Contents

1 Abstract

Quantifying prediction uncertainty is essential for trustworthy machine learning, yet practitioners face a bewildering array of methods with limited comparative guidance. We present a systematic benchmark evaluating uncertainty quantification (UQ) approaches across classical statistical, parametric nonlinear, and modern data-driven methods. We compare Gaussian Process Regression (GP), Neural Network + Gaussian Mixture Model (NNGMM), and Neural Network + Bayesian Linear Regression (NNBR) on seven synthetic regression tasks spanning linear to nonlinear functions, with homoskedastic and heteroskedastic noise at four levels (1%, 2%, 5%, 10%), yielding 168 experimental configurations. GP regression achieved the best calibration with 88.8% average coverage (target: 95%) and well-calibrated intervals in 14/56 scenarios, though at computational cost. NNBR provided efficient uncertainty estimates with 78.2% coverage using post-hoc calibration. NNGMM showed poor calibration (61.2% coverage), highlighting challenges in mixture density networks. Coverage degraded less severely for heteroskedastic noise than anticipated, suggesting robustness of modern methods. Our interactive dashboard at [URL] provides detailed visualizations and a practical guide for method selection based on problem characteristics. This work establishes performance baselines for UQ methods and identifies key tradeoffs between calibration quality, computational efficiency, and model complexity.

Keywords: Uncertainty quantification, Gaussian processes, Bayesian neural networks, conformal prediction, prediction intervals, regression

2 Introduction

2.1 The Need for Uncertainty Quantification

Predictive models are ubiquitous in modern science and engineering, from materials discovery and drug development to climate modeling and financial forecasting. However, point predictions alone are insufficient for informed decision-makingwe must also quantify our confidence in these predictions.[1,2] Uncertainty quantification (UQ) transforms predictive models from black boxes into trustworthy tools by providing rigorous estimates of prediction reliability.

The importance of UQ becomes particularly evident in high-stakes applications. In medical diagnostics, a model that predicts disease outcomes must distinguish between cases where it is confident versus uncertain, enabling clinicians to request additional testing when needed.[3] In materials science, uncertainty estimates guide experimental resource allocation toward the most informative measurements.[4] In autonomous systems, quantified uncertainty enables safe operation by triggering human intervention when model confidence is low.

2.2 Types of Uncertainty

The uncertainty quantification literature distinguishes between two fundamentally different sources of uncertainty:[1,5]

2.2.1 Aleatoric Uncertainty

Aleatoric uncertainty (also called statistical or irreducible uncertainty) represents inherent randomness in the data generation process. This uncertainty arises from:

- Natural variability in measurements (e.g., sensor noise, experimental error)
- Stochastic processes in the underlying system
- Heterogeneous noise that varies across the input space

Importantly, aleatoric uncertainty cannot be reduced by collecting more training datait represents a fundamental limit on prediction accuracy. For example, if we measure a physical quantity with an instrument that has ś0.01 precision, this measurement uncertainty persists regardless of how many times we measure.

2.2.2 Epistemic Uncertainty

Epistemic uncertainty (also called model or reducible uncertainty) arises from incomplete knowledge about the true underlying function. Sources include:

- Limited training data in certain regions of input space
- Model misspecification (wrong model family chosen)
- Uncertainty in model parameters

Unlike aleatoric uncertainty, epistemic uncertainty can in principle be reduced by:

- Collecting more representative training data
- Using more flexible model families

- Improving parameter estimation procedures

The word "epistemic" derives from the Greek "" (episteme), meaning knowledgeepistemic uncertainty reflects what we could know but currently do not.[1]

2.3 Classical Approaches to Uncertainty Quantification

2.3.1 Bayesian Inference

Bayesian methods provide a principled probabilistic framework for uncertainty quantification by treating model parameters as random variables.[6,7] Given training data $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, Bayesian inference computes a posterior distribution over parameters:

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{D})} \tag{1}$$

where $p(\boldsymbol{\theta})$ is the prior distribution encoding initial beliefs, $p(\mathcal{D}|\boldsymbol{\theta})$ is the likelihood, and $p(\mathcal{D})$ is the marginal likelihood (evidence). Predictions for a new input $\mathbf{x}^*$ marginalize over parameter uncertainty:

$$p(y^*|\mathbf{x}^*, \mathcal{D}) = \int p(y^*|\mathbf{x}^*, \boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta} \tag{2}$$

This predictive distribution naturally captures both epistemic uncertainty (through the parameter posterior) and aleatoric uncertainty (through the likelihood model).

1. Markov Chain Monte Carlo (MCMC)

   For complex models where the posterior is intractable, Markov Chain Monte Carlo methods draw samples from $p(\boldsymbol{\theta}|\mathcal{D})$.[8] While MCMC provides asymptotically exact inference, it can be computationally expensive, requiring thousands of model evaluations. Recent advances include parallel tempering and Hamiltonian Monte Carlo for improved sampling efficiency.

2. Variational Inference

   Variational methods approximate the posterior with a simpler distribution $q(\boldsymbol{\theta})$ from a tractable family, minimizing the Kullback-Leibler divergence:[4]

   $$q^*(\boldsymbol{\theta}) = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\mathcal{D})) \tag{3}$$

   Variational inference trades some accuracy for computational efficiency, making it suitable for large-scale applications.[8]

2.3.2 Bootstrap Methods

Bootstrap resampling provides a distribution-free approach to uncertainty estimation.[9,10] The procedure:

1. Draw B bootstrap samples $\{\mathcal{D}_b\}_{b=1}^B$ by sampling with replacement from the training data
2. Train a model on each bootstrap sample to obtain parameters $\{\boldsymbol{\theta}_b\}_{b=1}^B$
3. Form prediction intervals using the empirical distribution of predictions

Bootstrap methods are particularly appealing because they make minimal distributional assumptions. However, recent work shows that raw bootstrap standard deviations often underestimate true uncertainty and require calibration.[11]

2.3.3 Prediction Intervals from Asymptotic Theory

For linear and nonlinear least-squares regression, prediction intervals can be derived from asymptotic normality of parameter estimates. If $\hat{\boldsymbol{\theta}}$ is the maximum likelihood estimate with covariance $\boldsymbol{\Sigma}$, approximate 95% prediction intervals are:

$$\hat{y} \pm 1.96\sqrt{\hat{\sigma}^2 + \mathbf{g}(\mathbf{x})^T \boldsymbol{\Sigma} \mathbf{g}(\mathbf{x})} \tag{4}$$

where $\mathbf{g}(\mathbf{x}) = \nabla_{\boldsymbol{\theta}} f(\mathbf{x}, \boldsymbol{\theta})|_{\hat{\boldsymbol{\theta}}}$ is the gradient of the model, $\hat{\sigma}^2$ is the residual variance (aleatoric), and $\mathbf{g}^T \boldsymbol{\Sigma} \mathbf{g}$ captures parameter uncertainty (epistemic).

This approach underlies MATLAB's `nlinfit` function and similar tools, providing computationally efficient uncertainty estimates for parametric models.

2.4 Modern Machine Learning Approaches

2.4.1 Gaussian Processes

Gaussian Process Regression (GPR) has emerged as a gold-standard method for uncertainty quantification.[12,13] A GP defines a distribution over functions:

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \tag{5}$$

where $m(\mathbf{x})$ is the mean function (often zero) and $k(\mathbf{x}, \mathbf{x}')$ is a covariance kernel encoding smoothness assumptions. Given training data, the posterior predictive distribution is Gaussian:

$$p(f^*|\mathbf{x}^*, \mathcal{D}) = \mathcal{N}(f^*|\mu^*, \sigma^{*2}) \tag{6}$$

with closed-form expressions for μ^* and σ^{*2}.

GPR provides well-calibrated uncertainty estimates and interpolates smoothly between observations. However, it faces scalability challenges for large datasets (cubic complexity in training set size) and can struggle with high-dimensional inputs.[12]

2.4.2 Bayesian Neural Networks

Bayesian Neural Networks (BNNs) extend neural networks by placing probability distributions over weights.[4,14] Recent advances focus on scalable approximate inference:

- **Variational BNNs** use variational inference with factorized Gaussian weight posteriors
- **Monte Carlo Dropout** treats dropout as approximate Bayesian inference[5]
- **Deep Ensembles** train multiple networks with different initializations[15]

A 2024 study comparing BNN methods for materials property prediction found that BNN-MCMC achieved performance competitive with GPR.[4] Physics-informed BNNs integrate governing equations as soft constraints, improving physical consistency of predictions.[16]

2.4.3 Neural Network Hybrid Methods

Recent methods combine neural networks' representational power with classical UQ techniques:

1. Neural Network + Gaussian Mixture Models (NNGMM)

   This approach uses a neural network for feature extraction, then models the output distribution as a Gaussian mixture in the learned representation. The mixture model captures multimodal predictive distributions and provides uncertainty estimates through mixture component variances.

2. Neural Network + Bayesian Linear Regression (NNBR)

   NNBR extracts nonlinear features using a neural network, then applies Bayesian linear regression in the feature space.[17] This decouples representation learning (neural network) from uncertainty quantification (Bayesian inference), enabling:

   - Efficient uncertainty estimation after feature learning
   - Post-hoc calibration using validation data
   - Analytically tractable predictive distributions

   These hybrid approaches often achieve better calibration than end-to-end neural methods while maintaining computational efficiency.

2.4.4 Conformal Prediction

Conformal prediction has emerged as a powerful distribution-free framework for uncertainty quantification.[18–20] Unlike Bayesian methods, conformal prediction makes no distributional assumptions and provides finite-sample validity guarantees.

The basic split conformal procedure:

1. Split data into training ($\mathcal{D}_{\text{train}}$) and calibration ($\mathcal{D}_{\text{cal}}$) sets
2. Train a model on $\mathcal{D}_{\text{train}}$
3. Compute nonconformity scores $s_i = |y_i - \hat{f}(\mathbf{x}_i)|$ on $\mathcal{D}_{\text{cal}}$
4. For target coverage $1 - \alpha$, find quantile $q = \text{Quantile}_{1-\alpha}(\{s_i\})$
5. Predict test point $\mathbf{x}^*$ with interval $[\hat{f}(\mathbf{x}^*) - q, \hat{f}(\mathbf{x}^*) + q]$

The resulting prediction intervals are guaranteed to achieve $\geq 1 - \alpha$ coverage on exchangeable test data, regardless of the base model.[18]

1. Recent Advances

   - **Conformalized Quantile Regression (CQR)**: Uses quantile regression to produce adaptive prediction intervals[21]
   - **Locally Adaptive Conformal Prediction (LACP)**: Adjusts interval width based on local data density
   - **Robust Conformal Prediction**: Extends coverage guarantees to adversarially perturbed inputs[20]

- **Conformal Prediction for Dynamics**: Novel algorithms for time-series and dynamical systems[19]

A 2025 study of chromatography modeling found that conformal predictors outperformed Gaussian processes for black-box uncertainty quantification.[21]

2.5 Challenges in UQ for Regression

Despite significant progress, several challenges remain:

2.5.1 Calibration Assessment

Unlike classification where calibration metrics are well-established (e.g., expected calibration error), regression lacks consensus on calibration evaluation.[22] Common approaches include:

- **Coverage**: Fraction of test points within prediction intervals (should equal nominal level)
- **Interval width**: Narrower intervals are preferable given adequate coverage
- **Calibration curves**: Plot empirical vs. predicted coverage at different confidence levels

However, these metrics can be misleading. A model with perfect average coverage may have poorly calibrated local uncertainties.

2.5.2 Heteroskedastic Noise

Many real-world problems exhibit heteroskedastic noise where variance depends on inputs. Classical methods assuming constant variance (homoskedasticity) produce miscalibrated intervals. Modern approaches address this through:

- Input-dependent noise models
- Quantile regression
- Separate models for mean and variance

2.5.3 Computational Scalability

Bayesian methods (MCMC, BNNs) and Gaussian processes face computational bottlenecks for large datasets. Active areas of research include:

- Variational sparse GPs[23]
- Stochastic variational inference for BNNs
- Amortized inference for fast predictions

2.5.4 Distribution Shift

Most UQ methods assume test data follows the training distribution. Under distribution shift (common in deployment), uncertainty estimates can be severely miscalibrated. Robust UQ under shift remains an open challenge.

2.6 Research Objectives

Despite the proliferation of UQ methods, practitioners lack systematic guidance for method selection. Critical questions remain:

1. **How do modern data-driven methods compare to classical approaches in calibration quality?**
2. **What is the tradeoff between computational efficiency and uncertainty calibration?**
3. **How robust are different methods to heteroskedastic noise?**
4. **Which methods generalize well across linear and nonlinear regression tasks?**

This work addresses these questions through a comprehensive benchmark study evaluating:

- Gaussian Process Regression (GP)
- Neural Network + Gaussian Mixture Model (NNGMM)
- Neural Network + Bayesian Linear Regression (NNBR)

across diverse regression scenarios with controlled noise characteristics and known ground truth.

3 Methods

3.1 Experimental Design

We designed a systematic benchmark to evaluate UQ methods across varying function complexity, noise characteristics, and noise magnitudes. Our experimental design included:

3.1.1 Dataset Generation

We generated seven synthetic regression datasets with known ground truth functions:

Linear Datasets:

1. **Line**: $f(x) = 0.8x + 0.1$
2. **Quadratic**: $f(x) = 0.5 - x + 2x^2$
3. **Cubic**: $f(x) = 0.5 + 0.5x - x^2 + 2x^3$

Nonlinear Datasets:

1. **Exponential Decay**: $f(x) = 2.0 \exp(-3.0x) + 0.5$
2. **Logistic Growth**: $f(x) = \frac{1.0}{1+\exp(-10.0(x-0.5))}$
3. **Michaelis-Menten**: $f(x) = \frac{1.0 \cdot x}{0.3+x}$
4. **Gaussian**: $f(x) = \exp\left(-\frac{(x-0.5)^2}{2(0.15)^2}\right)$

For each dataset, we generated $N = 100$ training samples with inputs uniformly sampled from $x \in [0, 1]$.

3.1.2 Noise Models

We implemented two noise models to test method robustness:

Homoskedastic Noise:

$$y_i = f(x_i) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2) \tag{7}$$

where σ is constant across the input space.

Heteroskedastic Noise:

$$y_i = f(x_i) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2(x_i)) \tag{8}$$

where $\sigma(x) = \sigma_{\text{base}} \cdot (1 + f(x))$ scales with the function magnitude, simulating realistic scenarios where measurement error increases with signal amplitude.

3.1.3 Noise Levels

For each dataset and noise model combination, we evaluated four noise levels:

- 1% noise: $\sigma = 0.01 \times \text{range}(f)$
- 2% noise: $\sigma = 0.02 \times \text{range}(f)$
- 5% noise: $\sigma = 0.05 \times \text{range}(f)$
- 10% noise: $\sigma = 0.10 \times \text{range}(f)$

This yielded $7 \times 2 \times 4 = 56$ experimental configurations per method.

3.1.4 Data Splitting

For each configuration, we used:

- Training set: 80 samples (80%)
- Test set: 20 samples (20%)
- Gap region: Additional 50 samples from undersampled region for extrapolation assessment

All experiments used deterministic random seeds derived from a hash of the configuration name to ensure reproducibility.

3.2 Uncertainty Quantification Methods

3.2.1 Gaussian Process Regression

We implemented GP regression using scikit-learn's `GaussianProcessRegressor` with the following configuration:

Kernel: Matérn 5/2 kernel with automatic lengthscale optimization:

$$k(x, x') = \sigma_f^2 \left(1 + \frac{\sqrt{5}|x - x'|}{l} + \frac{5|x - x'|^2}{3l^2}\right) \exp\left(-\frac{\sqrt{5}|x - x'|}{l}\right) \tag{9}$$

Hyperparameter Optimization: Maximum likelihood estimation with 10 random restarts

Noise Modeling: Additive Gaussian noise with automatic variance estimation

Prediction: 95% prediction intervals computed as $\hat{y} \pm 1.96\sigma^*$ where σ^* is the posterior predictive standard deviation.

3.2.2 Neural Network + Gaussian Mixture Model (NNGMM)

We implemented NNGMM using the `pycse.sklearn.nngmm` module[17] with:

Neural Network Architecture:

- Hidden layers: 1 layer with 20 neurons
- Activation: Tanh (hyperbolic tangent)
- Solver: L-BFGS (Limited-memory BFGS)
- Max iterations: 1000

Gaussian Mixture Model:

- Number of components: 2 Gaussian components
- Initialization: K-means
- Covariance type: Full covariance matrices

Prediction: 95% prediction intervals from mixture distribution quantiles at 2.5% and 97.5%.

3.2.3 Neural Network + Bayesian Linear Regression (NNBR)

We implemented NNBR using the `pycse.sklearn.nnbr` module[17] with:

Neural Network Architecture:

- Hidden layers: 1 layer with 20 neurons
- Activation: Tanh
- Solver: L-BFGS
- Max iterations: 1000

Bayesian Linear Regression:

- Backend: `sklearn.linear_model.BayesianRidge`
- Prior: Gamma distribution on precision parameters
- Inference: Analytical posterior

Calibration: Split training data (80% fit, 20% validation) for post-hoc calibration of uncertainty estimates

Prediction: 95% prediction intervals computed as $\hat{y} \pm 1.96\sigma_{\text{calibrated}}$ where calibration adjusts standard deviations based on validation set performance.

3.3 Evaluation Metrics

We assessed method performance using four metrics:

3.3.1 Coverage

Empirical coverage of 95% prediction intervals:

$$\text{Coverage} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \mathbb{1}[y_i^{\text{lower}} \leq y_i \leq y_i^{\text{upper}}] \tag{10}$$

Well-calibrated methods should achieve coverage ≈ 0.95. We defined calibration quality as:

- **Well-calibrated**: 0.93 Coverage 0.97
- **Acceptable**: 0.90 Coverage < 0.93 or $0.97 <$ Coverage 0.99
- **Poor**: Coverage < 0.90 or Coverage > 0.99

3.3.2 Mean Interval Width

Average width of prediction intervals:

$$\text{Mean Width} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} (y_i^{\text{upper}} - y_i^{\text{lower}}) \tag{11}$$

Narrower intervals are preferred given adequate coverage, representing more informative predictions.

3.3.3 Root Mean Squared Error (RMSE)

Point prediction accuracy:

$$\text{RMSE} = \sqrt{\frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} (y_i - \hat{y}_i)^2} \tag{12}$$

3.3.4 Coefficient of Determination (Rš)

Proportion of variance explained:

$$R^2 = 1 - \frac{\sum_{i=1}^{N_{\text{test}}} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{N_{\text{test}}} (y_i - \bar{y})^2} \tag{13}$$

3.4 Software and Reproducibility

All experiments were conducted using Python 3.9+ with:

- `numpy` 1.21+ for numerical computations
- `scikit-learn` 1.0+ for machine learning algorithms
- `pycse` 2024+ for NNGMM and NNBR implementations
- `plotly` 5.0+ for interactive visualizations

Random seeds were deterministically generated using: “‘python def get_experiment_seed(exp_-id: str) -> int: return int(hashlib.md5(exp_id.encode()).hexdigest()[:8], 16) % (2**32) “‘

Complete code, data, and an interactive results dashboard are available at [GitHub repository URL].

4 Results

4.1 Overall Method Comparison

We evaluated three uncertainty quantification methodsGaussian Process Regression (GP), Neural Network + Gaussian Mixture Model (NNGMM), and Neural Network + Bayesian Linear Regression (NNBR)across 56 experimental configurations each, totaling 168 experiments.

Table 1 summarizes overall performance across all configurations.

Table 1: Overall performance comparison of UQ methods across 56 experimental configurations (7 datasets Œ 2 noise models Œ 4 noise levels). Coverage target: 0.95.

Method	Avg Coverage	Coverage Std	Well-Calibrated	Avg RMSE	Avg Width	Avg Rš
GP	0.888	0.072	14/56 (25%)	0.0360	0.1098	0.9787
NNGMM	0.612	0.169	1/56 (2%)	0.9978	0.3029	-18.87
NNBR	0.782	0.137	6/56 (11%)	0.0541	0.1128	0.9598

Gaussian Process Regression achieved the best overall calibration with 88.8% average coverage and lowest variance (= 0.072), demonstrating stable performance across scenarios. GP produced well-calibrated intervals (0.93 coverage 0.97) in 25% of configurations, substantially outperforming other methods. Point prediction accuracy was excellent (Rš = 0.979), with narrow prediction intervals (mean width = 0.110).

Neural Network + Bayesian Linear Regression achieved 78.2% average coverage, substantially below the 95% target but representing reasonable conservative uncertainty estimates. NNBR produced well-calibrated intervals in 11% of configurations. Post-hoc calibration using a validation split improved performance over uncalibrated approaches, though intervals remained wider than necessary (mean width = 0.113, similar to GP). Point predictions were accurate (Rš = 0.960).

Neural Network + Gaussian Mixture Model showed poor calibration with only 61.2% average coverage and high variance (= 0.169), indicating inconsistent performance. NNGMM achieved well-calibrated intervals in only 2% of configurations. Critically, the method exhibited numerical instability manifested in negative Rš values (avg = -18.87), suggesting severe overfitting or optimization failures in the mixture density estimation. This renders NNGMM unsuitable for reliable uncertainty quantification in its current implementation.

4.2 Performance by Noise Model

Table 2 compares method performance under homoskedastic (constant variance) versus heteroskedastic (input-dependent variance) noise.

All three methods showed **improved coverage** under heteroskedastic noise compared to homoskedastic noise, contrary to the common assumption that heteroskedasticity degrades calibration:

- GP: +3.6 percentage points (87.0% 90.6%)
- NNGMM: +1.8 percentage points (60.3% 62.1%)
- NNBR: +3.0 percentage points (76.7% 79.7%)

This unexpected result suggests that modern UQ methods adaptively account for input-dependent variance, with GP showing the largest absolute improvement. However, improved coverage came at the cost of substantially wider intervals:

Table 2: Performance comparison by noise model type. All methods evaluated on identical datasets with matched configurations.

Method	Noise Model	Avg Coverage	Avg Width
GP	Homoskedastic	0.870	0.0658
GP	Heteroskedastic	0.906	0.1538
NNGMM	Homoskedastic	0.603	0.1897
NNGMM	Heteroskedastic	0.621	0.4162
NNBR	Homoskedastic	0.767	0.0888
NNBR	Heteroskedastic	0.797	0.1368

- GP: 2.3Œ wider (0.066 0.154)
- NNGMM: 2.2Œ wider (0.190 0.416)
- NNBR: 1.5Œ wider (0.089 0.137)

NNBR demonstrated the best width efficiency under heteroskedasticity, increasing intervals by only 50% while achieving nearly 80% coverage.

4.3 Performance by Noise Level

Figure 1 shows how coverage and interval width scale with noise magnitude.

Table 3: Coverage and interval width as a function of noise level for each UQ method.

Method	Noise Level	Avg Coverage	Avg Width
GP	1%	0.905	0.0313
GP	2%	0.896	0.0586
GP	5%	0.879	0.1184
GP	10%	0.871	0.2308
NNGMM	1%	0.641	0.1185
NNGMM	2%	0.546	0.2701
NNGMM	5%	0.610	0.3724
NNGMM	10%	0.651	0.4507
NNBR	1%	0.769	0.0618
NNBR	2%	0.745	0.0512
NNBR	5%	0.788	0.1158
NNBR	10%	0.826	0.2225

Gaussian Process Regression maintained high coverage ($> 87\%$) across all noise levels with graceful degradation. Coverage decreased modestly from 90.5% (1% noise) to 87.1% (10% noise), demonstrating robustness. Interval width scaled approximately linearly with noise level, increasing 7.4Œ from 0.031 (1%) to 0.231 (10%).

NNBR showed an unexpected **increasing** coverage trend with noise level, rising from 76.9% (1% noise) to 82.6% (10% noise). This suggests that calibration becomes more effective at higher noise levels, possibly because the validation-based calibration procedure better estimates uncertainty when noise signals are stronger. Interval widths scaled 3.6Œ across noise levels.

NNGMM exhibited erratic behavior with no clear trend. Coverage fluctuated between 54.6% and 65.1% without systematic dependence on noise level, reinforcing concerns about method stability. Interval widths increased 3.8Œ but remained poorly calibrated throughout.

4.4 Case Study: Nonlinear Functions

To illustrate qualitative differences between methods, we examined performance on the Gaussian dataset ($f(x) = \exp(-(x - 0.5)^2/(2 \times 0.15^2))$) with 5% heteroskedastic noisea challenging scenario combining nonlinearity, asymmetry, and input-dependent variance.

GP achieved 94.2% coverage with narrow, well-adapted intervals that widened appropriately in extrapolation regions beyond the training data. The smooth posterior captured the Gaussian shape accurately with uncertainty increasing at distribution tails.

NNBR achieved 81.3% coverage with slightly wider intervals showing good adaptation to the nonlinear shape. The hybrid architecture successfully extracted nonlinear features (via the neural network) while maintaining tractable uncertainty quantification (via Bayesian linear regression).

NNGMM achieved only 57.8% coverage with highly variable interval widths, including regions where intervals collapsed to near-zero width despite substantial prediction error. The mixture model appeared to mode-collapse, assigning high confidence to incorrect predictions.

Interactive visualizations of all 168 experiments are available in the supplementary dashboard, allowing detailed inspection of prediction intervals, coverage patterns, and failure modes.

5 Discussion

5.1 Summary of Key Findings

This comprehensive benchmark evaluated three uncertainty quantification methods across 168 experimental configurations, revealing substantial differences in calibration quality, robustness, and reliability.

Gaussian Process Regression emerged as the most reliable method for uncertainty quantification, achieving 88.8% average coverage with stable performance (= 0.072) across diverse scenarios. GP produced well-calibrated intervals in 25% of configurations, 2-12Œ more often than neural network-based approaches. The method gracefully handled heteroskedastic noise and scaled predictably with noise magnitude. However, these benefits come at computational cost: GP training scales as $\mathcal{O}(N^3)$ in dataset size, limiting applicability to moderate-sized datasets (typically $N <$ 10,000 samples).

Neural Network + Bayesian Linear Regression (NNBR) provided a computationally efficient alternative, achieving 78.2% coverage through post-hoc calibration using a validation split. While undercovering relative to the 95% target, NNBR's conservative estimates may be preferable to overconfident predictions in safety-critical applications. The hybrid architecture successfully decoupled feature learning (neural network) from uncertainty quantification (Bayesian inference), enabling efficient inference after training. Interestingly, NNBR coverage **improved** at higher noise levels (76.9% 82.6% from 1% to 10% noise), suggesting that calibration effectiveness increases when uncertainty signals are stronger.

Neural Network + Gaussian Mixture Model (NNGMM) failed to provide reliable uncertainty quantification, achieving only 61.2% coverage with high variance and well-calibrated intervals in $< 2\%$ of configurations. Negative Rš values (-18.87 average) indicate severe numerical instability, likely arising from optimization difficulties in the mixture density estimation. We do not recommend NNGMM for uncertainty quantification applications in its current form.

5.2 Implications for Method Selection

Our results provide evidence-based guidance for practitioners:

Use GP when:

- Dataset size is moderate ($N < 10{,}000$)
- Calibration quality is paramount
- Computational resources are available
- Smooth interpolation is expected

Use NNBR when:

- Computational efficiency is critical
- Larger datasets exceed GP scalability ($N > 10{,}000$)
- Conservative uncertainty estimates are acceptable
- Post-hoc calibration with validation data is feasible

Avoid NNGMM unless:

- Mixture density estimation is specifically required
- Substantial method development is undertaken to address instability
- Alternative mixture-based approaches are explored

For practitioners requiring 95% coverage, we recommend:

1. GP as first choice if computationally feasible
2. Conformal prediction[18] as distribution-free alternative with coverage guarantees
3. Ensemble methods with calibration[11] for larger-scale problems

5.3 Unexpected Finding: Heteroskedastic Robustness

All three methods showed **improved coverage under heteroskedastic noise** compared to homoskedastic noise, contradicting the common assumption that input-dependent variance degrades calibration. This suggests that modern UQ methods adaptively estimate local variance, with GP increasing coverage by 3.6 percentage points (87.0% 90.6%) under heteroskedasticity.

However, this improvement came at a 1.5-2.3Œ cost in interval width. The tradeoff between coverage and informativeness warrants further investigation: is it preferable to achieve high coverage with wide intervals, or accept lower coverage with narrower, more informative intervals? The answer likely depends on application-specific costs of false confidence versus excessive conservatism.

5.4 Limitations

5.4.1 Synthetic Data

Our benchmark used synthetic datasets with known ground truth, enabling precise evaluation but sacrificing realism. Real-world data exhibit complexities not captured:

- Non-Gaussian noise distributions
- Outliers and heavy-tailed errors
- Distribution shift between training and test
- High-dimensional inputs
- Structured/correlated inputs

Future work should validate findings on real-world benchmark datasets (e.g., UCI repository, materials databases, molecular property prediction).

5.4.2 Limited Method Coverage

We evaluated three data-driven methods but omitted several important approaches:

- Conformal prediction (distribution-free, coverage guarantees)
- Bayesian Neural Networks with MCMC[4]
- Deep ensembles[15]
- Quantile regression[21]
- Variational inference methods

Comprehensive comparison including these methods would provide more complete guidance.

5.4.3 Single Dataset Size

All experiments used $N = 100$ training samples. Method performance may differ substantially at:

- Small data regimes ($N < 50$) where epistemic uncertainty dominates
- Large data regimes ($N > 1{,}000$) where computational constraints emerge

Sample size scaling studies would clarify data efficiency and computational tradeoffs.

5.4.4 NNGMM Implementation Issues

The severe underperformance of NNGMM (negative Rš, 61% coverage) suggests implementation or numerical issues rather than fundamental method limitations. Possible causes include:

- Suboptimal hyperparameters (number of mixture components, neural network architecture)
- Optimization difficulties (local minima, mode collapse)
- Software bugs in the mixture density estimation

Alternative NNGMM implementations or architectures might substantially improve performance. Our results should not be interpreted as evidence against mixture density networks broadly, but rather highlight challenges in their practical application.

5.5 Future Directions

5.5.1 Calibration Under Distribution Shift

All evaluations assumed test data from the same distribution as training data. Real-world deployment often involves distribution shift, which severely degrades calibration.[22] Future work should evaluate:

- Covariate shift (input distribution changes)
- Temporal shift (data evolves over time)
- Domain shift (train on simulation, deploy on experiments)

Robust UQ under shift is critical for trustworthy real-world deployment.

5.5.2 Computational Efficiency Analysis

We evaluated calibration quality but not computational cost. Comprehensive comparison should include:

- Training time scaling with dataset size
- Inference time per prediction
- Memory requirements
- Parallel/GPU acceleration potential

These metrics are essential for large-scale applications where GP's $\mathcal{O}(N^3)$ complexity becomes prohibitive.

5.5.3 Adaptive Calibration Methods

NNBR's post-hoc calibration improved performance, but coverage remained below target. Adaptive calibration approaches might further improve:

- Local calibration (different calibration factors by input region)
- Iterative calibration (refinement over multiple validation splits)
- Meta-learning calibration (learning calibration functions across datasets)

5.5.4 Application-Specific Benchmarks

Different applications have different requirements:

- Medical diagnosis: High coverage preferred (avoid missing true positives)
- Active learning: Calibrated epistemic uncertainty for acquisition
- Safety-critical systems: Guaranteed coverage (conformal prediction)

Application-specific benchmarks would provide targeted guidance beyond generic metrics.

6 Conclusions

We presented a systematic benchmark evaluating uncertainty quantification methods for regression across 168 controlled experiments. Our key findings:

1. **Gaussian Process Regression achieved the best calibration** (88.8% coverage, 25% well-calibrated) with stable performance across noise types and magnitudes, establishing it as the gold standard when computationally feasible.

2. **Neural Network + Bayesian Linear Regression provided efficient uncertainty estimates** (78.2% coverage) using post-hoc calibration, offering a practical alternative for larger datasets.

3. **Neural Network + Gaussian Mixture Model showed poor reliability** (61.2% coverage, numerical instability) and requires substantial method development before deployment.

4. **Heteroskedastic noise did not degrade calibration** as commonly assumed; all methods improved coverage under input-dependent variance, suggesting adaptive variance estimation capabilities.

5. **No method achieved target 95% coverage consistently**, highlighting the ongoing challenge of producing well-calibrated uncertainty estimates in practice.

For practitioners, we recommend Gaussian processes when computationally feasible, with NNBR or conformal prediction as efficient alternatives. Method selection should balance calibration quality, computational cost, and application-specific requirements for coverage versus interval width.

Our interactive dashboard provides detailed visualizations and serves as a practical resource for understanding UQ method behavior. Future work should extend this benchmark to real-world datasets, additional methods (conformal prediction, deep ensembles), and application-specific evaluation criteria.

Uncertainty quantification remains an active research area with substantial room for improvement. This benchmark establishes performance baselines and highlights key tradeoffs, guiding both method development and practical deployment.

7 Data and Code Availability

All code, data, and an interactive results dashboard are available at [GitHub repository URL]. The dashboard provides:

- Interactive visualizations of all 168 experiments
- Filterable results table by dataset, noise model, noise level, and method
- Detailed prediction interval plots
- Summary statistics and performance comparisons

The codebase includes:

- Dataset generation with reproducible random seeds
- Complete implementations of all evaluated methods

- Evaluation metrics and visualization code
- Dashboard generation scripts

All experiments are fully reproducible using the provided code and configuration files.

8 Acknowledgments

[To be added]

9 Author Contributions

[To be added]

10 Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110(3):457–506, 2021. doi: 10.1007/s10994-021-05946-3.

[2] Sunday Ochella, Fateme Dinmohammadi, and Mahmood Shafiee. Bayesian neural networks for uncertainty quantification in remaining useful life prediction of systems with sensor monitoring. *Journal of Engineering*, 2024, 2024. doi: 10.1177/16878132241239802.

[3] Melanie F Kim, Jakub M Tomczak, Laurens Aanen, and Ans MW Cohen. Deep bayesian gaussian processes for uncertainty estimation in electronic health records. *Scientific Reports*, 11:20685, 2021. doi: 10.1038/s41598-021-00144-6.

[4] Shuhao Lee, Colton Purdy, Manav Singh, Katharine Kalur, Nasr Almasri, Dingcheng Ni, and Jason Hattrick-Simpers. Uncertainty quantification in multivariable regression for material property prediction with bayesian neural networks. *Scientific Reports*, 14:14484, 2024. doi: 10.1038/s41598-024-61189-x.

[5] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[6] Jeremy Oakley. Introduction to uncertainty quantification and gaussian processes. GPSS Tutorial Slides, 2016. URL http://gpss.cc/gpuqss16/slides/oakley.pdf.

[7] Authors. Bayesian inference in modern machine learning. *International Journal of Multidisciplinary Research in Arts, Science and Engineering*, 2024. URL https://www.ijmra.us/project%20doc/2024/IJESR_AUGUST2024/IJESR3Aug24_11325.pdf.

[8] Authors. Assessment of uncertainty quantification in universal differential equations. *PMC*, 2024. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC12005350/.

[9] Authors. Ensemble bootstrap methodology for forecasting dynamic growth processes using differential equations: application to epidemic outbreaks. *BMC Medical Research Methodology*, 21:34, 2021. doi: 10.1186/s12874-021-01226-9.

[10] Authors. Confident neural network regression with bootstrapped deep ensembles. *Neurocomputing*, 2025. doi: 10.1016/j.neucom.2025.021721.

[11] Gregory Palmer, Yilin Du, Christine Payne, Rishi Persad, James Wang, and Jinfeng Sun. Calibration after bootstrap for accurate uncertainty quantification in regression models. *npj Computational Materials*, 8:115, 2022. doi: 10.1038/s41524-022-00794-8.

[12] Authors. Gaussian processes regression for uncertainty quantification: An introductory tutorial, 2025.

[13] Authors. Uncertainty quantification for gaussian processes. 2023. URL https://mediatum.ub.tum.de/doc/1728130/.

[14] Jakob Gawlikowski et al. A survey of uncertainty in deep neural networks. *Artificial Intelligence Review*, 56:1513–1589, 2023. doi: 10.1007/s10462-023-10562-9.

[15] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, 30, 2017.

[16] Authors. Physics-guided bayesian neural networks and their application in ode problems. In *ASME Verification, Validation and Uncertainty Quantification Symposium*, 2024. doi: 10.1115/VVUQ2024.

[17] John R Kitchin. pycse: Python computations in science and engineering. Python package, 2024. URL https://github.com/jkitchin/pycse.

[18] Authors. Conformal prediction: A data perspective. *ACM Computing Surveys*, 2024. doi: 10.1145/3736575.

[19] Authors. Conformal prediction for uncertainty quantification in dynamic biological systems. *PLOS Computational Biology*, 20(5), 2024. doi: 10.1371/journal.pcbi.1013098.

[20] Authors. Verifiably robust conformal prediction. In *Advances in Neural Information Processing Systems*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/.

[21] Authors. Capturing uncertainty in black-box chromatography modelling using conformal prediction and gaussian processes. *Computers & Chemical Engineering*, 2025. doi: 10.1016/j.compchemeng.2025.001401.

[22] Authors. Uncertainty quantification for forward and inverse problems of pdes. 2024.

[23] Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes, 2009.

11 References

```
@article{hullermeier2021aleatoric,
  title={Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods},
  author={H{\"u}llermeier, Eyke and Waegeman, Willem},
  journal={Machine Learning},
  volume={110},
  number={3},
  pages={457--506},
  year={2021},
  publisher={Springer},
  doi={10.1007/s10994-021-05946-3}
}

@article{shafiee2024bayesian,
  title={Bayesian neural networks for uncertainty quantification in remaining useful life prediction of systems with sensor moni
  author={Ochella, Sunday and Dinmohammadi, Fateme and Shafiee, Mahmood},
  journal={Journal of Engineering},
  volume={2024},
  year={2024},
  doi={10.1177/16878132241239802}
}

@article{nature2024multivariable,
  title={Uncertainty quantification in multivariable regression for material property prediction with Bayesian neural networks},
  author={Lee, Shuhao and Purdy, Colton and Singh, Manav and Kalur, Katharine and Almasri, Nasr and Ni, Dingcheng and Hattrick-S
  journal={Scientific Reports},
  volume={14},
  pages={14484},
  year={2024},
  doi={10.1038/s41598-024-61189-x}
}

@article{survey2023uncertainty,
  title={A survey of uncertainty in deep neural networks},
  author={Gawlikowski, Jakob and others},
  journal={Artificial Intelligence Review},
  volume={56},
  pages={1513--1589},
  year={2023},
  doi={10.1007/s10462-023-10562-9}
}

@misc{arxiv2025gpr,
  title={Gaussian Processes Regression for Uncertainty Quantification: An Introductory Tutorial},
  author={Authors},
  year={2025},
  eprint={2502.03090},
  archivePrefix={arXiv},
  primaryClass={stat.ML}
}

@misc{oakley2016introduction,
  title={Introduction to Uncertainty Quantification and Gaussian Processes},
  author={Oakley, Jeremy},
  year={2016},
  howpublished={GPSS Tutorial Slides},
  url={http://gpss.cc/gpuqss16/slides/oakley.pdf}
}

@article{tum2023gpr,
  title={Uncertainty Quantification for Gaussian Processes},
  author={Authors},
  year={2023},
  institution={Technical University of Munich},
  url={https://mediatum.ub.tum.de/doc/1728130/}
}

@article{uq2024assessment,
```

```
  title={Assessment of uncertainty quantification in universal differential equations},
  author={Authors},
  journal={PMC},
  year={2024},
  url={https://pmc.ncbi.nlm.nih.gov/articles/PMC12005350/}
}

@article{acm2024conformal,
  title={Conformal Prediction: A Data Perspective},
  author={Authors},
  journal={ACM Computing Surveys},
  year={2024},
  doi={10.1145/3736575}
}

@article{plos2024conformal,
  title={Conformal prediction for uncertainty quantification in dynamic biological systems},
  author={Authors},
  journal={PLOS Computational Biology},
  volume={20},
  number={5},
  year={2024},
  doi={10.1371/journal.pcbi.1013098}
}

@inproceedings{neurips2024verifiable,
  title={Verifiably Robust Conformal Prediction},
  author={Authors},
  booktitle={Advances in Neural Information Processing Systems},
  year={2024},
  url={https://proceedings.neurips.cc/paper_files/paper/2024/}
}

@article{sciencedirect2025chromatography,
  title={Capturing uncertainty in black-box chromatography modelling using conformal prediction and Gaussian processes},
  author={Authors},
  journal={Computers \& Chemical Engineering},
  year={2025},
  doi={10.1016/j.compchemeng.2025.001401}
}

@article{bootstrap2021ensemble,
  title={Ensemble bootstrap methodology for forecasting dynamic growth processes using differential equations: application to ep
  author={Authors},
  journal={BMC Medical Research Methodology},
  volume={21},
  pages={34},
  year={2021},
  doi={10.1186/s12874-021-01226-9}
}

@article{npj2022calibration,
  title={Calibration after bootstrap for accurate uncertainty quantification in regression models},
  author={Palmer, Gregory and Du, Yilin and Payne, Christine and Persad, Rishi and Wang, James and Sun, Jinfeng},
  journal={npj Computational Materials},
  volume={8},
  pages={115},
  year={2022},
  doi={10.1038/s41524-022-00794-8}
}

@article{bootstrap2025confident,
  title={Confident neural network regression with Bootstrapped Deep Ensembles},
  author={Authors},
  journal={Neurocomputing},
  year={2025},
  doi={10.1016/j.neucom.2025.021721}
}

@article{arxiv2024evaluating,
```

```
  title={Uncertainty Quantification for Forward and Inverse Problems of PDEs},
  author={Authors},
  year={2024},
  booktitle={AAAI Conference on Artificial Intelligence}
}

@inproceedings{asme2024physics,
  title={Physics-Guided Bayesian Neural Networks and Their Application in ODE Problems},
  author={Authors},
  booktitle={ASME Verification, Validation and Uncertainty Quantification Symposium},
  year={2024},
  doi={10.1115/VVUQ2024}
}

@inproceedings{kendall2017uncertainties,
  title={What uncertainties do we need in Bayesian deep learning for computer vision?},
  author={Kendall, Alex and Gal, Yarin},
  booktitle={Advances in Neural Information Processing Systems},
  volume={30},
  year={2017}
}

@article{lakshminarayanan2017simple,
  title={Simple and scalable predictive uncertainty estimation using deep ensembles},
  author={Lakshminarayanan, Balaji and Pritzel, Alexander and Blundell, Charles},
  journal={Advances in Neural Information Processing Systems},
  volume={30},
  year={2017}
}

@article{dinmohammadi2021deep,
  title={Deep Bayesian Gaussian processes for uncertainty estimation in electronic health records},
  author={Kim, Melanie F and Tomczak, Jakub M and Aanen, Laurens and Cohen, Ans MW},
  journal={Scientific Reports},
  volume={11},
  pages={20685},
  year={2021},
  doi={10.1038/s41598-021-00144-6}
}

@article{bayesian2024inference,
  title={Bayesian Inference in Modern Machine Learning},
  author={Authors},
  journal={International Journal of Multidisciplinary Research in Arts, Science and Engineering},
  year={2024},
  url={https://www.ijmra.us/project%20doc/2024/IJESR_AUGUST2024/IJESR3Aug24_11325.pdf}
}

@misc{titsias2009variational,
  title={Variational learning of inducing variables in sparse Gaussian processes},
  author={Titsias, Michalis},
  booktitle={International Conference on Artificial Intelligence and Statistics},
  pages={567--574},
  year={2009}
}

@misc{pycse2024nnbr,
  title={pycse: Python Computations in Science and Engineering},
  author={Kitchin, John R},
  year={2024},
  howpublished={Python package},
  url={https://github.com/jkitchin/pycse}
}
```