Bachelor's Seminar

MC-Dropout and SWAG for Uncertainty Estimation in Neural Networks

Department of Statistics
Ludwig-Maximilians-Universität München

Martin Kandlinger

Munich, Juli 6th, 2025

Submitted in partial fulfillment of the requirements for the degree of B. Sc.
Supervised by Dr. Ludwig Bothmann

Abstract

Estimating uncertainty in deep neural networks is essential for reliable decision making, especially in high stakes applications. This work examines two practical approximation methods, Monte Carlo Dropout (MC-Dropout) and Stochastic Weight Averaging Gaussian (SWAG), for their effectiveness in uncertainty quantification. A unified evaluation framework assesses both methods on synthetic datasets, where uncertainty behavior is controllable and a real-world benchmark, the Boston Housing dataset. Metrics include calibration quality, predictive accuracy, and robustness under distribution shift. Results indicate that MC-Dropout provides a computationally efficient uncertainty estimation, whereas SWAG yields richer posterior approximations and improved calibration. Practical recommendations are provided to guide method selection according to specific application requirements.

Contents

1 Introduction

Deep neural networks (DNNs) have achieved remarkable success across domains such as computer vision, natural language processing, and healthcare. When deploying such models in critical applications regarding safety, such as autonomous driving, medical diagnosis, or financial risk assessment, quantifying model uncertainty becomes essential. Overconfident predictions on noisy or out-of-distribution inputs can lead to disastrous outcomes and undermine trust in machine learning systems (Pereira and Thomas, 2020, Valentin, 2024).

In predictive modeling, uncertainty arises from two main sources: aleatoric uncertainty, which is inherent noise in the data (e.g., sensor noise or label ambiguity) and epistemic uncertainty, which stems from limited data, incomplete knowledge, or insufficient model capacity. Unlike aleatoric uncertainty, epistemic uncertainty can be reduced by collecting more observations or employing a more appropriate model architecture that better captures the underlying function (Hüllermeier and Waegeman, 2021). This work focuses on the quantification of epistemic uncertainty, as it signals when models operate beyond their knowledge boundaries.

We evaluate two practical approximation methods, Monte Carlo Dropout (MC-Dropout) and Stochastic Weight Averaging Gaussian (SWAG). MC-Dropout offers a lightweight approach by retaining dropout during inference, effectively simulating Bayesian sampling through multiple stochastic forward passes without altering the models architecture (Gal and Ghahramani, 2016). SWAG constructs a Gaussian posterior over the model weights by capturing stochastic gradient descent (SGD) trajectory moments during training, offering richer uncertainty representations with improved robustness to distribution shifts (Maddox et al., 2019).

After presenting these methods, their performance will be compared within a unified experimental framework. Both MC-Dropout and SWAG are applied to synthetic datasets, where true uncertainty is known, and also to real-world data (the boston housing dataset with ethical preprocessing). The evaluation metrics include predictive accuracy (RMSE), calibration quality (NLL, PICP), and out-of-distribution reliability.

The work is structured as follows: Section 2 reviews prior work on uncertainty quantification methods and Bayesian neural models, Section 3 details the MC-Dropout and SWAG approaches, but also explains the underlying mechanisms and prior knowledge needed to understand them, Section 4 presents comparative evaluations, Section 5 analyzes trade-offs and future extensions and Section 6 summarizes key findings and their implications.

2 Related Work

A diverse set of techniques has been proposed for quantifying uncertainty in neural networks, which can broadly be categorized into variational approaches, ensemble methods and approximate inference techniques.

Variational Approaches Monte Carlo Dropout (MC-Dropout), introduced by Gal and Ghahramani (2016), implements variational inference by retaining dropout at test time to obtain uncertainty estimates. While computationally efficient, subsequent research has identified limitations. MC-Dropout can underestimate uncertainty, particularly in regions with little to no data (Osband, 2016), which aligns with our observations of undercoverage and overconfidence in Section 4.2. Verdoja and Kyrki (2021) demonstrate that model architecture, training and tuning choices heavily influence the estimated uncertainty. Furthermore, Sicking et al. (2020) observe that MC-Dropout may produce non-Gaussian uncertainty distributions in wide networks.
Another noteworthy approach is Bayes by Backprop (Blundell et al., 2015). This method places a distribution (typically Gaussian) over model weights, learned via variational inference. The resulting weight uncertainty can then be used to generate predictive uncertainty estimates through weight sampling.

Ensemble Methods Deep Ensembles (Lakshminarayanan et al., 2017) combine predictions from multiple independently trained models and consistently outperform MC-Dropout, especially under dataset shifts, as shown by Fort et al. (2020). They require high computational and memory costs, which makes them less practical for large-scale applications.
Stochastic Weight Averaging Gaussian (SWAG), introduced by Maddox et al. (2019), constructs a Gaussian posterior over weights based on Stochastic Gradient Descent (SGD) trajectory statistics. As validated in our experiments (Section 4.2), SWAG demonstrates improved calibration and robustness compared to MC-Dropout while maintaining reasonable computational overhead. The Extension MultiSWAG (Onal et al., 2024) bridges the gap between SWAG and full ensembles.

Approximate Inference Methods like Laplace approximations, expectation propagation perform local Gaussian posterior approximation around a maximum a posteriori (MAP) estimation. These methods yield competitiveperformance in both efficiency and uncertainty quality (see, e.g., Ritter et al., 2018). Linearized Laplace (Antorán et al., 2022), and more efficient variants such as Laplace Redux (Daxberger et al., 2022) offer scalable posterior estimation with good calibration and performance. These techniques offer a middle ground between simplicity and robustness.

This work focuses on MC-Dropout and SWAG, which cover both ends of the trade-off spectrum discussed in prior work. MC-Dropout for its ease-of-use and efficiency, and SWAG for its richer posterior structure and empirical strength.

3 Methodology

This section presents the theoretical foundations and detailed mechanics necessary to understand the methods used in Section 4.

3.1 Bayesian Neural Networks

Bayesian inference allows updating prior beliefs using previously unseen data via Bayes' theorem:

$$P(\theta \mid \mathcal{D}) = \frac{P(\mathcal{D} \mid \theta)P(\theta)}{P(\mathcal{D})} \quad \text{where} \quad P(\mathcal{D}) = \int P(\mathcal{D} \mid \theta')P(\theta')d\theta' \tag{1}$$

Here, $P(\theta)$ is the prior distribution over parameters, $P(\mathcal{D} \mid \theta)$ is the likelihood (encoding aleatoric uncertainty), and $P(\theta \mid \mathcal{D})$ is the posterior (capturing epistemic uncertainty) (Jospin et al., 2022). This formalism enables quantification of both uncertainty types.

Before introducing Bayesian neural networks, we briefly recall the fundamentals of standard neural networks. Artificial neural networks (NNs) are inspired by biological brains, they consist of layered interconnected neurons. Each neuron computes a weighted sum of inputs x_i plus a bias b and applies a nonlinear activation function $g(\cdot)$:

$$z = \sum_i w_i x_i + b, \quad \text{output} = g(z)$$

In a deep neural network, neurons are organized in multiple layers. During training, the network adjusts its parameters (weights $\mathbf{W}$ and biases $\mathbf{b}$) via backpropagation and gradient descent to minimize a loss function on training data (Haykin, 2009, López et al., 2022). These learned parameters are treated as fixed point estimates at test time, and the network produces a single prediction without expressing uncertainty. So while NNs are powerful in pattern recognition, they lack a mechanism to quantify confidence in their predictions.

Bayesian Neural Networks (BNNs) address this limitation by treating weights $\mathbf{W}$ as random variables with specified prior distributions. Given training data $\mathcal{D} = \{(x_i, y_i)\}$, Bayes' theorem yields the posterior distribution:

$$p(\mathbf{W} \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \mathbf{W})p(\mathbf{W})}{p(\mathcal{D})} \tag{2}$$

Predictions incorporate uncertainty through marginalization over the posterior:

$$p(y^* \mid x^*, \mathcal{D}) = \int p(y^* \mid x^*, \mathbf{W})p(\mathbf{W} \mid \mathcal{D})d\mathbf{W} \tag{3}$$

Unlike deterministic NNs, BNNs output predictive distributions where variance reflects model confidence. Narrow distributions indicate high certainty, while broad distributions reveal epistemic uncertainty, especially in regions with limited data (Jospin et al., 2022). Training a BNN involves specifying a prior $p(\mathbf{W})$ and computing the posterior $p(\mathbf{W} \mid \mathcal{D})$. However, computing the exact posterior is not feasible for modern deep networks because it requires calculating $p(\mathcal{D})$ in Eq. (2), which is the integral shown in Eq. (1). It

integrates over the weight space, for modern NNs with millions of weights, this integral requires massive amounts of computing power. Furthermore, predictions require solving yet another high-dimensional integral (Eq. (3)) over the same parameter space. Consequently, approximate inference methods are essential, as discussed in subsequent sections.

3.2 Dropout Regularization

Dropout, introduced by Srivastava et al. (2014), is a regularization technique for neural networks. During training, dropout temporarily removes a random subset of neurons by multiplying their activations with independent Bernoulli-distributed variables:

$$r_j^{(l)} \sim \text{Bernoulli}(p), \quad \tilde{\mathbf{y}}^{(l)} = \mathbf{r}^{(l)} \odot \mathbf{y}^{(l)}$$

where p is the retention probability (dropout rate $= 1-p$) and $\mathbf{r}^{(l)}$ is a binary mask vector for layer l.
A key motivation for dropout is to prevent co-adaptation. In standard neural networks, neurons may develop complex interdependencies that cause them to function only in specific combinations. This means if one neuron fails (e.g., due to a noisy input), the entire model may break down, leading to poor generalization and overfitting (Hinton et al., 2012). Dropout improves robustness by forcing neurons to learn useful features independently.
This process effectively trains an ensemble of exponentially many "thinned" subnetworks that share weights, which prevents complex co-adaptation of features and improves generalization (Srivastava et al., 2014).

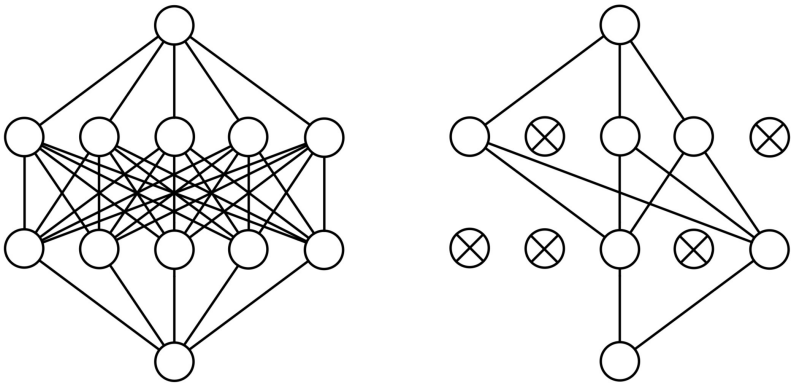

Figure 1: Visualization of the dropout mechanism: (a) Standard neural network with two hidden layers; (b) Thinned network after applying dropout masks. Crossed neurons are temporarily deactivated during training.
This figure was manually reillustrated with minor adjustments, the original version was published by Srivastava et al. (2014)

At test time, averaging across all possible thinned subnetworks is infeasible due to the exponential growth of possible configurations with increasing network size. Instead, dropout is disabled, resulting in a single network. Each of the network's weights is scaled by p,

which approximates the geometric mean of predictions from all subnetworks without the heavy computation needed for explicit averaging.

Unlike Monte Carlo Dropout, standard dropout functions purely as a regularizer, which means it improves generalization and reduces overfitting but doesn't provide uncertainty estimates (Srivastava et al., 2014).

3.3 Monte Carlo Dropout

Gal and Ghahramani (2016) introduced Monte Carlo dropout (MC-Dropout) as an efficient method to estimate uncertainty in neural networks. Unlike classic dropout, MC-Dropout keeps dropout active during testing. It performs multiple stochastic forward passes through the network, each with different randomly activated neurons, effectively sampling predictions $\{f^{\hat{\mathbf{W}}_t}(x^*)\}_{t=1}^{T}$ from T thinned subnetworks. Each subnetwork's weights $\hat{\mathbf{W}}_t$ can be thought of as a sample from the posterior $p(\mathbf{W} \mid \mathcal{D})$ and each prediction $f^{\hat{\mathbf{W}}_t}(x^*)$ as a sample from the predictive distribution $p(y^* \mid x^*, \mathcal{D})$. The predictive distribution quantifies the uncertainty about the models response for a new observation x^* given the observed data.
The collection of these predictions forms a distribution where the mean approximates the expected output and the variance quantifies epistemic uncertainty. Tightly clustered predictions indicate high confidence, while dispersed predictions reveal model uncertainty.

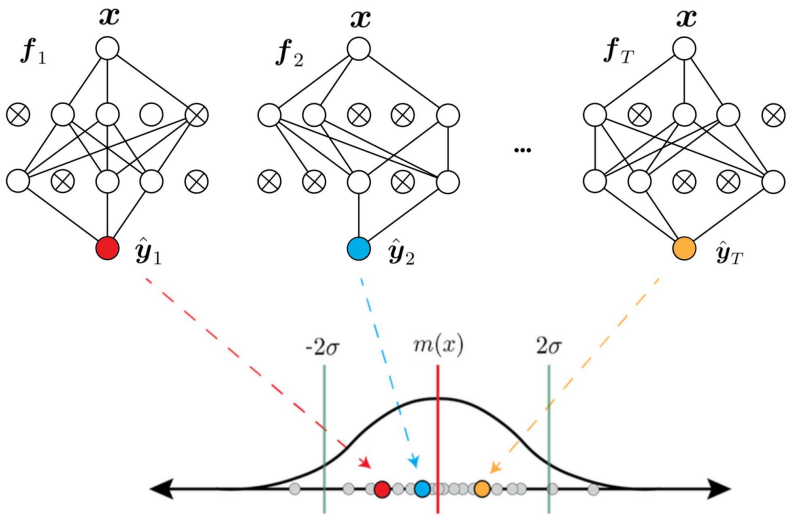

Figure 2: MC-Dropouts predictive distribution. Colored and grey dots represent predictions from different thinned networks, crossed neurons are deactivated.
This figure was manually reillustrated with minor adjustments, the original version was published by Van Katwyk et al. (2023)

MC-Dropout provides a Bayesian approximation without formal Bayesian neural networks. By interpreting dropout as variational inference, it mimics posterior sampling through the predictive distribution of subnetworks (Gal and Ghahramani, 2016).

Advantages: MC-Dropout introduces minimal computational overhead since it requires only one trained model and uses the same computational graph with dropout enabled

during inference. It scales efficiently, providing reasonable uncertainty estimates with significantly lower cost than full ensembling. Implementation is straightforward in popular frameworks like PyTorch and TensorFlow.

Limitations: The method tends to underestimate uncertainty in data-sparse regions and may not concentrate appropriately as the amount of data increases (Osband, 2016). The estimated uncertainty is sensitive to changes in hyperparameters (e.g. retention probability p) (Verdoja and Kyrki, 2021). As a variational method, MC-Dropout primarily explores uncertainty around a single point (mode) in the posterior space, potentially underestimating the true posteriors variance. This makes sense because the diversity among subnetworks is low, since all of them are dropout versions of the main network. In contrast, deep ensembles with independently trained models capture multiple modes, offering greater predictive diversity and robustness to distribution shifts (Fort et al., 2020).

Despite its limitations, MC-Dropout remains a popular baseline for practical uncertainty quantification due to its simplicity, low computational cost (especially compared to ensemble techniques, as the model is trained only once), and compatibility with other uncertainty estimation methods. It is particularly advantageous when training resources are limited or when the true posterior distribution is expected to be simple.

3.4 Stochastic Weight Averaging Gaussian

Stochastic Weight Averaging with Gaussian posterior approximation (SWAG) builds on Stochastic Weight Averaging (SWA), a technique introduced by Izmailov et al. (2019). The core idea behind SWA is, that after training a neural network normally, one restarts stochastic gradient descent (SGD) from that trained state using a constant or cyclical learning rate to explore the weight space. The weights w_i obtained after each iteration i of the SGD algorithm are periodically saved and once the algorithm finishes after T or more iterations, the SWA mean (or SWA solution) is computed:

$$\mathbf{w}_{\text{SWA}} = \frac{1}{T}\sum_{i=1}^{T}\mathbf{w}_i$$

Izmailov et al. (2019) argue, that the SWA mean converges toward the center of a wide, flat region in the loss surface, which leads to a more robust model and better generalization compared to the solution that conventional SGD produces.

SWAG (Maddox et al., 2019) extends SWA by constructing a Gaussian posterior approximation from these weight snapshots. It estimates not just the mean $\mathbf{w}_{\text{SWA}}$, but also a covariance matrix capturing parameter uncertainty. The covariance comprises two complementary components:

1. **Diagonal covariance (Σ_{diag}):** Computes per-parameter variance $\Sigma_{\text{diag}} = \text{diag}\left(\frac{1}{T-1}\sum_{i=1}^{T}(\mathbf{w}_i - \mathbf{w}_{\text{SWA}})^2\right)$.
   This captures independent uncertainty estimates for individual weights.

2. **Low-rank covariance ($\Sigma_{\text{low-rank}}$):** Constructed from deviation vectors $\Sigma_{\text{low-rank}} = \frac{1}{2(T-1)}\mathbf{D}\mathbf{D}^\top$ where $\mathbf{D} = [\mathbf{w}_1 - \mathbf{w}_{\text{SWA}}, \ldots, \mathbf{w}_T - \mathbf{w}_{\text{SWA}}]$.
   This low-rank approximation captures dominant correlations between parameters.

The combined covariance $\Sigma = \Sigma_{\text{diag}} + \Sigma_{\text{low-rank}}$ addresses limitations of purely diagonal approximations, which often underestimate uncertainty by ignoring parameter correlations (Maddox et al., 2019). The full posterior approximation is:

$$q_{\text{SWAG}}(\mathbf{w}) = \mathcal{N}\left(\mathbf{w}_{\text{SWA}}, \Sigma_{\text{diag}} + \Sigma_{\text{low-rank}}\right)$$

At test time, SWAG samples weights $\mathbf{w}^{(s)} \sim q_{\text{SWAG}}$ and computes predictions $f^{\mathbf{w}^{(s)}}(x^*)$ for each sample. The resulting empirical distribution estimates the predictive distribution $p(y^* \mid x^*, \mathcal{D})$, which as stated in Section 3.3 quantifies the uncertainty about the models response for a new observation given the observed data, its mean approximates the expected output and its variance the epistemic uncertainty.

Advantages: By analyzing the SGD weight trajectories, SWAG captures posterior structure around flat minima in the loss surface. While more computationally intensive than MC-Dropout, since it requires extended training for weight exploration and covariance calculation, it remains significantly cheaper than full ensembles because only one model is trained. Maddox et al. (2019) demonstrate its superior uncertainty calibration and robustness to distribution shifts compared to MC-Dropout and other baselines, particularly on vision benchmarks like CIFAR and ImageNet.

Limitations: Although SWAG approximates uncertainty better than simpler methods, it still, similar to MC-Dropout, represents a single mode of the posterior and does not learn true multimodal structures (Onal et al., 2024). Memory requirements increase with snapshot frequency due to covariance storage, though remain manageable versus full ensembles. The quality of the uncertainty estimation depends on the learning rate set for the exploration phase (Maddox et al., 2019).

SWAG thus provides an efficient midpoint between simple variational methods and computationally intensive approaches, offering structured uncertainty estimation with minimal training overhead beyond standard SGD.

4 Experiments

This section applies the methods introduced in Section 3 to both synthetic and real-world datasets and evaluates their capabilities in predictive accuracy, uncertainty calibration, and out-of-distribution (OOD) detection.

4.1 Synthetic Dataset

To evaluate MC-Dropout and SWAG for uncertainty quantification, experiments are conducted on a synthetic regression and binary classification task. Both tasks used a neural network with two fully-connected hidden layers (32 neurons each with ReLU activations). For MC-Dropout, dropout layers are added after each hidden layer.

Regression Task A synthetic dataset is generated following:

$$y = \sin(x) + 0.3\varepsilon, \quad \varepsilon \sim \mathcal{N}(0, 1)$$

Both models are trained on $x \in [-5, 5]$ and evaluated on $x \in [-8, 8]$.

As shown in Figure 3, both methods produce accurate mean predictions within the training region ($|x| \leq 5$), closely approximating $\sin(x)$. SWAG estimates thinner uncertainty intervals than MC-Dropout, aligning with findings by Maddox et al. (2019).
Outside this region, the uncertainty bands increase as both methods detect OOD inputs. MC-Dropout exhibits faster uncertainty growth, with uncertainty intervals widening more rapidly than SWAG beyond $|x| > 5$. SWAG maintains tighter confidence bands while still showing some uncertainty expansion.
The sample points form vertical patterns ("pillars") in both cases. These result from 200 stochastic predictions being sampled at each of the 100 equidistant test points along the x-axis. For MC-Dropout, predictions come from different thinned subnetworks, while for SWAG they derive from weight samples $\mathbf{w}^{(s)} \sim q_{\text{SWAG}}$.
SWAG's samples follow a certain structure, where connecting e.g. the maximum prediction at each test point will form a smooth curve. This is because these samples come from the same set of weights, i.e. the same model. MC-Dropout's more irregular dispersion reflects the random neuron deactivations from different dropout masks (Gal and Ghahramani, 2016).

These observations align with theoretical expectations, MC-Dropout's sampling produces noisier but more conservative uncertainty estimates (Gal and Ghahramani, 2016), while SWAG's weight-space sampling provides a geometrically structured uncertainty quantification. Both methods distinguish between in-domain confidence and out-of-domain uncertainty.

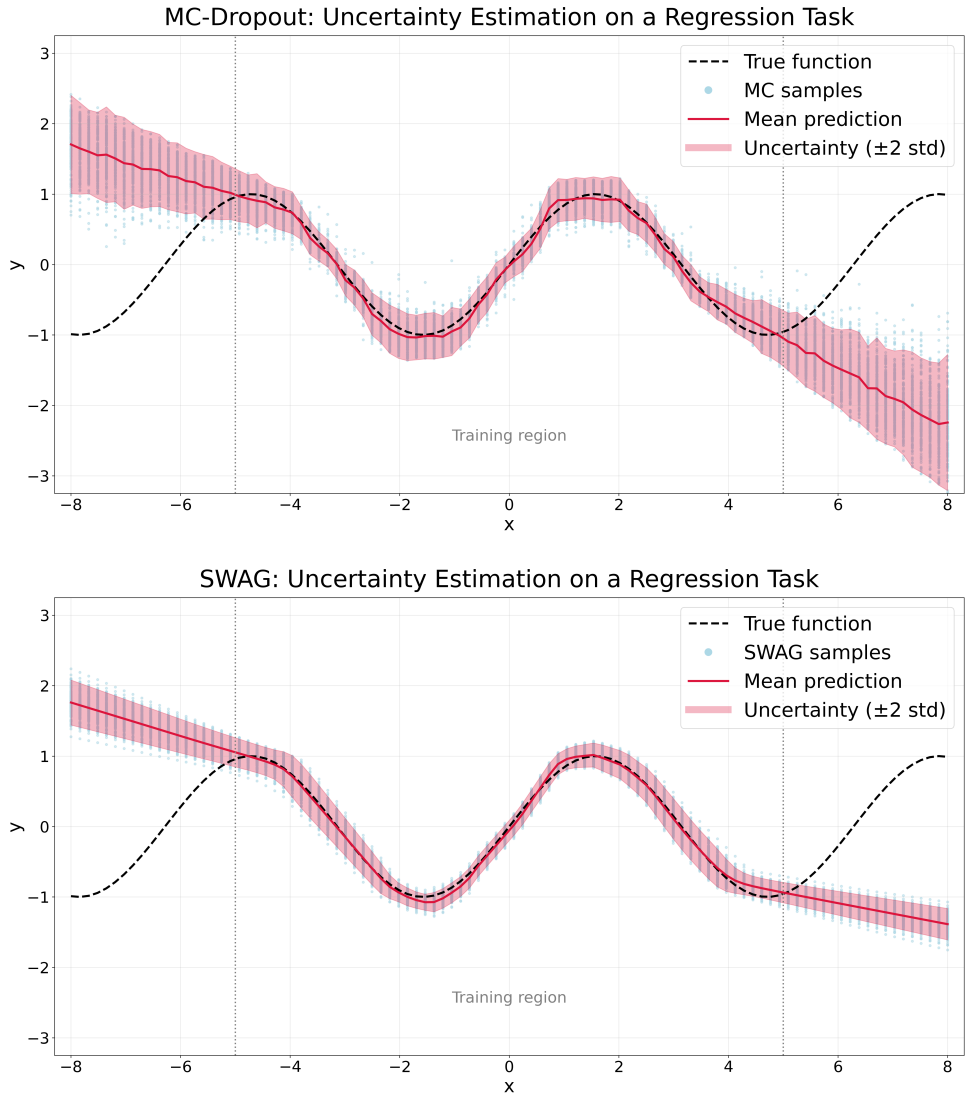

Figure 3: Regression uncertainty estimation: MC-Dropout (top) and SWAG (bottom). Training domain ($|x| \leq 5$) marked by dashed lines.

Classification Task The binary classification task uses a "two moons" dataset generated with scikit-learn (Pedregosa et al., 2011), it contains 500 training points with a nonlinear class boundary.

Figure 4 shows both methods assign low uncertainty near training data, with sharp decision boundaries (zone where the model transitions from predicting one class to the other). Both increase predictive uncertainty in regions far from training data, successfully identifying OOD areas. SWAG provides a smoother uncertainty estimation, while MC-Dropout produces a more scattered predictive uncertainty, especially in the extrapolation region. Notably, some noisy points located in the wrong class' region (at approximately $(1.5, -0.5)$ and $(0.5, 0.5)$) induce uncertainty spikes in SWAG but not MC-Dropout. This demonstrates SWAG's sensitivity to weight-space perturbations (Izmailov et al., 2019), as its covariance propagates uncertainty from anomalies to neighboring regions. Near these noisy

points, SWAG shows broader uncertainty intervals while maintaining thinner boundaries elsewhere (e.g., near $(0, 0)$).

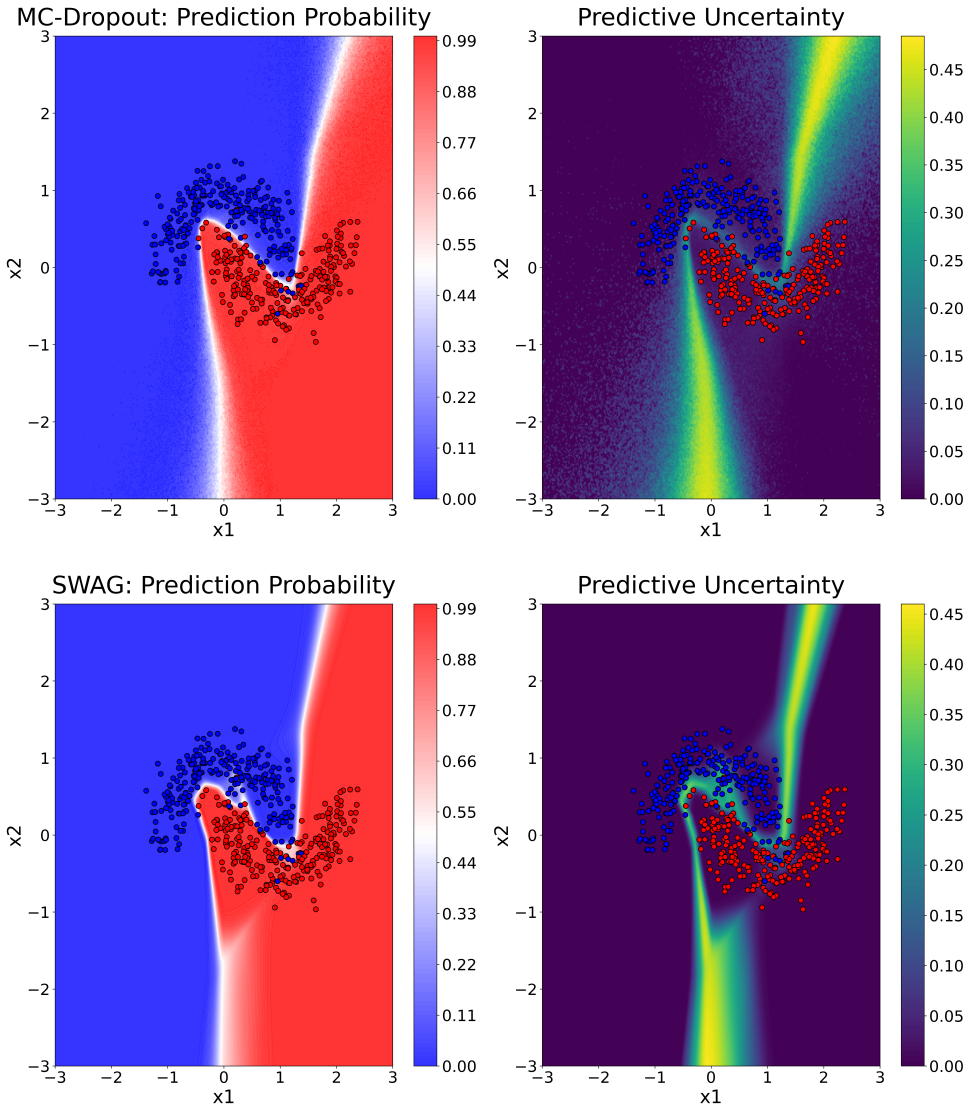

Figure 4: Classification uncertainty: MC-Dropout (top) and SWAG (bottom). Color indicates true class (red/blue).

4.2 Evaluation on Real-world Data

To assess practical performance, the Boston Housing dataset is used (Harrison and Rubinfeld, 1978). It consists of 506 observations of housing prices in Boston suburbs. The response variable is the median home value (in $1000s), with features including crime rate, average rooms and highway accessibility. Following ethical research standards, the racially problematic `B` variable is excluded. It encoded the proportion of Black residents and reflected discriminatory assumptions in its original formulation (see, e.g., Carlisle, 2020).

Model Architecture and Training A neural network with four fully connected hidden layers is trained to predict standardized housing prices. Hyperparameter tuning is conducted using a randomized grid search on a previously unseen validation set. For MC-Dropout, dropout layers are once again added after each hidden layer. Both models are trained using the Adam optimizer with a tuned learning rate and weight decay.
Table 1 summarizes the evaluation results on the test set using three key metrics: Root Mean Squared Error (RMSE) measures predictive accuracy (lower values preferred), Negative Log-Likelihood (NLL) assesses uncertainty calibration (lower values indicate better calibration) and Prediction Interval Coverage Probability (PICP) evaluates the reliability of 95 % uncertainty intervals (values closer to 95 % reflect superior uncertainty estimation).

Method	RMSE	NLL	PICP (95 %)
MC-Dropout	0.4033	0.6814	0.7352
SWAG	0.3947	0.2218	0.9314

Table 1: Test performance on the Boston Housing dataset.

Both methods achieve comparable point-prediction accuracy, as evidenced by their similar RMSE values. However, SWAG demonstrates significantly better uncertainty calibration, with a much lower NLL and a near-ideal PICP of 93.14 %. MC-Dropout's PICP of 73.52 % indicates systematic undercoverage. The notable discrepancy in NLL values further reveals MC-Dropout's tendency toward overconfidence, as it frequently assigns high certainty to incorrect predictions.

Uncertainty Behavior Analysis Figure 5 provides a comparative analysis of uncertainty characteristics between MC-Dropout and SWAG.
The top-left subplot displays histograms of predictive standard deviations, revealing distinct distributional profiles. MC-Dropout exhibits a sharp peak near 0.16, indicating overconfident estimates with limited variability, whereas SWAG generates a broader distribution centered near 0.2 with an extended right tail, reflecting more moderate and diverse uncertainty quantification.
In the top-right scatter plot of absolute residuals versus predictive standard deviations, SWAG displays a wider vertical spread of points, suggesting that it assigns greater uncertainty to predictions with larger errors. On the other hand, MC-Dropout's points are

more tightly clustered near the origin, indicating uniformly low uncertainty estimates even when residuals are non-negligible. This pattern reinforces the impression of overconfidence in MC-Dropout's uncertainty outputs.
The bottom row consists of two calibration plots, where predicted means are plotted against true target values and shaded confidence bands represent 95 % predictive intervals. While both models align well along the identity line in high-density regions, SWAG produces slightly wider confidence bands. This demonstrates more conservative and better-calibrated uncertainty estimates.

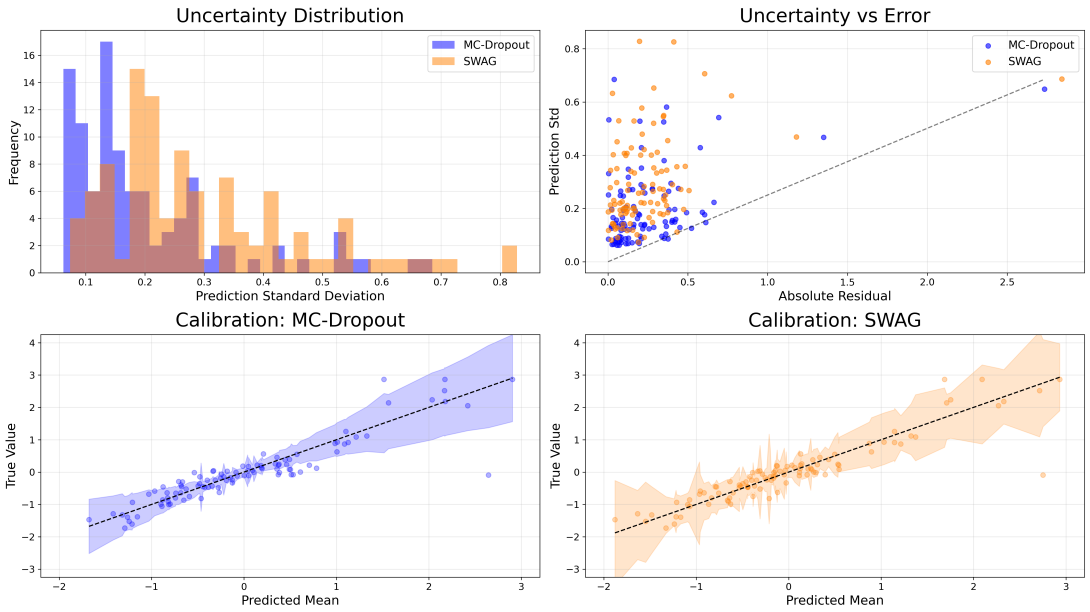

Figure 5: Comparison of uncertainty behaviors for MC-Dropout and SWAG on the Boston Housing dataset.

Out-of-Distribution Analysis To evaluate the robustness of uncertainty estimation under distribution shift, two types of out-of-distribution (OOD) scenarios are examined. Natural OOD samples with exceptionally high median income (`MedInc`) and artificially perturbed samples with amplified income and reduced occupancy. Table 2 displays predictive accuracy (RMSE) and uncertainty calibration (NLL) across in-domain and OOD conditions.

	MC-Dropout		SWAG	
Scenario	RMSE	NLL	RMSE	NLL
In-domain	0.4033	0.6814	0.3947	0.2218
Natural OOD	0.2768	0.2331	0.3838	0.6708
Artificial OOD	0.4283	0.7249	0.4094	0.3006

Table 2: OOD performance comparison.

For natural OOD samples, MC-Dropout shows point-prediction accuracy with the RMSE

decreasing from 0.4033 to 0.2768, but exhibits a substantial NLL reduction (0.6814 to 0.2331). This divergence suggests potentially overconfident uncertainty estimates that fail to properly reflect distribution shift. In contrast, SWAG maintains stable predictive accuracy (RMSE: 0.3947 to 0.3838) while appropriately increasing NLL (0.2218 to 0.6708), indicating more cautious uncertainty estimation under distribution shift.
In the artificial OOD scenario, both methods show degraded performance. MC-Dropout's RMSE increases to 0.4283 and NLL to 0.7249, while SWAG's RMSE increases to 0.4094 and NLL to 0.3006. The increased NLL indicates worse uncertainty calibration compared to in-domain samples and SWAG maintains its superior calibration. Both methods show a very similar relative response under artificial perturbations.
Figure 6 complements the tabular metrics by visualizing the distribution of predictive standard deviations (i.e., uncertainty) shifts between in-domain and artificial OOD conditions. For MC-Dropout (left panel), the histogram shows negligible distributional change with marginally increased maximum uncertainty, indicating limited responsiveness to distribution shift. SWAG (right panel) displays a substantial increase in maximum uncertainty (33 %), this demonstrates that SWAG has the capacity to effectively register distributional changes and inflate uncertainty accordingly (Maddox et al., 2019).

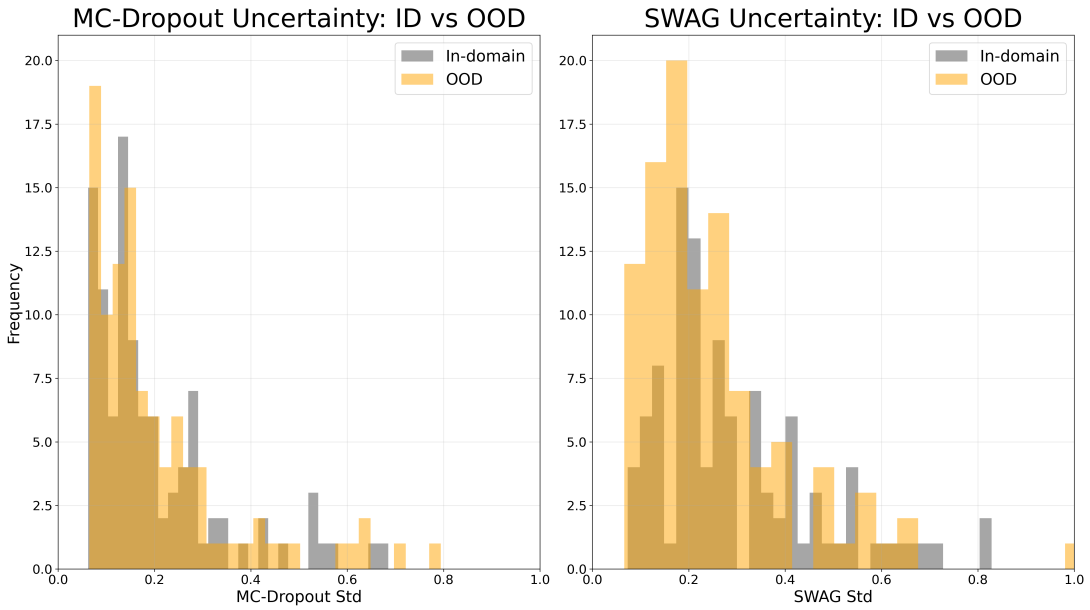

Figure 6: Uncertainty comparison between in-domain and out-of-distribution (OOD) samples for MC-Dropout (left) and SWAG (right).

The Boston Housing evaluation demonstrates distinct characteristics between SWAG and MC-Dropout. Both methods achieve comparable point-prediction accuracy, but SWAG exhibits superior uncertainty calibration as evidenced by lower NLL and near-ideal PICP. Uncertainty visualization reveals SWAG's more diverse uncertainty distribution and stronger error-uncertainty correlation. Under distribution shift, SWAG maintains stable accuracy while appropriately inflating uncertainty, whereas MC-Dropout shows concerning overconfidence.

5 Discussion

This work systematically compares MC-Dropout and SWAG for uncertainty quantification, revealing a fundamental trade-off between computational efficiency and and the quality of uncertainty estimates. MC-Dropout's advantages lie in computational efficiency, generating uncertainty estimates through simple stochastic forward passes. It however shows limitations including undercoverage and overconfidence under distribution shift. SWAG addresses these issues through SGD-based weight-space exploration, yielding better-calibrated uncertainties with improved responsiveness to distributional changes, but requiring extended training and increased memory overhead.

Given these differences, our findings suggest practical recommendations for choosing between MC-Dropout and SWAG. MC-Dropout suits applications prioritizing speed and simplicity where uncertainties play a secondary role, particularly in stable environments. Its deficient OOD performance (Table 2) makes it unsuitable for safety-critical systems. SWAG proves preferable when the reliability of uncertainty is central to decision making, especially where training budget permits extended exploration and deployment contexts involve potential distribution shifts.

Looking ahead, both methods present opportunities for improvement. Extensions of MC-Dropout could reduce its overconfidence through techniques such as Concrete Dropout (Gal et al., 2017), which learns layer-specific retention probabilities to better calibrate uncertainty. SWAG can be further improved in terms of expressiveness and efficiency. MultiSWAG (Onal et al., 2024), for instance, ensemble the several independently trained SWAG models to approximate multimodal posteriors, while SWALP (Yang et al., 2019) enables low-precision inference, reducing computational cost without degrading performance. For applications where multimodality capture is essential, Deep Ensembles (Lakshminarayanan et al., 2017) remain superior despite high computational costs, with MultiSWAG offering a promising balance between efficiency and robustness.

The selection of an uncertainty quantification method should ultimately be guided by the specific requirements of the target application. MC-Dropout is well suited for lightweight and scalable solutions, SWAG for reliable calibration, and ensemble variants where resources permit robust uncertainty modeling.

6 Conclusion

Through rigorous evaluation across synthetic and real-world datasets, this work has illustrated a fundamental trade-off in uncertainty estimation quality between MC-Dropout and SWAG. The systematic comparison reveals distinct characteristics that guide practical method selection.

Our experiments demonstrate that both methods effectively capture predictive uncertainty within their training distribution. In synthetic tasks (Figures 3, 4), they successfully identify out-of-distribution regions while maintaining accurate in-domain predictions. SWAG produced smoother uncertainty estimates in both regression and classification tasks, while MC-Dropout yielded noisier predictions. In the regression task specifically, MC-Dropout generated more conservative uncertainty intervals.
When applied to the Boston Housing dataset (Section 4.2), critical differences emerged. SWAG produced better-calibrated uncertainty estimates, achieving near-ideal prediction interval coverage (Table 1) and appropriately increasing uncertainty under distributional changes (Table 2). MC-Dropout showed undercoverage and concerning overconfidence in OOD scenarios. SWAG's uncertainty distributions showed greater diversity and stronger error correlation (Figure 5), while providing more reliable confidence intervals.

Ultimately, this work demonstrates that Bayesian approximations don't need to sacrifice practical utility, as both methods provide valuable uncertainty estimates, with SWAG offering superior calibration at moderately higher computational overhead. As neural networks increasingly support critical decision-making, selecting appropriate uncertainty quantification methods becomes essential for developing trustworthy AI systems.

A Appendix

A.1 Usage of AI

OpenAI's free LLM, ChatGPT-4o, was used to improve the grammar and wording in this work and to document the code. It also assisted in understanding the implementation of SWAG as described by Maddox et al. (2019) (`https://github.com/wjmaddox/swa_gaussian/`). The tool was used to aid editorial refinements and technical understanding, not to generate original research content.

A.2 Additional plots

Tuned NN on synthetic regression and classification This appendix presents uncertainty estimates from the same model architectures used in Section 4.1, now with optimized hyperparameters. Figures 7 and 8 show uncertainty visualizations comparable to their non-tuned counterparts in Figures 3 and 4.
In the regression task (Figure 7), both methods exhibit visibly reduced uncertainty compared to non-tuned results, evidenced by tighter confidence intervals throughout the domain. For classification (Figure 8), SWAG shows substantially thinner uncertainty estimates across the input space. MC-Dropout, however, maintains nearly identical uncertainty patterns to its non-tuned version, with only subtle adjustments to the decision boundary attributable to improved model performance rather than changes in uncertainty quantification behavior.

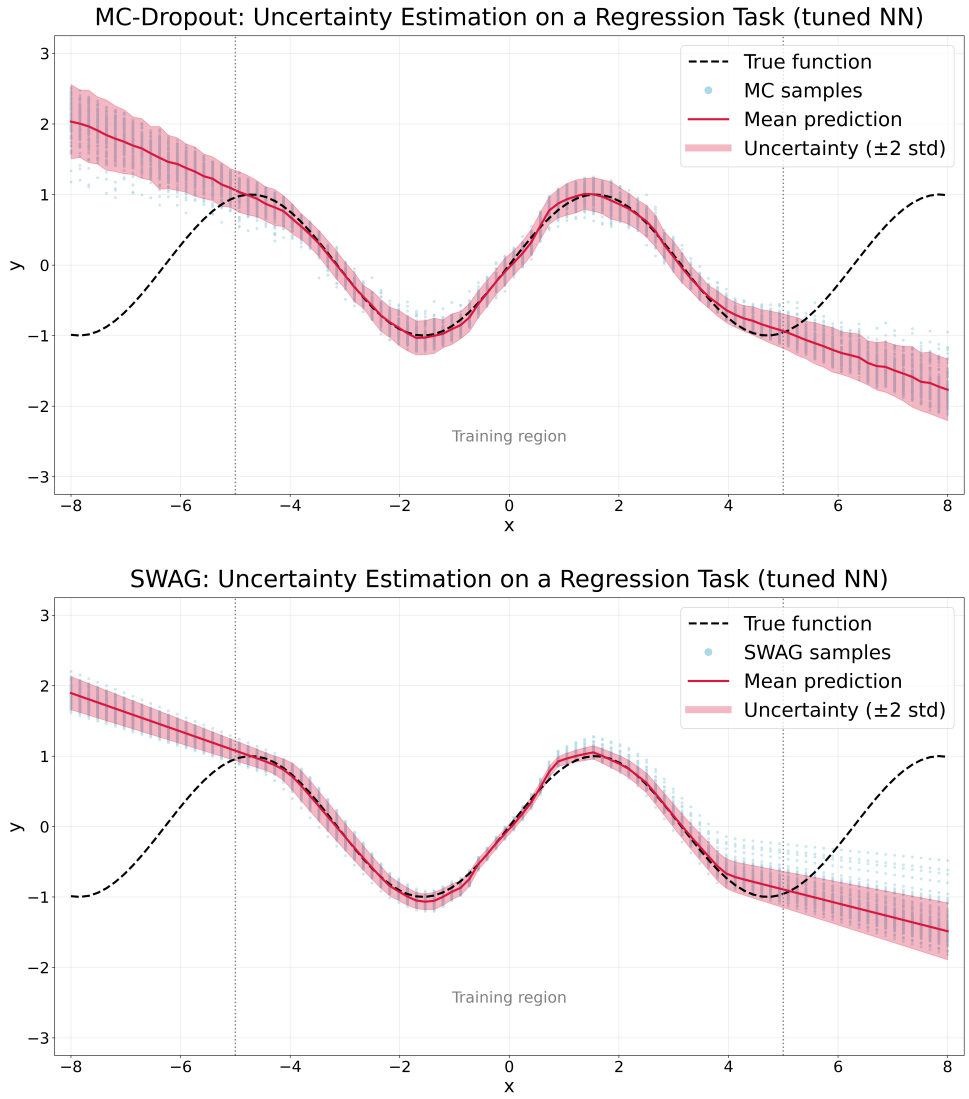

Figure 7: Regression uncertainty estimation of a tuned Neural Network: MC-Dropout (top) and SWAG (bottom). Training domain ($|x| \leq 5$) marked by dashed lines.

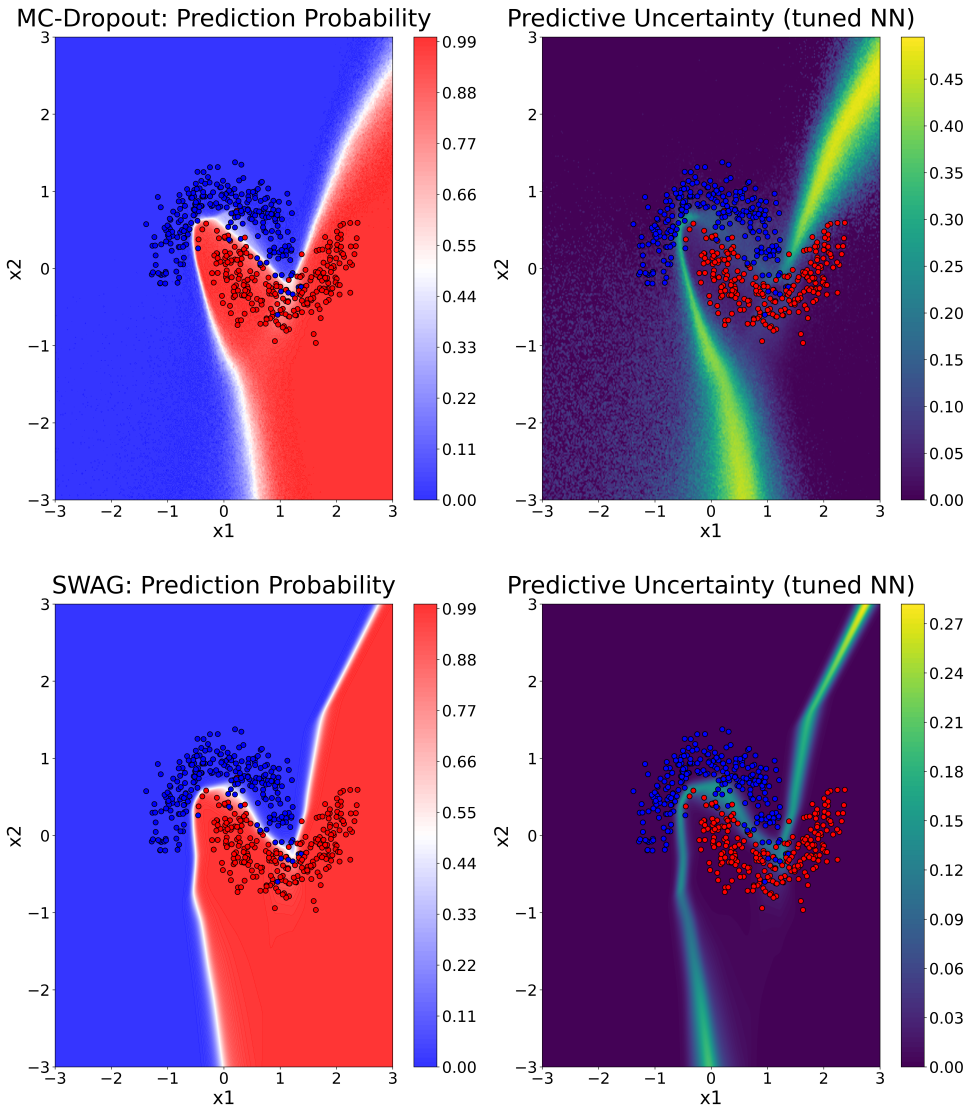

Figure 8: Classification uncertainty of a tuned Neural Network: MC-Dropout (top) and SWAG (bottom). Color indicates true class (red/blue).

Effect of Dropout Rate on MC-Dropout Uncertainty We investigate the sensitivity of MC-Dropout uncertainty estimates to the dropout rate hyperparameter using the synthetic regression task introduced in Section 4.1. While Figure 3 presented results with dropout rate = 0.2, we now systematically compare rates of 0.1, 0.3, 0.5 and 0.7. As shown in Figure 9, higher dropout rates produce progressively wider uncertainty intervals throughout the domain, particularly in extrapolation regions ($|x| > 5$). The mean prediction also increasingly deviates from the true function as dropout rate increases. These observations align with findings from Verdoja and Kyrki (2021), confirming that dropout rate selection critically impacts both uncertainty quantification and predictive accuracy.

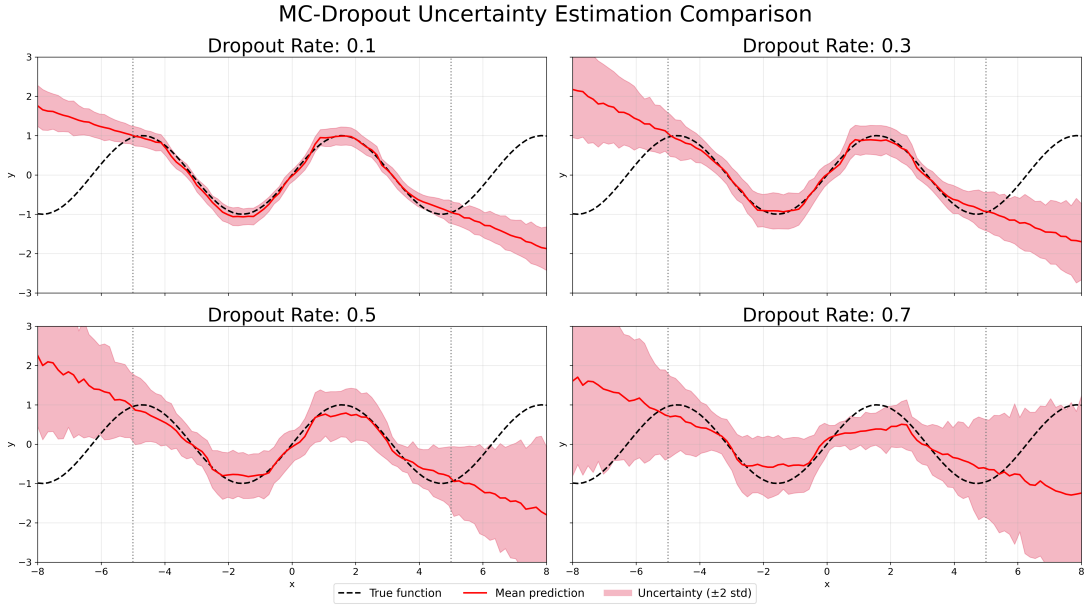

Figure 9: Uncertainty estimation using MC-Dropout with different dropout rates (0.1, 0.3, 0.5, 0.7) on the same regression task.

Predictions with 95 % Confidence Intervals on Boston Housing Figure 10 visualizes predictions against standardized true values with 95 % confidence intervals for both methods. The visualization reveals two significant patterns, SWAG exhibits consistently wider confidence intervals (orange bars), particularly pronounced for lower true values in the left side of the plot, confirming its more conservative uncertainty estimation approach. In contrast, MC-Dropout maintains narrower confidence bands that may fail to contain the true values, consistent with its undercoverage tendency. Both methods show generally accurate point predictions closely distributed around the identity line, with only two notable outliers where predictions significantly deviate from true values.

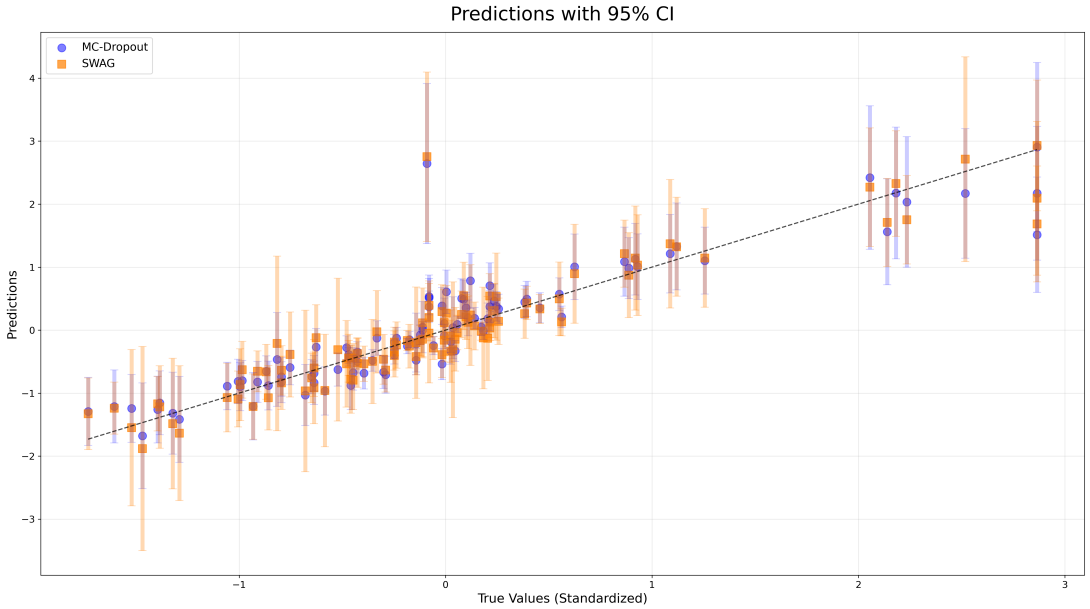

Figure 10: Predictions with 95 % confidence intervals on Boston Housing test set. Dashed line represents perfect predictions (y=x).

Residual Analysis on Boston Housing Figure 11 presents a diagnostic examination of model performance through residual analysis, where residuals represent the difference between observed and predicted values. The visualization reveals randomly distributed points for both MC-Dropout (blue) and SWAG (orange) that symmetrically scatter around the zero-reference line throughout the prediction range. Residual magnitudes demonstrate comparable dispersion for both methods, with no evidence of systematic patterns. This absence of structure indicates a well-calibrated model without significant bias or heteroskedasticity. The random error distribution confirms that both methods capture the underlying relationships in the data effectively, with prediction errors showing no directional tendency.

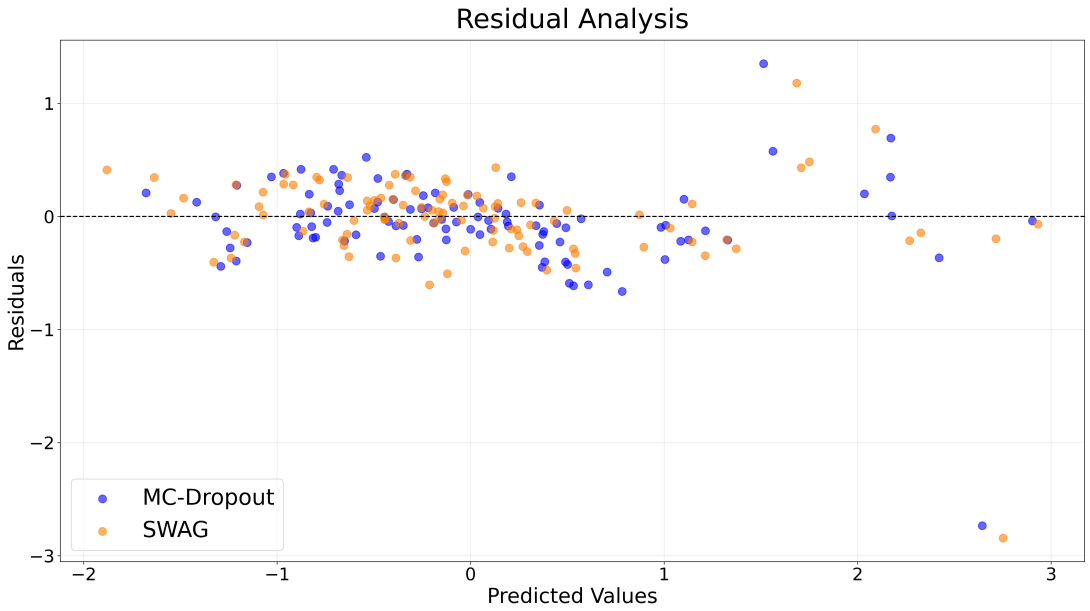

Figure 11: Residual analysis for Boston Housing predictions. Dashed line represents perfect prediction (residual = 0).

B Electronic appendix

The data, code and figures are available on `https://github.com/martin-javier/Seminar_ProbabilisticML`

References

Antorán, J., Janz, D., Allingham, J. U., Daxberger, E., Barbano, R., Nalisnick, E. and Hernández-Lobato, J. M. (2022). Adapting the linearised laplace model evidence for modern deep learning.
URL: *https://arxiv.org/abs/2206.08900*

Blundell, C., Cornebise, J., Kavukcuoglu, K. and Wierstra, D. (2015). Weight uncertainty in neural networks.
URL: *https://arxiv.org/abs/1505.05424*

Carlisle, M. (2020). Racist data destruction?, *Medium* .
URL: *https://medium.com/@docintangible/racist-data-destruction-113e3eff54a8*

Daxberger, E., Kristiadi, A., Immer, A., Eschenhagen, R., Bauer, M. and Hennig, P. (2022). Laplace redux – effortless bayesian deep learning.
URL: *https://arxiv.org/abs/2106.14806*

Fort, S., Hu, H. and Lakshminarayanan, B. (2020). Deep ensembles: A loss landscape perspective.
URL: *https://arxiv.org/abs/1912.02757*

Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning.
URL: *https://arxiv.org/abs/1506.02142*

Gal, Y., Hron, J. and Kendall, A. (2017). Concrete dropout.
URL: *https://arxiv.org/abs/1705.07832*

Harrison, D. and Rubinfeld, D. L. (1978). Hedonic prices and the demand for clean air, *Journal of Environmental Economics and Management* **5**: 81–102.

Haykin, S. (2009). *Neural Networks and Learning Machines*, 3 edn, Pearson Education.
URL: *https://www.pearson.com/en-us/subject-catalog/p/neural-networks-and-learning-machines/P200000003278*

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2012). Improving neural networks by preventing co-adaptation of feature detectors, *arXiv preprint arXiv:1207.0580* .
URL: *https://www.bibsonomy.org/bibtex/2e5f5aed38402149446e20df050243e30/hprop*

Hüllermeier, E. and Waegeman, W. (2021). Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods, *Machine Learning* **110**: 457–506.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D. and Wilson, A. G. (2019). Averaging weights leads to wider optima and better generalization.
URL: *https://arxiv.org/abs/1803.05407*

Jospin, L. V., Laga, H., Boussaid, F., Buntine, W. and Bennamoun, M. (2022). Hands-on bayesian neural networks—a tutorial for deep learning users, *IEEE Computational Intelligence Magazine* **17**: 29–48.
URL: *http://dx.doi.org/10.1109/MCI.2022.3155327*

Lakshminarayanan, B., Pritzel, A. and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles.
URL: *https://arxiv.org/abs/1612.01474*

López, O. A. M., López, A. M. and Crossa, J. (2022). *Fundamentals of Artificial Neural Networks and Deep Learning*, Springer International Publishing, pp. 379–425.
URL: *https://doi.org/10.1007/978-3-030-89010-0_10*

Maddox, W., Garipov, T., Izmailov, P., Vetrov, D. and Wilson, A. G. (2019). A simple baseline for bayesian uncertainty in deep learning.
URL: *https://arxiv.org/abs/1902.02476*

Onal, E., Flöge, K., Caldwell, E., Sheverdin, A. and Fortuin, V. (2024). Gaussian stochastic weight averaging for bayesian low-rank adaptation of large language models.
URL: *https://arxiv.org/abs/2405.03425*

Osband, I. (2016). Risk versus uncertainty in deep learning: Bayes, bootstrap and the dangers of dropout.
URL: *http://bayesiandeeplearning.org/2016/papers/BDL_4.pdf*

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E. (2011). Scikit-learn: Machine learning in python, *Journal of Machine Learning Research* **12**: 2825–2830.
URL: *https://scikit-learn.org/stable/*

Pereira, A. and Thomas, C. (2020). Challenges of machine learning applied to safety-critical cyber-physical systems, *Machine Learning and Knowledge Extraction* **2**(4): 579–602.

Ritter, H., Botev, A. and Barber, D. (2018). A scalable laplace approximation for neural networks.
URL: *https://iclr.cc/Conferences/2018/Schedule?showEvent=224*

Sicking, J., Akila, M., Wirtz, T., Houben, S. and Fischer, A. (2020). Characteristics of monte carlo dropout in wide neural networks.
URL: *https://arxiv.org/abs/2007.05434*

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting, *Journal of Machine Learning Research* **15**(56): 1929–1958.
URL: *http://jmlr.org/papers/v15/srivastava14a.html*

Valentin, R. (2024). Towards a framework for deep learning certification in safety-critical applications using inherently safe design and run-time error detection.
URL: *https://arxiv.org/abs/2403.14678*

Van Katwyk, P., Fox-Kemper, B., Seroussi, H., Nowicki, S. and Bergen, K. J. (2023). A variational lstm emulator of sea level contribution from the antarctic ice sheet, *Journal of Advances in Modeling Earth Systems* **15**: 6.

Verdoja, F. and Kyrki, V. (2021). Notes on the behavior of mc dropout.
URL: *https://arxiv.org/abs/2008.02627*

Yang, G., Zhang, T., Kirichenko, P., Bai, J., Wilson, A. G. and Sa, C. D. (2019). Swalp : Stochastic weight averaging in low-precision training.
URL: *https://arxiv.org/abs/1904.11943*

Declaration of authorship

I hereby declare that the report submitted is my own unaided work. All direct or indirect sources used are acknowledged as references. I am aware that the Thesis in digital form can be examined for the use of unauthorized aid and in order to determine whether the report as a whole or parts incorporated in it may be deemed as plagiarism. For the comparison of my work with existing sources I agree that it shall be entered in a database where it shall also remain after examination, to enable comparison with future Theses submitted. Further rights of reproduction and usage, however, are not granted here. This paper was not previously presented to another examination board and has not been published.

Munich, 06.07.2025
Location, date

Name