Quantum-Aided Bayesian Learning for the Prediction and Uncertainty Quantification of Remaining Useful Life

Giorgio Tosti Balducci[1], Nick Eleftheroglou[2]

[1,2] *Intelligent System Prognostics Group, Aerospace Structures and Materials Department, Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, Delft, 2629 HS, The Netherlands*

g.b.l.tostibalducci@tudelft.nl

n.eleftheroglou@tudelft.nl

ABSTRACT

To make predictions on the future states of engineering systems, prognostics has been increasingly relying on data-driven models and machine learning. Recent work has also looked at the potential of quantum machine learning to provide insight on the health state of systems. Nevertheless, these approaches treated the quantum component only as a deterministic predictor, in fact disregarding the uncertainty information. In this work, we propose a different approach to exploiting quantum circuits in prognostics. We focus on Bayesian learning of neural networks for Remaining Useful Life (RUL) prediction and uncertainty quantification. Here, the quantum circuit is introduced not as a data-driven model, but as a generator of neural network weights. Using Variational Inference, the quantum circuit can be trained to approximate the true posterior of the classical machine learning predictor. The overall method retains the data-processing ability of state-of-the-art machine learning models, while exploiting the quantum circuit to introduce uncertainty by sampling the model space from a distribution that is classically nontrivial. We validate our approach on the task of predicting the End of Discharge of Li-ion batteries, using data generated from a simulator with tunable process uncertainty, and we compare the predictions obtained through quantum sampling with those from Flipout Bayesian neural networks, heteroscedastic neural networks and Monte Carlo Dropout. The results show that our quantum circuits learn to approximate the weight posterior and that the resulting data-driven models demonstrate accuracy and uncertainty quantification that is comparable if not superior to the baselines. Overall, our work demonstrates the potential of quantum computing for uncertainty-aware prognostics, and sets the stage for further investigations in this area.

1. INTRODUCTION

Prognostics is concerned with predicting the future health state of engineering systems and, in particular, their remaining useful life (RUL). Accurate RUL estimates are a prerequisite for condition-based maintenance decisions, enabling operators to intervene before failure while avoiding unnecessary downtime (Sikorska, Hodkiewicz, & Ma, 2011; Lei et al., 2018). Because prognostic predictions deals with future events, uncertainty quantification is as important as the predictions themselves (Sankararaman & Goebel, 2015).

In order to achieve accurate and reliable RUL predictions, the field has adopted a broad range of data-driven approaches, also thanks to the availability of run-to-failure sensor data (Ferreira & Gonçalves, 2022). These include the early recurrent neural networks (Heimes, 2008), modern deep convolutional architectures (Li, Ding, & Sun, 2018) and lately attention mechanisms (Zhang et al., 2022) and physics-informed machine learning (Shi, Rivera, & Wu, 2022).

Despite the importance of quantifying uncertainty in prognostics, most machine learning techniques showcase improvements in accuracy, but they do not provide uncertainty estimates (Salinas-Camus, Goebel, & Eleftheroglou, 2025a). Nevertheless, the framework of Bayesian Neural Networks (MacKay, 1992) and heuristics such as Monte Carlo Dropout (Gal & Ghahramani, 2016) and deep ensembles (Lakshminarayanan, Pritzel, & Blundell, 2017) allow machine learning models to predict distributions rather than single point estimates and have seen a growing interest in PHM (Nemani et al., 2023). Accurate uncertainty quantification depends then not only on the expressive power of the data-driven model, but also on the choice of the distribution used to approximate the true posterior of the model's parameters. Whereas choices such as mean-field Gaussian distributions (Blundell, Cornebise, Kavukcuoglu, & Wierstra, 2015) are simple and computationally efficient, they may not be flexible enough to capture the true posterior, which can be highly complex and multi-modal.

In this work, we propose to use quantum circuits as a way to represent complex distributions over the weights of a classical data-driven model. We do this in the context of Variational Inference (Jordan, Ghahramani, Jaakkola, & Saul, 1999; Blei, Kucukelbir, & McAuliffe, 2017), which consists in approximating the weight posterior with a tractable parametric distribution and to learn its parameters within a training loop. In our approach, a parametrized quantum circuit (PQC) defines the shape of the variational distribution and its free parameters are trained to minimize the Kullback-Leibler (KL) divergence with respect to the true posterior (Benedetti, Coyle, Fiorentini, Lubasch, & Rosenkranz, 2021). A similar technique was proposed for the task of medical image diagnosis (Sakhnenko, Sikora, & Lorenz, 2024).

We remark that other studies have investigated the application of quantum computing to PHM, focusing specifically on quantum machine learning (Biamonte et al., 2017; Schuld & Petruccione, 2021) in PHM. In particular (Maior, Araújo, Lins, Moura, & Droguett, 2023; Martín Silva & López Droguett, 2024) demonstrated a hybrid workflow where a quantum-classical neural network is used to classify the health state of bearings. However, the rationale of such an approach can be questioned from two main standpoints. First, because the quantum circuit is used as a deterministic predictor, no uncertainty information is produced. Therefore it is impossible to assess the reliability of the model. Second, routing condition-monitoring data (CMD) through near-term quantum hardware is challenging due to the constraints of current quantum devices in terms of qubit count and coherence time. On the other hand, the design we propose consists in delegating to the quantum circuit only the task of representing the weight distribution, which can be achieved with limited quantum resources and without the need of processing large amounts of data on the quantum hardware. In this way, we preserve the data-processing ability of classical machine learning models, while exploiting the quantum circuit to introduce uncertainty by sampling the model space from a distribution that is classically nontrivial.

We validate the proposed approach on the task of predicting the time to End of Discharge (EoD) of lithium-ion batteries, using data generated from a circuit-equivalent simulator with tunable process uncertainty. The accuracy of the RUL estimates and the quality of the uncertainty quantification are compared against established techniques, namely Heteroscedastic Neural Networks (Zhao, Wu, Wong, Sun, & Yan, 2020), Bayesian Neural Networks with Flipout (Blundell et al., 2015; Wen, Vicol, Ba, Tran, & Grosse, 2018) and Monte Carlo Dropout (Gal & Ghahramani, 2016).

2. BACKGROUND

In the following the essential concepts of Bayesian neural networks, variational inference and quantum circuits, which constitute the main components of the proposed method, are briefly reviewed.

2.1. Bayesian Neural Networks

Rather than learning a single point estimate of the network weights, a BNN places a prior distribution $p(w)$ over the weight space $\mathcal{W}$ and updates it according to the knowledge gained from the observed data $\mathcal{D}$. The predictive distribution for a new input x^* is obtained by marginalising over the posterior distribution of the weights:

$$p(y^*|x^*, \mathcal{D}) = \int_{\mathcal{W}} p(y^*|x^*, w)p(w|\mathcal{D})dw. \tag{1}$$

The weight posterior $p(w|\mathcal{D})$ is given by Bayes' theorem:

$$p(w|\mathcal{D}) = \frac{p(\mathcal{D}|w)p(w)}{p(\mathcal{D})}, \tag{2}$$

where $p(w|\mathcal{D})$ is the likelihood of the weights given the data, and $p(\mathcal{D})$ is the marginal likelihood, computed by integrating over the entire weight space:

$$p(\mathcal{D}) = \int_{\mathcal{W}} p(\mathcal{D}|w)p(w)dw. \tag{3}$$

Direct sampling of the posterior distribution is unfeasible in practice, due to the impossibility of computing the integral in Eq. (3). Markov Chain Monte Carlo (MCMC) methods (Neal, 1996) which cover the posterior by sampling high probability regions, are also not a viable alternative when the weight space is very high-dimensional, as it is the case for neural networks. Variational inference and Monte Carlo Dropout, reviewed in the following subsections, offer scalable alternatives by replacing the exact posterior with a tractable approximation.

2.2. Variational Inference

Variational inference (VI) introduces a parametric family of distributions $q(w|\theta)$ and seeks the member closest to the true posterior (Jordan et al., 1999; Blei et al., 2017):

$$q(w|\theta) \approx p(w|\mathcal{D}). \tag{4}$$

The optimal variational parameters θ^* are found by minimising the Kullback–Leibler (KL) divergence between the approximate and true posteriors:

$$\theta^* = \arg\min_{\theta} \mathrm{KL}[q(w|\theta)||p(w|\mathcal{D})]. \tag{5}$$

A common simplifying assumption is the *mean-field* factorisation, in which each weight is treated as independent with its

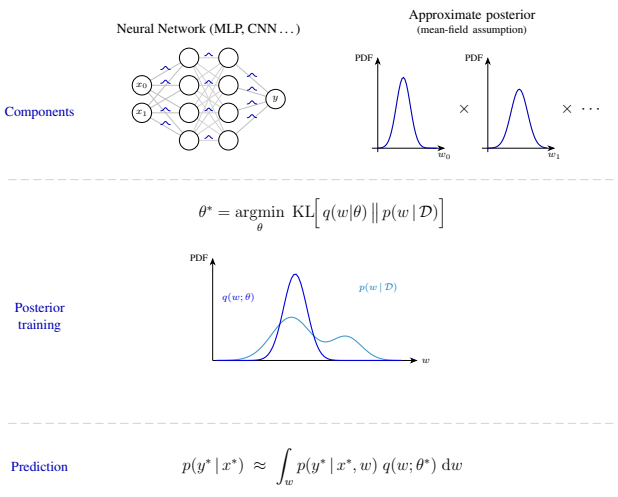

Figure 1. Schematic representation of mean-field variational inference (MFVI) for Bayesian neural networks.

own variational distribution:

$$q(w|\theta) = \prod_{i=1}^{N} q(w_i|\theta_i). \tag{6}$$

When each factor is chosen to be Gaussian, the resulting scheme is known as mean-field variational inference (MFVI) (Blundell et al., 2015):

$$q(w_i|\theta_i) = \mathcal{N}(\mu_i, \sigma_i^2). \tag{7}$$

Figure 1 illustrates the concept of Bayesian neural networks with mean-field variational inference.

Because the KL divergence in Eq. (5) requires the knowledge of the true posterior, it cannot be computed directly. Instead, minimising the KL divergence is shown to be equivalent to maximising the Evidence Lower Bound (ELBO), which decomposes into a data-fit term and a regularisation term penalising the deviation from the prior:

$$\mathcal{L}(\theta) = \mathbb{E}_{q(w|\theta)}[\log p(\mathcal{D}|w)] - \mathrm{KL}[q(w|\theta)||p(w)]. \tag{8}$$

A different and often more popular choice in PHM is Monte Carlo Dropout (MCD), which can be interpreted as a particular case of variational inference in which the approximate posterior consists a of Bernoulli mask over the network weights (Gal & Ghahramani, 2016). In practice, this technique averages over an ensemble of sub-networks with some of the weights switched off, and therefore it embeds in standard training procedures and it only requires to run the model multiple times at inference time.

2.3. Quantum Computation

Quantum computation describes how to operate on quantum bits (qubits) to implement algorithms. Thanks to the principles of superposition and entanglement, quantum computers can represent and process information in ways that are fundamentally different from classical computers (Nielsen & Chuang, 2012).

A state of Q qubits is represented as a vector in a 2^Q-dimensional complex Hilbert space, where the basis states are the set of all possible binary strings of length Q. The generic quantum state $|\psi\rangle$ can be prepared by applying a unitary transformation U to the reference state $|0\rangle$, corresponding to the all-zero binary string:

$$|\psi\rangle = U\,|0\rangle\,. \tag{9}$$

The braket notation $|\cdot\rangle$ and $\langle\cdot|$ is used to denote quantum states, where $|\psi\rangle$ is a column vector and $\langle\psi|$ is its conjugate transpose (a row vector).

Because the information in the quantum state cannot be directly accessed, it is necessary to perform a measurement, which collapses the state to one of the basis states with a probability given by the squared amplitude of the corresponding state vector component:

$$p(b) = |\,\langle b|\psi\rangle\,|^2, \tag{10}$$

where $|b\rangle$ is a basis state. Because of the probabilistic nature of the wave function, repeated measurements can can yield statistics about the state, such as its expectation value with respect to an observable O:

$$\mathbb{E}(O) = \langle\psi|O|\psi\rangle\,. \tag{11}$$

Thanks to the above principles and to the possibility of parametrizing the unitary transformations, quantum circuits can be designed to represent continuous functions, as it is commonly done in quantum machine learning (Schuld & Petruccione, 2021). Therefore, a parametrized quantum circuit (PQC) can be used to model a function $\xi(x;\theta)$, where x is the input and θ are circuit parameters that are usually learned in an optimization loop. The output of the function is obtained by measuring an observable O on the state prepared by the circuit:

$$\xi(x;\theta) = \langle\psi(x;\theta)|O|\psi(x;\theta)\rangle\,, \tag{12}$$

where

$$|\psi(x;\theta)\rangle = U_1(\theta)U_0(x)\,|0\rangle\,. \tag{13}$$

Figure 2 shows a generic PQC for representing continuous functions, where the input x is encoded in the state through the unitary $U_0(x)$, and the tunable parameters are applied through the unitary $U_1(\theta)$. Furthermore, Fig. 3 gives an ex-

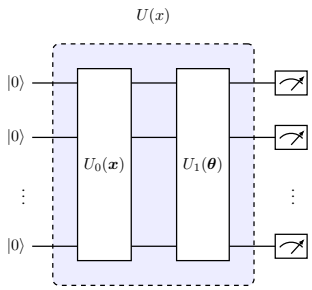

Figure 2. Generic parametrized quantum circuit for representing continuous functions.

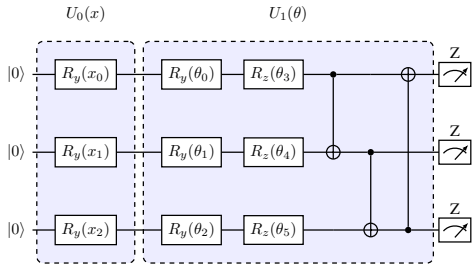

Figure 3. Quantum circuit with expanded unitary operations for representing continuous functions.

ample of the elementary gates that can compose the parametrized unitaries, such as rotations and two-qubit gates and of the choice of measurement operator. The interested reader is referred to standard quantum computing textbooks (Nielsen & Chuang, 2012) for more details on quantum operations and the quantum circuit formalism.

3. QUANTUM ADVERSARIAL VARIATIONAL INFERENCE

The main idea of the proposed method is to use a parametrized quantum circuit to approximate the weight posterior of a classical neural network. Despite the fact that quantum circuits are inherently stochastic, they are limited to sample from discrete distributions, whereas a generator for the weights of a regression model is required to produce continuous values. The approach proposed in (Romero & Aspuru-Guzik, 2021) is followed, in order to fit a quantum circuit within a generator of continuous weight values.

The stochasticity of the generator is delegated to a classical noise variable z, which is chosen to be uniformly sampled from the interval $[0, 2\pi]$:

$$z \sim \mathcal{U}(0, 2\pi). \tag{14}$$

A parametrized quantum circuit maps the noise variable to a quantum state $|\psi(z;\theta)\rangle$ as in Eq. (13), where θ are the tunable parameters of the circuit, and therefore also of the variational weight distribution. Depending on the number of weights required by the classical data model, a set of observables must be chosen as one of the hyperparameters of the method:

$$\{O_0, O_1, \ldots, O_{|w|-1}\}, \tag{15}$$

where $|w|$ is the number of weights in the data-driven model.

For observable O_i, the output of the generator corresponding to a sample z will be the continuous random variable obtained as the expectation value of O_i in the state $|\psi(z;\theta)\rangle$,

$$\xi_i = \langle\psi(z;\theta)|O_i|\psi(z;\theta)\rangle\,, \tag{16}$$

whereas the vector of all outputs correponding to the weights will be

$$\xi = [\xi_0, \xi_1, \ldots, \xi_{|w|-1}]^\top. \tag{17}$$

Because the output of the quantum circuit is limited to the range of the expectation value of the chosen observables, a classical post-processing step maps the observed vector ξ to the weight space $\mathcal{W}$ through a trainable linear transformation

$$w = W_{pp}\,\xi + b_{pp}. \tag{18}$$

The overall quantum-enhanced generator is schematically illustrated in Fig. 4. The weights θ need to be learned to best represent the true weight posterior. Unlike the case where the generator distribution has a closed form expression, like in the mean-field Gaussian case, the quantum generator does not have an explicit density function that can be evaluated. Therefore, the KL divergence in Eq. (5) cannot be computed directly and the ELBO in Eq. (8) is not a viable objective function for training. A way to learn distributions with only the possibility of sampling the generator is to use adversarial learning. This idea was proposed for general quantum variational inference in (Benedetti et al., 2021) and it is formalized here for the task of Bayesian regression with a data-driven model.

In essence, a generative learning task can be rephrased as an adversarial game between a generator and a discriminator. The generator needs to produce samples that are indistinguishable from those of the true data distribution, while the discriminator is trained to classify real and generated samples. The discriminator can be any model able to perform binary classification, for example a neural network with a final sigmoid layer.

In the following, the probability distribution of the generator is referred to as $p_g(w|\Theta_g)$, where $\Theta_g = \{\theta, W_{pp}, b_{pp}\}$ are the parameters of the generator, including both the quantum circuit parameters and the post-processing parameters. The discriminator is a function $d(x, \hat{y}|\Theta_d)$, where Θ_d are the discriminator parameters, x is the input to the data-driven model and $\hat{y}$ is either the true label or the predicted label obtained

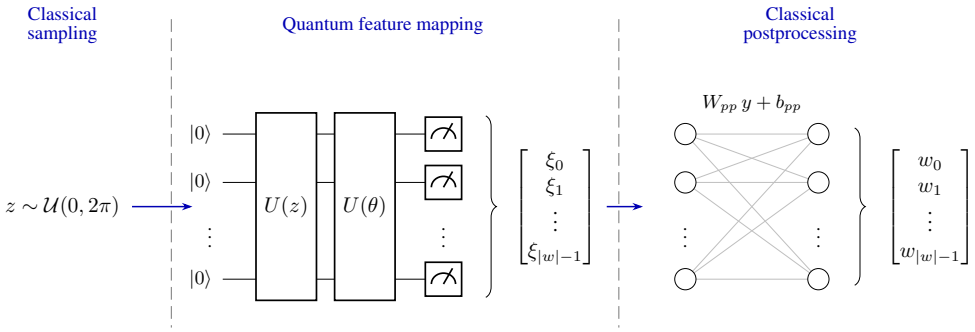

Figure 4. Overall structure of the weight generator with quantum feature transformation.

by feeding x through the data-driven model f with weights sampled from the generator distribution, thus

$$\hat{y} = \begin{cases} y, & \text{(true RUL)} \\ f(x|w), & \text{(predicted RUL).} \end{cases} \tag{19}$$

The discriminator works to assign high probability to the true RUL labels and low probability to the generated ones. Therefore, it tries to maximize the following objective function:

$$\begin{aligned} \mathcal{L}_d(\Theta_d) = \mathbb{E}_{p(x,y)}[\log d(x, y|\Theta_d)] \\ + \mathbb{E}_{p(x)p_g(w|\Theta_g)}[\log(1 - d(x, f(x|w)|\Theta_d))], \end{aligned} \tag{20}$$

where $p(x, y)$ is the true data distribution and $p(x)$ is the marginal distribution of the inputs. The generator, on the other hand, pursues two objectives, expressed conventionally as a minimization problem. First, it tries to minimize the success rate of the discriminator and then it works to reduce the error between the generated samples and the true data, which corresponds to the likelihood term in Eq. (8). The generator loss can therefore be expressed as

$$\begin{aligned} \mathcal{L}_g(\Theta_g) = \mathbb{E}_{p(x)p_g(w|\Theta_g)}[\operatorname{logit} d(x, f(x|w)|\Theta_d)] \\ - \mathbb{E}_{p(x,y)p_g(w|\Theta_g)}[\log p(y|x, w)], \end{aligned} \tag{21}$$

where $p(y|x, w)$ is the likelihood of the data given the weights, which is usually expressed as a Gaussian distribution in regression tasks. The logit terms expands as

$$\operatorname{logit} d(x, f(x|w)|\Theta_d) = \log \frac{d(x, f(x|w)|\Theta_d)}{1 - d(x, f(x|w)|\Theta_d)}, \tag{22}$$

and it is used to avoid vanishing gradients of the generator early in training (Goodfellow et al., 2014).

In conclusion, the quantum generator learns to approximate the true weight posterior by playing the following minimax game with the discriminator:

$$\begin{aligned} &\min_{\Theta_g} \mathcal{L}_g(\Theta_g) \\ &\max_{\Theta_d} \mathcal{L}_d(\Theta_d), \end{aligned} \tag{23}$$

a process which we refer to as Quantum Adversarial Variational Inference (QAVI).

4. CASE STUDY

To study the ability of the proposed method to predict and quantify the uncertainty of the remaining useful life of a system, the task of predicting the End of Discharge (EoD) of lithium-ion batteries is considered. For this particular problem, the RUL is intended as the time to EoD, that is the time after which the battery voltage drops below a known threshold V_{EoD}. The following sections elucidate the details of the battery model, data generation and machine learning techniques used as benchmarks and finally present and discuss the results of the case study.

4.1. Battery Model

In order to produce the discharge voltage hisories, the Thévenin-equivalent circuit model in Fig. 5 is employed. The model consists of an open circuit voltage (OCV) source, function of the state of charge (SoC) of the battery and a resistor R. The readout voltage V of the circuit is used as the input condition-monitoring data for the RUL prediction task.

The SoC is updated at each time step according to the constant loading current I, negative during discharge, as

$$\mathrm{SoC}_{t+1} = \mathrm{SoC}_t + \frac{I\Delta t}{Q} + \varepsilon_{SoC}, \tag{24}$$

where Δt is the time step, Q is the constant battery capacity and the process uncertainty is modeled as a Gaussian noise term ε_{SoC}. The open-circuit voltage V_{OC} follows the follow-

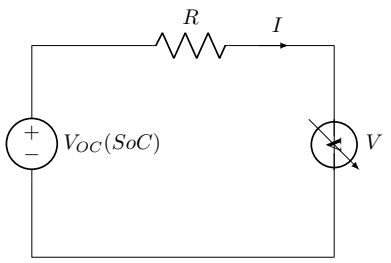

Figure 5. Equivalent circuit model used to generate the discharge voltages.

ing empirical law (Bustos et al., 2025):

$$\begin{aligned} V_{OC}(\mathrm{SoC}_t) = v_L \\ + (v_0 - v_L)\exp\left[\gamma(\mathrm{SoC}_t - 1)\right] \\ + \alpha v_L(\mathrm{SoC}_t - 1) \\ + (1-\alpha)v_L\left(\exp(-\beta) - \exp\left(-\beta\sqrt{\mathrm{SoC}_t}\right)\right), \end{aligned} \tag{25}$$

Finally, the readout voltage V is given by

$$V_t = V_{OC}(\mathrm{SoC}_t) + RI. \tag{26}$$

The EoD is defined as the time step at which the voltage V drops below a known threshold V_{EoD}. Both the battery model and the discharge simulation parameters are available in the Appendix at the end of the paper.

4.2. Design of Experiments

The battery model described in Section 4.1 is used to generate training and test data histories. Operatively speaking, the simulator is initialized with a SoC value and then runs the SoC and V_t updates until reaching t_{EoD}, which is the time when $V_t < V_{EoD}$. The generated voltage history is then used as the input for the data-driven model, while the RUL labels are obtained following a linear degradation model,

$$\mathrm{RUL}_t = t_{EoD} - t. \tag{27}$$

For the training set, 100 discharge histories are generated. Each simulation is initialized with a SoC value sampled uniformly from the interval $[0.05,\ 1.0]$, which ensures that the dispersion of remaining useful life narrows down towards the end of the discharge. For testing the models, ten different battery discharges are simulated from a full charge state. In order to obtain the target RUL distributions for later assessing the epistemic uncertainty of the models, 100 simulations are run from 200 decreasing SoC values, picked along the SoC profile of each test case. Training and testing sets are generated with the same simulator and battery parameters, same amplitude of the process noise, but different random seeds.

4.3. Data-Driven Model and Uncertainty Quantification

The case study compares together four different uncertainty modeling techniques for the same data-driven model. The latter consists of a convolutional neural network (CNN) with one convolutional layer of four filters and kernel size of 5, a global average pooling layer and a final fully-connected layer with two outputs. The CNN takes as input a time-window of voltage data of duration of roughly 10% of the total simulation time and it predicts the RUL of the battery at the last time step of the window. As output, the CNN returns the predicted mean RUL and the predicted variance, which represents the heteroscedastic aleatoric uncertainty coming from the process noise of the battery data.

In terms of modeling uncertainty, the following techniques are compared:

- *Heteroscedastic neural network* (HNN): the data-driven model is trained with a negative log-likelihood loss, which encourages the model to learn the variance of the data distribution as well as the mean. This method accounts for aleatoric but not for epistemic uncertainty.
- *Monte Carlo Dropout* (MCD): the data-driven model is trained with dropout and the epistemic uncertainty is obtained by performing multiple forward passes with dropout at inference time.
- *Bayesian neural network with Flipout* (BNN): the mean-field variational Gaussian distribution learns the weight posterior by minimizing the ELBO loss in Eq. (8) and the Flipout weight update (Wen et al., 2018) rule stabilizes the training by decorrelating the weight perturbations.
- *Quantum Adversarial Variational Inference* (QAVI): the proposed method, detailed in Section 3. For sampling of the weights, one generator of the type illustrated in Fig. 4 is used for each filter of the convolutional layer.

Both for the maximum likelihood estimators (HNN and MCD) and for the variational inference methods (BNN and QAVI), the likelihood term is given by the negative log-likelihood of a Gaussian distribution, expressed as

$$\mathcal{L}_{NLL} = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{(y_i - \hat{y}_i)^2}{\sigma_i^2} + \log\sigma_i^2\right], \tag{28}$$

where y_i is the RUL label, $\hat{y}_i$ is the predicted mean RUL and σ_i^2 is the predicted variance.

4.4. Results

All models were trained for a maximum of 500 epochs, using the Adam optimizer with a cosine annealing schedule. Early stopping with a patience of 50 epochs monitored the training in case of increasing loss over a validation set.

Figure 6 shows the RUL predictions and the Continuous Ranked Probability Score (CRPS) (Hersbach, 2000) over time for

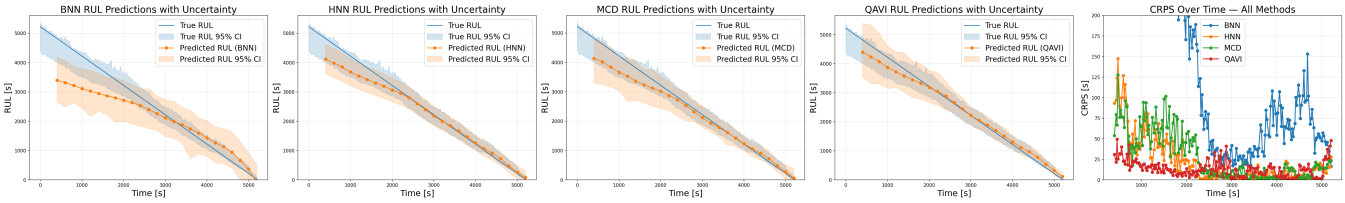

Figure 6. RUL predictions and CRPS over time for the test discharge history. The true RUL corresponds to the single simulated discharge path, while the predicted RUL is the mean-predicting output of the CNN. 95% confidence intervals are shown for both ground truth and predictions.

the first test discharge histories among those generated. Three more test cases are available in the Appendix of this paper and confirm the findings reported here.

Some general observations help with the interpretation of the results. First, it is important to notice that the CRPS values are obtained as the integrated square difference of the cumulative distribution functions (CDFs) of the predicted and true RUL distributions, therefore,

$$\mathrm{CRPS} = \int_{-\infty}^{\infty} \left[F_{\hat{y}}(r) - F_y(r)\right]^2 dr. \tag{29}$$

It was possible to compute this quantity since the true RUL distribution could be reconstructed by following the discharge path and running Monte Carlo simulations from intermediate states. Furthermore, it can be observed that all predictions start from a later time than the beginning of the discharge, due to the fact that the CNN is trained on time windows of data and it cannot output a prediction before the first time window is filled. Also worth mentioning is the fact that all models have a roughly symmetric confidence interval around the predicted mean, which is a consequence of the fact that the output of the CNN is a Gaussian distribution by construction. Finally, it should be kept in mind that, because HNN only accounts for aleatoric uncertainty, the width of its confidence interval is expected to be smaller than the other methods, and it acts as a lower bound for the overall uncertainty.

It can be noticed by the RUL plots that the QAVI method achieves good accuracy and that its confidence intervals are well-calibrated, in the sense that they contain the true RUL values for most of the discharge time. Its mean prediction is also slightly more accurate than HNN and MCD even if the true RUL is away from the median of the distribution. While HNN, MCD and QAVI follow the RUL trend and progressively shrink their confidence intervals, the BNN method seems to struggle in learning the weight posterior, as the RUL is significantly underestimated in the early discharge and the uncertainty remains almost constant over time. The CRPS trend over time confirms the previous analysis and it highlights how QAVI compromises best between accuracy and uncertainty especially away from the end of discharge.

An even more comprehensive comparison of the CRPS is

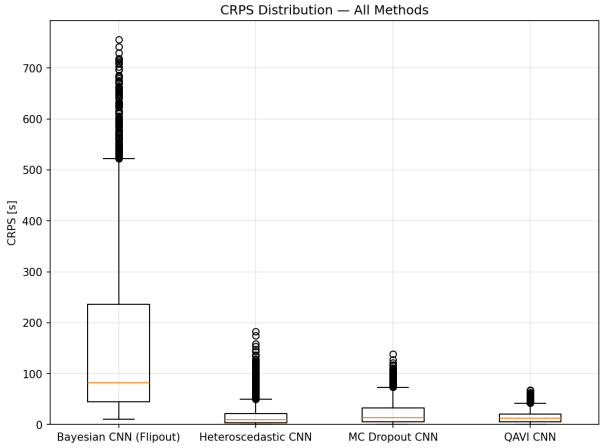

Figure 7. Box plot of the CRPS values for different methods.

shown in Fig. 7, where a box plot shows the CRPS statistics across 10 different test cases and over time. The results are consistent with the previous ones and show that QAVI scores comparably with HNN, hinting at the fact that this technique achieves the lowest epistemic uncertainty among the methods compared. Interestingly, the dispersion of outliers for QAVI is even smaller than that of HNN, which might be a sign of robustness of the method that should be investigated in future work.

5. CONCLUSIONS

This paper presented the QAVI method for uncertainty-aware RUL prediction, which performs variational inference with quantum-enhanced generators. The technique was validated on a RUL-estimation problem for simulated lithium-ion batteries, where the remaining useful life was meant as the time to end of discharge. In this context, QAVI was also benchmarked against state-of-the-art uncertainty quantification schemes for machine learning models, namely heteroscedastic neural networks, Monte Carlo Dropout and Mean-Field Variational Inference. The results show that QAVI can be a promising approach for uncertainty-aware prognostics, as it achieves good accuracy and well-calibrated uncertainty ranges, while it keeps the epistemic uncertainty comparable or lower than those produced by the more established techniques.

It is important to stress, however, that the results of this paper constitutes only a proof of concept for QAVI and that future work in multiple directions can help establish the value of this method for real-world prognostic applications. In this sense, QAVI should be further characterized according to a set of key criteria for data-driven models in prognostics, such as those identified in (Salinas-Camus, Goebel, & Eleftheroglou, 2025b).

The robustness of the method should be assessed from a few different angles. It would be instructive to see how QAVI scores on out-of-distribution data, such as unseen loading current profiles in the case of battery modeling. Furthermore, future investigations should explore the sensitivity of the method to the choice of hyperparameters, such as the number of qubits and the structure of the quantum circuit. Both in terms of robustness, but also flexibility of the methods, it should be seen how QAVI performs on standard PHM benchmarks, such as the C-MAPSS dataset for turbofan engines (Saxena, Goebel, Simon, & Eklund, 2008) and real-world datasets.

Especially important would be to assess the feasibility of deploying QAVI on online systems. We remark here that until the quantum circuit resources are small, there might not be a need of quantum hardware to run the generator, as the quantum circuit can be simulated classically. In terms of online service, this might be beneficial, as there would be no need for the asset to exchange information with a quantum server, which might cause an unacceptable latency for some real-time prognostic scenarios. Naturally, this aspect would need to be addressed when the quantum resources required do not allow for simulation anymore.

A final important question concerns the interpretability of the method. Because of the complexity of the quantum-enhanced generator, it might be difficult to understand how the generator is learning to represent the weight posterior. Future work should explore techniques for interpreting the QAVI predictions, for example following interpretable machine learning settings that incorporate Bayesian learning (Kraus & Feuerriegel, 2019).

ACKNOWLEDGEMENT

The research presented in this paper has been performed in the framework of the MODABAT project (Modular, scalable and technology-Open Design for future Aviation BATteries) and has received funding from the European Union Clean Aviation Program under grant agreement n^{o} 101251224.

REFERENCES

Benedetti, M., Coyle, B., Fiorentini, M., Lubasch, M., & Rosenkranz, M. (2021, October). Variational inference with a quantum computer. *Physical Review Applied*, *16*(4), 044057. doi: 10.1103/physrevapplied.16.044057

Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., & Lloyd, S. (2017). Quantum machine learning. *Nature*, *549*, 195–202. doi: 10.1038/nature23474

Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, *112*, 859–877. doi: 10.1080/01621459.2017.1285773

Blundell, C., Cornebise, J., Kavukcuoglu, K., & Wierstra, D. (2015). Weight uncertainty in neural networks. In *Proceedings of the 32nd international conference on machine learning (icml)* (pp. 1613–1622).

Bustos, J. E. G., Schiele, B. B., Baldo, L., Masserano, B., Jaramillo-Montoya, F., Troncoso-Kurtovic, D., ... Silva, J. F. (2025, December). In situ estimation of li-ion battery state of health using on-board electrical measurements for electromobility applications. *Batteries*, *11*(12), 451. doi: 10.3390/batteries11120451

Ferreira, C., & Gonçalves, G. (2022). Remaining useful life prediction and challenges: A literature review on the use of machine learning methods. *Journal of Manufacturing Systems*, *63*, 550–562. doi: 10.1016/j.jmsy.2022.05.010

Gal, Y., & Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd international conference on machine learning (icml)* (pp. 1050–1059).

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2014, June). *Generative adversarial networks.* arXiv. doi: 10.48550/arXiv.1406.2661

Heimes, F. O. (2008). Recurrent neural networks for remaining useful life estimation. In *2008 international conference on prognostics and health management* (pp. 1–6). IEEE. doi: 10.1109/phm.2008.4711422

Hersbach, H. (2000, October). Decomposition of the continuous ranked probability score for ensemble prediction systems. *Weather and Forecasting*, *15*(5), 559–570. doi: 10.1175/1520-0434(2000)015¡0559:dotcrp¿2.0.co;2

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., & Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine Learning*, *37*, 183–233. doi: 10.1023/A:1007665907178

Kraus, M., & Feuerriegel, S. (2019, October). Forecasting remaining useful life: Interpretable deep learning approach via variational bayesian inferences. *Decision Support Systems*, *125*, 113100. doi:

10.1016/j.dss.2019.113100

Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In I. Guyon et al. (Eds.), *Advances in neural information processing systems* (Vol. 30). Curran Associates, Inc.

Lei, Y., Li, N., Guo, L., Li, N., Yan, T., & Lin, J. (2018). Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mechanical Systems and Signal Processing*, *104*, 799–834. doi: 10.1016/j.ymssp.2017.11.016

Li, X., Ding, Q., & Sun, J.-Q. (2018). Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering & System Safety*, *172*, 1–11. doi: 10.1016/j.ress.2017.11.021

MacKay, D. J. C. (1992). A practical Bayesian framework for backpropagation networks. *Neural Computation*, *4*, 448–472. doi: 10.1162/neco.1992.4.3.448

Maior, C. B. S., Araújo, L. M. M., Lins, I. D., Moura, M. D. C., & Droguett, E. L. (2023). Prognostics and health management of rotating machinery via quantum machine learning. *IEEE Access*, *11*, 25132–25151. doi: 10.1109/access.2023.3255417

Martín Silva, G. S., & López Droguett, E. (2024, June). Quantum kernel functions for the prognosis and health management of ball-bearing elements. In *2024 ieee international conference on prognostics and health management (icphm)* (pp. 257–264). IEEE. doi: 10.1109/icphm61352.2024.10626686

Neal, R. M. (1996). *Bayesian learning for neural networks* (Vol. 118). Springer. doi: 10.1007/978-1-4612-0745-0

Nemani, V., Biggio, L., Huan, X., Hu, Z., Fink, O., Tran, A., ... Hu, C. (2023, December). Uncertainty quantification in machine learning for engineering design and health prognostics: A tutorial. *Mechanical Systems and Signal Processing*, *205*, 110796. doi: 10.1016/j.ymssp.2023.110796

Nielsen, M. A., & Chuang, I. L. (2012). *Quantum computation and quantum information: 10th anniversary edition*. Cambridge University Press. doi: 10.1017/cbo9780511976667

Romero, J., & Aspuru-Guzik, A. (2021). Variational quantum generators: Generative adversarial quantum machine learning for continuous distributions. *Advanced Quantum Technologies*, *4*(1), 2000003.

Sakhnenko, A., Sikora, J., & Lorenz, J. (2024, June). Building continuous quantum-classical bayesian neural networks for a classical clinical dataset. In *Proceedings of recent advances in quantum computing and technology* (pp. 62–72). ACM. doi: 10.1145/3665870.3665872

Salinas-Camus, M., Goebel, K., & Eleftheroglou, N. (2025a, August). A comprehensive review and evaluation framework for data-driven prognostics: Uncertainty, robustness, interpretability, and feasibility. *Mechanical Systems and Signal Processing*, *237*, 113015. doi: 10.1016/j.ymssp.2025.113015

Salinas-Camus, M., Goebel, K., & Eleftheroglou, N. (2025b, August). A comprehensive review and evaluation framework for data-driven prognostics: Uncertainty, robustness, interpretability, and feasibility. *Mechanical Systems and Signal Processing*, *237*, 113015. doi: 10.1016/j.ymssp.2025.113015

Sankararaman, S., & Goebel, K. (2015). Uncertainty quantification in remaining useful life prediction for aerospace components. *International Journal of Prognostics and Health Management*, *6*(4), 1–17. doi: 10.36001/ijphm.2015.v6i4.2285

Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008, October). Damage propagation modeling for aircraft engine run-to-failure simulation. In *2008 international conference on prognostics and health management* (pp. 1–9). IEEE. doi: 10.1109/phm.2008.4711414

Schuld, M., & Petruccione, F. (2021). *Machine learning with quantum computers*. Springer. doi: 10.1007/978-3-030-83098-4

Shi, J., Rivera, A., & Wu, D. (2022). Battery health management using physics-informed machine learning: Online degradation modeling and remaining useful life prediction. *Mechanical Systems and Signal Processing*, *179*, 109347. doi: 10.1016/j.ymssp.2022.109347

Sikorska, J. Z., Hodkiewicz, M., & Ma, L. (2011). Prognostic modelling options for remaining useful life estimation by industry. *Mechanical Systems and Signal Processing*, *25*, 1803–1836. doi: 10.1016/j.ymssp.2010.11.018

Wen, Y., Vicol, P., Ba, J., Tran, D., & Grosse, R. (2018, March). *Flipout: Efficient pseudo-independent weight perturbations on mini-batches.* arXiv. doi: 10.48550/arXiv.1803.04386

Zhang, J., Jiang, Y., Wu, S., Li, X., Luo, H., & Yin, S. (2022). Prediction of remaining useful life based on bidirectional gated recurrent unit with temporal self-attention mechanism. *Reliability Engineering & System Safety*, *221*, 108377. doi: 10.1016/j.ress.2021.108297

Zhao, Z., Wu, J., Wong, D., Sun, C., & Yan, R. (2020). Probabilistic remaining useful life prediction based on deep convolutional neural network. *SSRN Electronic Journal*. doi: 10.2139/ssrn.3717738

APPENDIX

Table 1 lists the parameters of the battery model and of the simulation used to generate the data for the case study and Fig. 8 shows the RUL predictions and CRPS over time for three additional random discharge histories.

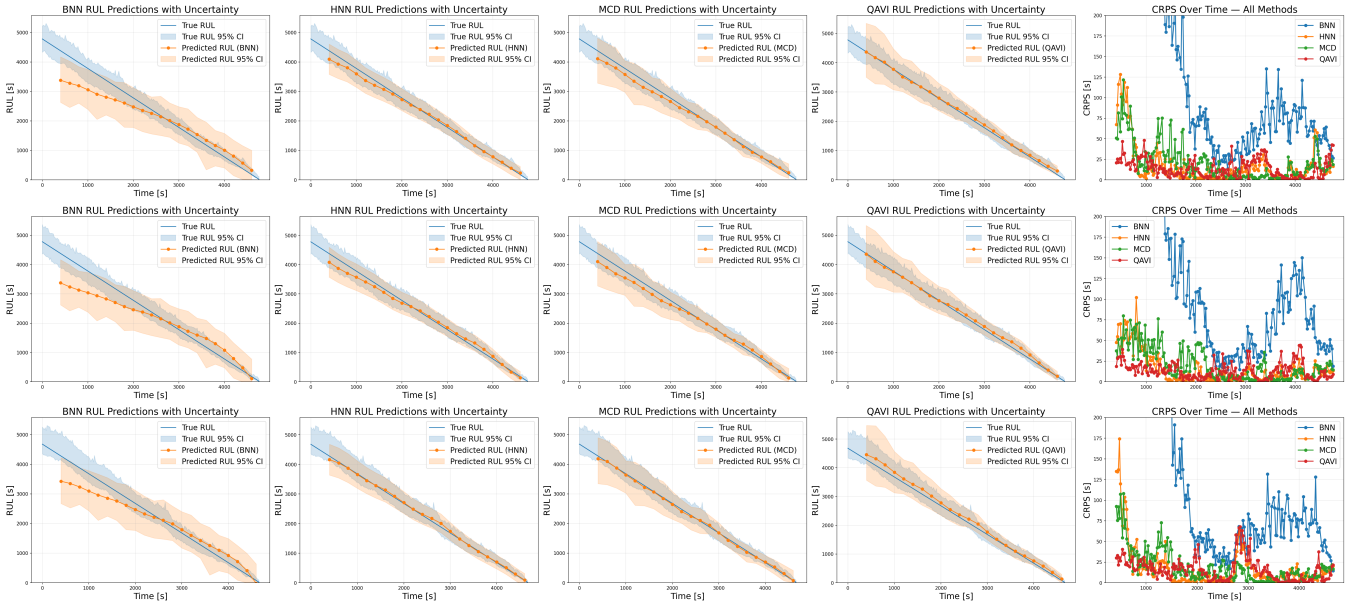

Figure 8. Additional test discharge histories comparing the four approaches of the case study.

Table 1. Battery and simulation parameters of the case study.

Parameter	Symbol	Value
Battery model		
Battery capacity	Q	2.8 A h
Internal resistance	R	0.1 Ω
OCV lower asymptote	V_L	1.3553 V
OCV reference voltage	V_0	4.1202 V
OCV shape parameter	γ	0.1329
OCV shape parameter	α	0.1695
OCV shape parameter	β	2.3454
Discharge simulation		
Discharge current amplitude	I	−2.1 A
Cut-off voltage	V_{EoD}	2.5 V
Simulation time step	Δt	20 s
Process noise std. dev.	σ_ω	3×10^{-3}

BIOGRAPHIES

Giorgio Tosti Balducci is a postdoc researcher in the iSP Group within the Faculty of Aerospace Engineering at TU Delft. He received his degrees in Aerospace Engineering from Politecnico di Milano, Italy (B.Sc.) and TU Delft (M. Sc.). He also earned his PhD from TU Delft in 2025 on the topic of quantum computing applications in structural mechanics. His work in the iSP Group focuses on the development of quantum-assisted methodologies for PHM with applications specifically in battery prognostics.

Nick Eleftheroglou is an Assistant Professor in the Faculty of Aerospace Engineering at Delft University of Technology an the head of the iSP Group. He received his Diploma in Mechanical and Aeronautics Engineering, cum laude, from the University of Patras, Greece, in 2015, and earned his PhD, cum laude, from TU Delft in October 2020. His research interests lie in the area of PHM, developing PHM frameworks with enhanced reliability, robustness, and feasibility for operations and maintenance.