew Gordon Gordon Wilson. What are bayesian neural network posteriors really like? In *International conference on machine learning*, pages 4629–4640. PMLR, 2021.

[32] Asanobu Kitamoto, Jared Hwang, Bastien Vuillod, Lucas Gautier, Yingtao Tian, and Tarin Clanuwat. Digital typhoon: Long-term satellite image dataset for the spatio-temporal modeling of tropical cyclones. *Advances in Neural Information Processing Systems*, 36, 2024.

[33] Roger Koenker and Gilbert Bassett Jr. Regression quantiles. *Econometrica: Journal of the Econometric Society*, pages 33–50, 1978.

[34] Katrina Krämer. Daily briefing: Why forecasters failed to predict hurricane otis. *Nature*, 2023.

[35] Ranganath Krishnan, Pi Esposito, and Mahesh Subedar. Bayesian-torch: Bayesian neural network layers for uncertainty estimation, January 2022.

[36] Adrian Lafage and Olivier Laurent. Torch Uncertainty. `https://github.com/ENSTA-U2IS-AI/torch-uncertainty`, 2024.

[37] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

[38] Sungyoon Lee, Hoki Kim, and Jaewook Lee. GradDiv: Adversarial robustness of randomized neural networks via gradient diversity regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[39] Nils Lehmann, Nina Maria Gottschling, Stefan Depeweg, and Eric Nalisnick. Uncertainty aware tropical cyclone wind speed estimation from satellite data. *arXiv preprint arXiv:2404.08325*, 2024.

[40] Jeremiah Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax Weiss, and Balaji Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in Neural Information Processing Systems*, 33:7498–7512, 2020.

[41] Alexander Lyzhov, Yuliya Molchanova, Arsenii Ashukha, Dmitry Molchanov, and Dmitry Vetrov. Greedy policy search: A simple baseline for learnable test-time augmentation. In *Conference on Uncertainty in Artificial Intelligence*, pages 1308–1317. PMLR, 2020.

[42] Zhaoyang Ma, Yunfeng Yan, Jianmin Lin, and Dongfang Ma. A multi-scale and multi-layer feature extraction network with dual attention for tropical cyclone intensity estimation. *IEEE Transactions on Geoscience and Remote Sensing*, 2024.

[43] Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. *Advances in neural information processing systems*, 32, 2019.

[44] M. Maskey, R. Ramachandran, I. Gurung, B. Freitag, M. Ramasubramanian, and J. Miller. Tropical Cyclone Wind Estimation Competition Dataset. `https://doi.org/10.34911/rdnt.xs53up`, 2021.

[45] Riccardo Miotto, Fei Wang, Shuang Wang, Xiaoqian Jiang, and Joel T Dudley. Deep learning for healthcare: review, opportunities and challenges. *Briefings in bioinformatics*, 19(6):1236–1246, 2018.

[46] Zachary Nado, Neil Band, Mark Collier, Josip Djolonga, Michael W Dusenberry, Sebastian Farquhar, Qixuan Feng, Angelos Filos, Marton Havasi, Rodolphe Jenatton, et al. Uncertainty baselines: Benchmarks for uncertainty & robustness in deep learning. *arXiv preprint arXiv:2106.04015*, 2021.

[47] Yuhao Nie, Xiatong Li, Andea Scott, Yuchi Sun, Vignesh Venugopal, and Adam Brandt. Skipp'd: A sky images and photovoltaic power generation dataset for short-term solar forecasting. *Solar Energy*, 255:171–179, 2023.

[48] Yuhao Nie, Yuchi Sun, Yuanlei Chen, Rachel Orsini, and Adam Brandt. Pv power output prediction from sky images using convolutional neural network: The comparison of sky-condition-specific sub-models and an end-to-end model. *Journal of Renewable and Sustainable Energy*, 12(4), 2020.

[49] David A Nix and Andreas S Weigend. Estimating the mean and variance of the target probability distribution. In *Proceedings of 1994 ieee international conference on neural networks (ICNN'94)*, volume 1, pages 55–60. IEEE, 1994.

[50] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32, 2019.

[51] Quentin Paletta, Guillaume Arbod, and Joan Lasenby. Benchmarking of deep learning irradiance forecasting models from sky images–an in-depth analysis. *Solar Energy*, 224:855–867, 2021.

[52] Quentin Paletta, Anthony Hu, Guillaume Arbod, and Joan Lasenby. Eclipse: Envisioning cloud induced perturbations in solar energy. *Applied Energy*, 326:119924, 2022.

[53] Theodore Papamarkou, Maria Skoularidou, Konstantina Palla, Laurence Aitchison, Julyan Arbel, David Dunson, Maurizio Filippone, Vincent Fortuin, Philipp Hennig, Aliaksandr Hubin, et al. Position paper: Bayesian deep learning in the age of large-scale AI. *arXiv preprint arXiv:2402.00809*, 2024.

[54] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[55] Claudio Persello, Jan Dirk Wegner, Ronny Hänsch, Devis Tuia, Pedram Ghamisi, Mila Koeva, and Gustau Camps-Valls. Deep learning and earth observation to support the sustainable development goals: Current approaches, open challenges, and future opportunities. *IEEE Geoscience and Remote Sensing Magazine*, 10(2):172–200, 2022.

[56] Harry A Pierson and Michael S Gashler. Deep learning in robotics: a review of recent research. *Advanced Robotics*, 31(16):821–835, 2017.

[57] Ritesh Pradhan, Ramazan S Aygun, Manil Maskey, Rahul Ramachandran, and Daniel J Cecil. Tropical cyclone intensity estimation using a deep convolutional neural network. *IEEE Transactions on Image Processing*, 27(2):692–702, 2017.

[58] David Radke, Anna Hessler, and Dan Ellsworth. Firecast: Leveraging deep learning to predict wildfire spread. In *IJCAI*, pages 4575–4581, 2019.

[59] Stephan Rasp, Peter D Dueben, Sebastian Scher, Jonathan A Weyn, Soukayna Mouatadid, and Nils Thuerey. Weatherbench: a benchmark data set for data-driven weather forecasting. *Journal of Advances in Modeling Earth Systems*, 12(11):e2020MS002203, 2020.

[60] Markus Reichstein, Gustau Camps-Valls, Bjorn Stevens, Martin Jung, Joachim Denzler, Nuno Carvalhais, and fnm Prabhat. Deep learning and process understanding for data-driven earth system science. *Nature*, 566(7743):195–204, 2019.

[61] Hippolyt Ritter, Aleksandar Botev, and David Barber. A scalable laplace approximation for neural networks. In *6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings*, volume 6. International Conference on Representation Learning, 2018.

[62] Yaniv Romano, Evan Patterson, and Emmanuel Candes. Conformalized quantile regression. *Advances in neural information processing systems*, 32, 2019.

[63] Franko Schmähling, Jörg Martin, and Clemens Elster. A framework for benchmarking uncertainty in deep regression. *Applied Intelligence*, pages 1–14, 2022.

[64] Florian Seligmann, Philipp Becker, Michael Volpp, and Gerhard Neumann. Beyond deep ensembles: A large-scale evaluation of bayesian deep learning under distribution shift. *Advances in Neural Information Processing Systems*, 36, 2024.

[65] Mrinank Sharma, Sebastian Farquhar, Eric Nalisnick, and Tom Rainforth. Do bayesian neural networks need to be fully stochastic? In *International Conference on Artificial Intelligence and Statistics*, pages 7694–7722. PMLR, 2023.

[66] Robert H Simpson. The hurricane disaster—potential scale. *Weatherwise*, 27(4):169–186, 1974.

[67] Yuchi Sun, Gergely Szűcs, and Adam R Brandt. Solar pv output prediction from video streams using convolutional neural networks. *Energy & Environmental Science*, 11(7):1811–1818, 2018.

[68] Yuchi Sun, Vignesh Venugopal, and Adam R Brandt. Short-term solar power forecast with deep learning: Exploring optimal input and output configuration. *Solar Energy*, 188:730–741, 2019.

[69] Joost van Amersfoort, Lewis Smith, Andrew Jesson, Oscar Key, and Yarin Gal. On feature collapse and deep kernel learning for single forward pass uncertainty. *arXiv preprint arXiv:2102.11409*, 2021.

[70] Joost Van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. In *International Conference on Machine Learning*, pages 9690–9700. PMLR, 2020.

[71] Hongxin Wei and Jianguo Huang. TorchCP: A library for conformal prediction based on PyTorch, 2024.

[72] Ross Wightman. PyTorch Image Models. `https://github.com/rwightman/pytorch-image-models`, 2019.

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

[74] Anthony Wimmers, Christopher Velden, and Joshua H Cossuth. Using deep learning to estimate tropical cyclone intensity from satellite passive microwave imagery. *Monthly Weather Review*, 147(6):2261–2282, 2019.

[75] Chang-Jiang Zhang, Xiao-Jie Wang, Lei-Ming Ma, and Xiao-Qin Lu. Tropical cyclone intensity classification and estimation using infrared satellite images with deep learning. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:2070–2086, 2021.

[76] Jinsong Zhang, Rodrigo Verschae, Shohei Nobuhara, and Jean-François Lalonde. Deep photovoltaic nowcasting. *Solar Energy*, 176:267–276, 2018.

# NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We introduce the toolbox in Section 2 and analyze its usefulness on three example datasets in Section 4.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations, which is the embedding into the `PyTorch` framework, within a paragraph of Section 2.

   Guidelines:

   - The answer NA means that the paper has no limitations, while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper provides a toolbox and does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All experiments are reproducible with the presented toolbox and the provided code. Further, we describe the experimental setups in Section 3 and in the supplement and reference related works.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The whole toolbox is under Apache-2.0 license. The full code for the presented example experiments, utilizing the toolbox, is provided together with instructions and explanations: https://github.com/lightning-uq-box/experiments.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The paper mentions experimental setups that is needed to understand the presented results and further references to works on which the experimental setup builds upon.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [NA]

Justification: The experiments are utilized to represent the usability and potential advantages of the toolbox.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We stated the resources (Nvidia A100 GPU 40GB) and the computation time for all experiments of less than 10 hours when automated run with the UQ toolbox.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We do not see any potential harm caused by the research process and no negative societal and potentially harmful consequences.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We do not see negative societal impacts and point out the positive impact of open-source uncertainty quantification frameworks in supporting the work on topics with positive social impact.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We do not see the potential for direct misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example, by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We reference the used data sets and experimental setups from other works. We further reference the frameworks on which our toolbox builds up, `Lightning` and `PyTorch`.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The asset is given by the `Lightning UQ Box` that is linked and fully open-source. For the experiments there is further code for reproduction of the experiments provided.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The work does not contain crowdsourcing experiments and research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: There were no study participants for this work.

    Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymymity (if applicable), such as the institution conducting the review.