Computer Modeling in Engineering & Sciences

Tech Science Press

https://doi.org/10.32604/cmes.2026.080808

ARTICLE

# A Stochastic Ensemble Physics-Informed Neural Networks via Bagging and Monte Carlo Dropout

**Thao Nguyen-Trang**[1,2,*] **and Hiep Ha-Hoang**[3]

[1]Laboratory for Artificial Intelligence, Institute for Computational Science and Artificial Intelligence, Van Lang University, Ho Chi Minh City, Vietnam
[2]Faculty of Information Technology, Van Lang School of Technology, Van Lang University, Ho Chi Minh City, Vietnam
[3]Department of Computer Science, University of York, York, UK

*Corresponding Author: Thao Nguyen-Trang. Email: thao.nguyentrang@vlu.edu.vn

Received: 15 February 2026; Accepted: 27 April 2026; Published: 27 May 2026

**ABSTRACT:** Solving differential equations (DEs), including ordinary differential equations (ODEs) and partial differential equations (PDEs), is fundamental to scientific computing and engineering. The development of deep learning has led to Physics-Informed Neural Networks (PINNs), in which physical laws are embedded directly into the loss function. However, PINNs inherit the intrinsic instability of deep neural networks (DNNs) and lack an effective mechanism for Uncertainty Quantification (UQ). This paper proposes a stochastic ensemble framework to address these limitations. The proposed method is a double-stochastic ensemble framework that combines bagging (via bootstrap resampling and randomized collocation points) with Monte Carlo dropout (MCDO) applied at inference time to provide consistent UQ. This double-stochastic design yields two main contributions. First, it reduces the intrinsic variance of the predicted solution and improves accuracy, particularly in small-data and/or noisy-data regimes. Second, the variance of the ensemble outputs serves as a reliable and computationally feasible proxy for the solution uncertainty. Numerical experiments on ODEs and nonlinear PDEs (e.g., Burgers' equation) under impulse noise demonstrate that the proposed method outperforms standard baselines in predictive accuracy. Furthermore, the obtained Prediction Interval Coverage Probability (PICP) and Mean Prediction Interval Width (MPIW) confirm that our framework yields well-calibrated prediction intervals and effectively mitigates the severe performance degradation observed in single PINNs.

**KEYWORDS:** Physics-informed neural networks; bagging; Monte Carlo dropout; differential equations; uncertainty quantification; machine learning

## 1 Introduction

Mathematical models governing complex physical phenomena, ranging from wave propagation in fluid mechanics to simpler time-evolving systems, are commonly formulated in terms of differential equations (DEs) [1]. DEs are typically categorized into ordinary differential equations (ODEs) and partial differential equations (PDEs). Developing robust and accurate solvers for both ODEs and PDEs remains a cornerstone of scientific computing and engineering.

Historically, these DEs have been solved using classical numerical methods such as the finite element method (FEM) and the finite difference method (FDM) [2–5]. Despite their maturity, these approaches exhibit notable limitations, particularly when addressing the curse of dimensionality in high-dimensional

problems, which can lead to exponential growth in computational cost [6,7]. Moreover, mesh generation for complex domains or time dependent boundary conditions remains a major practical challenge [8].

The rise of deep learning has enabled an alternative paradigm: physics-informed neural networks (PINNs). PINNs represent a principled integration of deductive reasoning from physical laws and inductive learning from data [6,9–12]. In PINNs, the approximate solution $u(\mathbf{x}, t)$ is represented by a deep neural network (DNN) [13]. The key idea is to embed the governing physical laws, expressed as ODEs or PDEs, directly into the neural network loss function, thereby enforcing consistency with both available observations and the underlying physics [14]. This design is particularly advantageous in small data regimes, as the physics constraints act as a strong regularizer, improving robustness and generalization relative to purely data-driven models.

Nevertheless, because PINNs are built upon DNNs, they inherit a major challenge from deep learning: model instability. Neural networks are often regarded as "unstable procedures," meaning that small variations in weight initialization, the selection of collocation points ($N_f$), or fluctuations in the training data (e.g., initial/boundary conditions) can yield substantially different predicted solutions [15]. This lack of robustness, especially pronounced in small-data settings, is a critical limitation that must be addressed if PINNs are to become reliable tools for engineering applications.

In addition, measurement noise is inevitable in physical systems [9]. Estimating high-order derivatives (required to evaluate PDE residuals) from noisy data can substantially amplify errors in either numerical differentiation or automatic differentiation [16]. The numerical results reported in the aforementioned studies clearly illustrate this brittleness. A single PINN faces a coupled optimization challenge: it must simultaneously reduce data mismatch (under noisy observations) and the physical residual error. When observations are corrupted by noise, the optimizer may produce physically inconsistent solutions elsewhere in the domain in an attempt to fit noisy outliers.

Beyond these issues, standard PINNs typically provide only point estimates of the solution and lack an integrated mechanism for uncertainty quantification (UQ). The need for UQ has been recognized as an open research question since the foundational PINN literature. While probabilistic frameworks such as Bayesian PINNs (B-PINNs) have been proposed to handle noisy data and to infer parameter distributions, they often require more sophisticated computational pipelines, and systematic specification of priors remains a nontrivial open problem [17]. Conversely, applying simpler UQ techniques such as Monte Carlo dropout (MCDO) to a single PINN often yields overly conservative (overestimated) uncertainty intervals that provide limited practical insight into predictive reliability.

To improve the robustness and performance of PINNs, numerous research directions have been explored, including global optimization strategies, adaptive loss balancing ($\lambda$), adaptive activation functions, and domain decomposition to address large-scale problems [4,14,17–19]. However, these methods primarily target scalability and gradient-based optimization challenges.

Ensemble learning has also been investigated in the PINN context. For example, reference [9] proposed ensemble PINNs (ePINNs), which adopt a construction strategy by embedding multiple sub-networks in parallel within a main network. In this approach, the randomization dimensions arise entirely from the intentional initialization of specific patterns, allowing the sub-networks to specialize in learning these patterns. This implies that both the data and the collocation points are kept fixed throughout the study. Moreover, UQ has not been provided in this work yet. Similarly, Haitsiukevich and Ilin [15] used the median of individual neural networks as the final prediction. A distinctive aspect of this method is the expansion of collocation points based on the agreement among ensemble members. In this study, randomness comes only from the random initialization of neural network weights. In addition, the prediction variance is used merely

as a threshold to guide the expansion of collocation points, rather than to provide a true prediction interval for the solution. It can be observed that existing ensemble PINN methods often rely on base learners that are either pre-trained neural networks or networks with randomly initialized weights. This reduces the diversity of the ensemble, as the models tend to converge to a limited number of local minima during training. In contrast, randomizing observations and collocation points provides genuinely different perspectives for the PINNs to learn from. Furthermore, when observations contain noise, random resampling can significantly reduce the effect of such noise. Finally, the provision of reliable UQ has not been adequately addressed in existing studies.

Therefore, a key research gap lies in the lack of a method that exploits classical statistical principles (bagging with bootstrap data and randomized collocation points, together with unbiased MCDO) to simultaneously mitigate intrinsic instability and provide reliable UQ, particularly for PDE problems in small-data or noisy-data regimes where vanilla PINNs remain challenging.

To fill this gap, this study proposes *A Stochastic Ensemble Physics-Informed Neural Networks via Bagging and MCDO* (Bagging-PINNs). The framework is designed as a double-stochastic ensemble that leverages the variance-reduction capability of bagging and the UQ capability of MCDO, with the aim of systematically addressing instability and improving the reliability of UQ in PINNs.

This simple but novel combination (bagging during training and MCDO during inference) reduces intrinsic variance and provides more reliable UQ than many more complex alternatives, while incurring minimal additional computational cost at inference. Moreover, our numerical studies indicate that the original PINN formulation, often referred to as vanilla PINNs, can perform worse than a purely data-driven DNN under noisy observations, whereas Bagging-PINNs preserve the benefits of physics constraints without becoming brittle in noisy data settings. While a single PINN may struggle to jointly optimize noisy data loss and physical residual loss, leading to physically inconsistent solutions, bagging alleviates this issue by averaging predictions across multiple submodels; the adverse effect of outliers present in one bootstrap sample is counteracted by other samples that do not contain them, thereby reducing variance and maintaining the benefits of the physics constraints.

After this introduction, Section 2 presents background on DNNs and PINNs. Section 3 describes the Bagging-PINNs architecture in detail. Section 4 reports numerical experiments and results. Finally, Section 5 summarizes the findings and discusses future research directions.

## 2 Preliminary Issues

### *2.1 Artificial Neural Networks*

Artificial Neural Networks (ANNs) are general function approximators that map inputs $\mathbf{x} \in \mathbb{R}^d$ to outputs $\mathbf{y} \in \mathbb{R}^m$ through stacked linear transformations and nonlinear activations. A feed-forward network with $L$ layers can be written as

$$u_\theta(\mathbf{x}) = \sigma\Big(W_L\,\sigma(\cdots\sigma(W_1\mathbf{x}+b_1)\cdots)+b_L\Big), \tag{1}$$

where $\theta$ denotes the collection of weights $W_l$ and biases $b_l$, and $\sigma$ (e.g., tanh) is typically chosen to be smooth for differential-equation settings. Training minimizes the empirical risk

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\|u_\theta(\mathbf{x}_i)-\tilde{u}(\mathbf{x}_i)\|^2 \tag{2}$$

using stochastic gradient descent (SGD) and its variants. The universal approximation theorem guarantees that $\|u_\theta - u^*\|_\infty \to 0$ as the network capacity increases. Nevertheless, DNNs often struggle in sparse or noisy data regimes, resulting in high variance and poor generalization.

### *2.2 Physics-Informed Neural Networks*

Physics-informed neural networks build upon DNNs by embedding physical constraints, enabling semi-supervised learning that does not rely solely on labeled data. Specifically, they incorporate the residual of a governing equation $\mathcal{N}[u(\mathbf{x})] = f(\mathbf{x})$ over a domain $\Omega \subset \mathbb{R}^d$ (e.g., $\mathcal{N} = \partial_t - \alpha\nabla^2$ for a parabolic PDE) and boundary/initial constraints $\mathcal{B}[u] = g$ on $\partial\Omega$ into the loss function. Using automatic differentiation to compute $\mathcal{N}[u_\theta]$, the loss is typically formulated as

$$\begin{aligned}\mathcal{L}(\theta) = \lambda_d \frac{1}{N_d}\sum_{i=1}^{N_d} \|u_\theta(\mathbf{x}_i^d) - u(\mathbf{x}_i^d)\|^2 \\ + \lambda_r \frac{1}{N_r}\sum_{j=1}^{N_r} \|\mathcal{N}[u_\theta(\mathbf{x}_j^r)] - f(\mathbf{x}_j^r)\|^2 \\ + \lambda_b \frac{1}{N_b}\sum_{k=1}^{N_b} \|\mathcal{B}[u_\theta(\mathbf{x}_k^b)] - g(\mathbf{x}_k^b)\|^2,\end{aligned} \tag{3}$$

where the sampling sets $\mathbf{x}^d$, $\mathbf{x}^r$, and $\mathbf{x}^b$ are typically generated using Latin hypercube sampling or quasi-Monte Carlo methods, and the weights $\boldsymbol{\lambda} = (\lambda_d, \lambda_r, \lambda_b)$ can be tuned via gradient normalization.

## 3 The Proposed Method

The proposed method, referred to as Bagging-PINNs, is a double-stochastic ensemble framework designed to combine the strengths of PINNs with variance-reduction and uncertainty-quantification (UQ) techniques from ensemble learning.

### *3.1 Stochastic Ensemble Framework*

Bagging-PINNs leverage two sources of stochasticity to construct $M$ independent and diverse PINN models ($\mathbf{u}_{\theta_m}$).

#### *3.1.1 Bagging and Input Randomization*

Bagging is implemented by randomizing both the training data and the physics sampling points for each submodel, encouraging different members to converge to different solutions within the large hypothesis space of neural networks:

- **Bootstrap sampling:** The $m$-th PINN is trained on a bootstrap dataset ($\mathcal{Z}^{(m)}$) obtained by sampling with replacement from the original training set.
- **Collocation-point randomization:** Instead of bootstrapping, each submodel uses a newly and independently generated set of collocation points ($X_r^{(m)}$) drawn from a uniform distribution over the spatiotemporal domain. To ensure a fair comparison, the number of collocation points for each submodel strictly matches that of the single PINN baseline.
- Furthermore, to ensure ensemble diversity, each base learner in the Bagging architecture is independently initialized using Kaiming Uniform initialization prior to bootstrap training.

#### *3.1.2 Monte Carlo Dropout*

In this work, MCDO is applied at inference (test) time to provide consistent UQ for every model. Specifically, for each sample in the $T$ Monte Carlo samples, the dropout rate $p$ is drawn from a continuous uniform distribution $U(0.2, 0.5)$. Dropout is applied immediately after each nonlinear activation function in all hidden layers, while the input and output layers remain fully connected. After the MCDO process, we obtain $T$ independent predictions from Bagging-PINN, which can then be aggregated to estimate the predictive uncertainty.

It is important to note that Bagging randomness and MCDO randomness do not introduce redundant uncertainty, as they operate at different stages with complementary roles. Bootstrap re-sampling is applied during the training phase to enhance ensemble diversity and reduce the intrinsic variance of the model; whereas MCDO is applied only during the test phase to quantify the predictive uncertainty of the already trained models.

### *3.2 Training Procedure and Result Aggregation*

Training consists of constructing and fitting $M$ independent PINNs, followed by aggregating their predictions to obtain both the final solution estimate and uncertainty measures:

- Train $M$ independent PINNs. Each network minimizes $\mathcal{L}(\theta)$ using its bootstrap dataset $\mathcal{Z}^{(m)}$ and randomized collocation points $\mathcal{X}_r^{(m)}$.
- Compute the Bagging-PINNs prediction $\mathbf{u}_{\text{Bagging-PINN}}(\mathbf{z})$ by averaging across submodels:

$$\mathbf{u}_{\text{Bagging-PINN}}(\mathbf{z}) = \frac{1}{M}\sum_{m=1}^{M}\mathbf{u}_{\theta_m}(\mathbf{z}). \tag{4}$$

The averaging operation in Eq. (4) is the key mechanism that reduces the instability of individual PINN models. The improvement is rooted in the statistical principle of variance reduction for unstable predictors, as introduced by [20]. In essence, neural networks are inherently unstable, and this issue becomes more severe in PINNs due to the highly non-convex loss landscape, as well as gradient pathologies arising from the conflict between data fitting and physical constraints [21,22]. By applying Bagging, the averaging mechanism ensures that random deviations and locally perturbed gradients from individual base models are effectively canceled out. From a probabilistic perspective, the Central Limit Theorem implies that the ensemble average of $M$ independent trials converges to a normal distribution, where the prediction standard deviation is reduced by a factor of $\sqrt{M}$. This significantly reduces the overall variance of the model without increasing the systematic error (bias), thereby stabilizing the optimization process of the PINN system.

Mathematically, the proposed framework can minimize the global error. According to Breiman's error decomposition,

$$MSE_{ens} = \overline{MSE}_{base} - Var_{base},$$

where $MSE_{ens}$ denotes the mean square error of the ensemble, $\overline{MSE}_{base}$ represents the average mean square error of the base models, and $Var_{base}$ captures the variance among the base models' predictions. In conventional ensemble learning, inducing model diversity typically necessitates a strict trade-off: artificially weakening individual models to force varied perspectives, thereby inflating the $\overline{MSE}_{base}$ term. However, our double-stochastic architecture inherently bypasses this limitation, yielding a dual-advantage mechanism.

Regarding $\overline{MSE}_{base}$, our algorithm keeps the number of collocation points, $N_f$, the same for each sub-model, as detailed in Section 3.1.1. Consequently, the base learners are neither structurally weakened nor underparameterized when enforcing the underlying physical constraints. Especially, under impulse noise conditions, the bootstrap re-sampling mechanism systematically excludes extreme outliers from the training subsets of most base models. As a result, these base PINNs optimize over a significantly 'cleaner' data manifold while retaining 100% of their physical regularization capacity. This mechanism guarantees that $\overline{MSE}_{base}$ remains tightly bounded, often achieving superior baseline accuracy compared to a standard PINN directly exposed to the full impulse noise distribution.

Regarding $Var_{base}$, the ensemble variance is approximated by

$$E[Var_{base}] \approx \left(1 - \frac{1}{M}\right) \bar{\sigma}^2 (1 - \rho),$$

where $\rho$ denotes the average prediction correlation coefficient, and $\bar{\sigma}^2$ is the mean of all individual variances $\sigma_m^2$ across the $M$ sub-models. Training multiple PINNs on an identical set of collocation points frequently traps them in identical local minima. As a result, their prediction errors are similar and highly correlated ($\rho \to$ high) and the term $(1 - \rho)$ goes to 0. Consequently, $Var_{base}$ is nullified, rendering the ensemble aggregation statistically trivial. Meanwhile, Bagging-PINN's spatial randomization of collocation points serves as a robust decorrelation mechanism. This structural perturbation dynamically alters the highly non-convex loss landscape for each sub-model, forcing the individual PINNs to traverse distinct optimization trajectories and evaluate physical residuals at disparate spatial coordinates. Consequently, predictive codependence is effectively shattered, precipitating a sharp decline in the correlation coefficient ($\rho \to$ low). Following the expectation formula, this minimization of $\rho$ directly amplifies the $(1 - \rho)$ term. This rigorously maximizes the variance component ($Var_{base}$) in Breiman's decomposition, functioning as an active noise-cancellation mechanism specifically concentrated at regions corrupted by localized artifacts.

In summary, the proposed Bagging-PINN can increase $Var_{base}$ without significantly reducing the individual accuracy of $\overline{MSE}_{base}$; hence, the proposed method can successfully overcome the limitations of traditional ensemble methods. This dual mechanism theoretically ensures a substantial reduction in the total predictive error, $MSE_{ens}$, particularly under severely noisy data conditions.

- Perform UQ via test-time MCDO using an average-then-aggregate rule: for each sample $s = 1, \ldots, T$, enable dropout in all sub-models, compute their predictions, and then average over $M$ to obtain $\tilde{u}^{(s)}(z)$. After $T$ samples,

$$\mu(z) = \frac{1}{T} \sum_{s=1}^{T} \tilde{u}^{(s)}(z), \quad \sigma(z) = \sqrt{\frac{1}{T} \sum_{s=1}^{T} \left(\tilde{u}^{(s)}(z) - \mu(z)\right)^2}, \tag{5}$$

and the prediction interval is reported as $\mu \pm 1.96\,\sigma$. The prediction interval $\mu \pm 1.96\,\sigma$ implicitly assumes a Gaussian distribution for the $T$ predictive samples. First, according to [23], the use of Monte Carlo Dropout can be considered an approximation of a Deep Gaussian Process. Therefore, the predictive distribution from each PINN sub-model is already approximately Gaussian. Second, the final prediction of Bagging-PINN is obtained as the average of $M$ independent base models. As a result of the Central Limit Theorem, even if the predictive distributions of individual models exhibit slight deviations, the aggregated distribution of the ensemble is guaranteed to converge asymptotically to a Gaussian distribution. The proposed algorithm is summarized in Fig. 1.

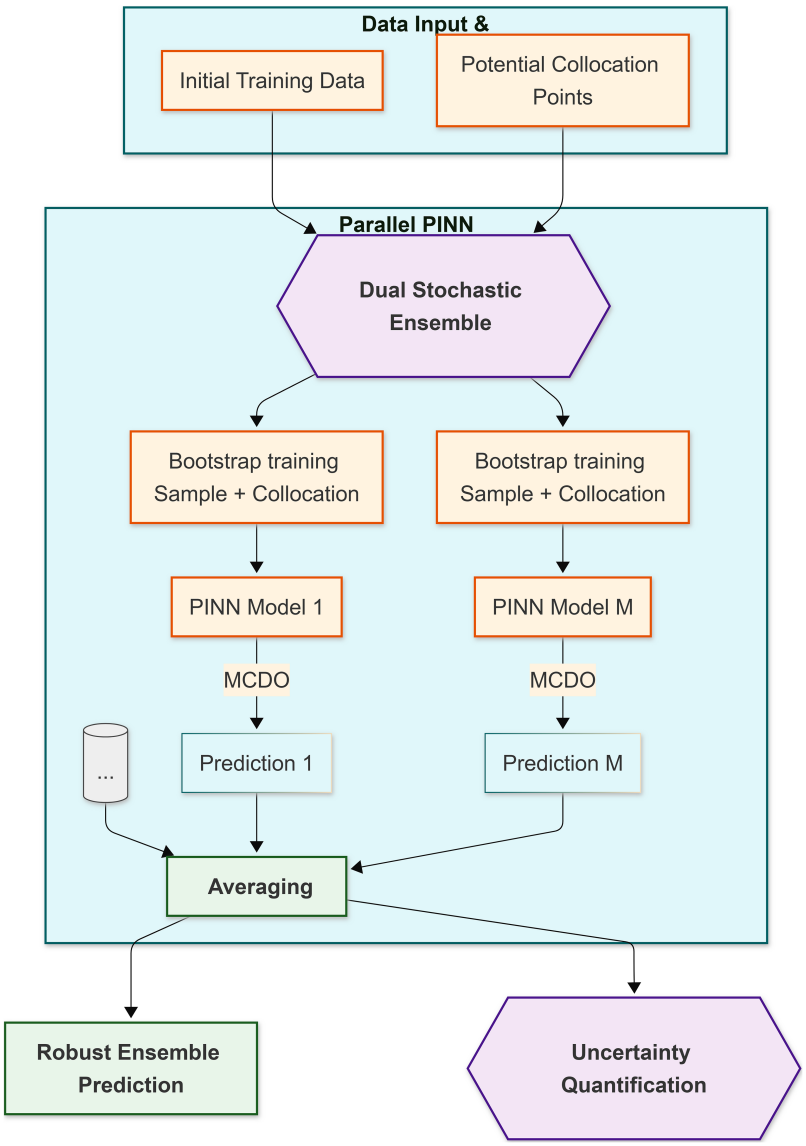


**Figure 1:** The diagram for Bagging-PINN.

## 4 Numerical Results

This section systematically analyzes and compares the performance of Bagging-PINN with several baseline models. The selected models include ANN, PINN, and a representative ensemble-PINN method that introduces randomness only through weight initialization, namely Init-Ensemble PINNs. We evaluate these methods on three benchmark problems, including an ODE, a PDE representing the wave equation, and a PDE representing the Burgers equation. All prediction errors as well as UQ metrics are computed on a fully noise-free high-resolution reference grid (e.g., a $60 \times 60$ grid, corresponding to 3600 evaluation points for PDEs). The training dataset (observations) consists of 100 points for the ODE problem, 200 points for the wave equation, and 2000 points on the boundary and initial conditions for the Burgers equation. In addition, for ensemble-based methods, the neural network weights are randomly initialized using the default Kaiming uniform distribution. For the Bagging-PINN, each ensemble member is optimized on an

independently sampled set of collocation points within the computational domain, with $N_f = 1000$ points for the wave equation and $N_f = 5000$ points for the Burgers equation.

The models are evaluated under different data conditions, including ideal noise-free data and data with impulse noise at varying noise levels and corruption rates, to provide a comprehensive view of the strengths and limitations of each approach. To simulate extreme outliers encountered in practice, the observational data (i.e., initial and boundary conditions) are corrupted using an impulse noise model. Specifically, a certain portion of the training data, determined by the Corruption Rate (*CR*), is randomly selected, and their target values are perturbed with an amplitude controlled by the Impulse Factor (*IF*). It is important to emphasize that this noise is applied only to the observational data. The collocation points, which are merely space–time coordinates $(x, t)$ used to evaluate the error between the model and the given partial differential equation, remain completely noise-free. The values of *CR* and *IF* are set across the three examples as $(CR = 0.2, IF = 2)$, $(CR = 0.05, IF = 3)$, and $(CR = 0.02, IF = 5)$, respectively. These settings are chosen to illustrate scenarios in which Bagging-PINN provides significant advantages in practical applications.

To ensure a fair comparison, all four models use the same network architectures and hyperparameters. Specifically, for the ODE problem, the network has 2 hidden layers with 64 neurons per layer. For the wave equation, the network is expanded to 3 hidden layers with 128 neurons per layer to handle the higher complexity. For the Burgers equation, a neural network with three hidden layers and 64 neurons per layer is used. The remaining hyper-parameters are summarized in Table 1. To evaluate deterministic prediction accuracy, we use standard metrics including Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). In addition, the quality of UQ is evaluated based on the Prediction Interval Coverage Probability (PICP) and the Mean Prediction Interval Width (MPIW), which are calculated based on $T = 1000$ samples for all four methods. For ensemble methods, the number of ensemble members $M$ is set to 20, 5, and 50 for the three examples, respectively, based on the sensitivity analysis presented in Section 4.4.

**Table 1:** Summary of core training hyperparameters.

| Parameter | Value |
|---|---|
| Activation function (ACT) | tanh |
| Optimization solver | Adam |
| Learning rate (LR) | $10^{-3}$ |
| Maximum epochs | 1000 |
| Patience | 20 |
| Minimum change threshold ($\Delta_{\min}$) | $10^{-4}$ |

### *4.1 Ordinary Differential Equation*

This subsection simulates the process of drug concentration decay $y(t)$ in the body over time, described by the first-order ODE $\frac{dy}{dt} = -2y$ with the initial condition $y(0) = 1$. The exact analytical solution is $y(t) = e^{-2t}$.

As shown in Table 2, under noise-free conditions, both Bagging-PINN and Init-Ensemble PINNs outperform the pure ANN and the standard PINN in terms of deterministic accuracy (as reflected by MAE and RMSE). Moreover, Bagging-PINN provides highly reliable uncertainty estimates, as indicated by an MPIW of 0.4363, which is significantly narrower and more informative than that produced by ANN (MPIW = 1.2056) and PINN (MPIW = 1.7902). Compared to Init-Ensemble PINNs, Bagging-PINN is slightly worse in deterministic error but slightly better in uncertainty estimation.

**Table 2:** Test-set error metrics for ODE under different noise settings.

| Setting | Model | MAE | MSE | RMSE | MPIW | PICP (%) |
|---|---|---|---|---|---|---|
| Noise-free | ANN | 0.0099 | 0.0002 | 0.0141 | 1.2056 | 100 |
| | PINN | 0.0017 | 0.0000 | 0.0019 | 1.7902 | 100 |
| | Bagging-PINN | 0.0016 | 0.0000 | 0.0018 | 0.4363 | 100 |
| | Init-Ensemble PINNs | 0.0014 | 0.0000 | 0.0016 | 0.4451 | 100 |
| Impulse noise $(CR = 0.2, IF = 2)$ | ANN | 0.4462 | 0.2883 | 0.5369 | 4.4445 | 100 |
| | PINN | 0.0625 | 0.0045 | 0.0670 | 1.7250 | 100 |
| | Bagging-PINN | 0.0609 | 0.0043 | 0.0656 | 0.4270 | 100 |
| | Init-Ensemble PINNs | 0.0627 | 0.0045 | 0.0672 | 0.4344 | 100 |

Under impulse noise, the ANN model results in the large error with a MAE of 0.4462. Although the standard PINN still maintains physical constraints and keeps the error at a moderate level (MAE = 0.0625), its UQ performance is unreliable due to the large MPIW of 1.725. In contrast, Bagging-PINN not only achieves the best MAE (0.0609), but also maintains a sharp prediction interval (MPIW = 0.427) while ensuring full coverage of the data (see Fig. 2). The representative method, Init-Ensemble PINNs, also shows relatively good performance; however, its MAE (0.0627) remains higher than that of Bagging-PINN.

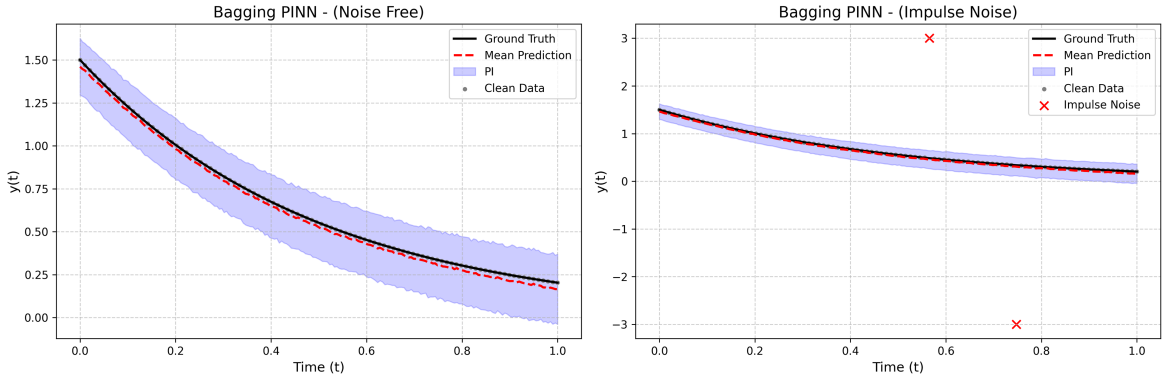


**Figure 2:** Predictions vs. the exact solution of Bagging-PINN on noise data.

### *4.2 1D Linear Wave Equation*

We consider the one-dimensional wave equation $u_{tt} - u_{xx} = 0$ (with wave speed $c = 1$). The exact solution is $u(x, t) = \sin(\pi x)\cos(\pi t)$.

Under the noise-free condition, the results in Table 3 show that Init-Ensemble PINNs and Bagging-PINN achieve competitive performance, with MPIW values of 1.2106 and 1.2241, respectively. These methods also outperform the standard PINN. The advantage of Bagging-PINN becomes most evident when dealing with impulse noise $(CR = 0.05,\ IF = 3)$. For the standard PINN, its MAE increases from 0.0260 to 0.1061, and its MPIW increases from 2.6153 to 6.3677. The Init-Ensemble PINNs method, although ensemble-based, remains sensitive to these extreme outliers, resulting in an MAE of 0.1106 and an expanded MPIW of 2.7788. In contrast, by leveraging bootstrap sampling (which naturally excludes a portion of extreme noisy samples in each base model), Bagging-PINN effectively mitigates this disturbance. It achieves the lowest MAE (0.0902), maintains a sharp prediction interval (MPIW = 1.2388), and ensures near-complete coverage (PICP = 99.97%).

**Table 3:** Test-set error metrics for wave equation under different noise settings.

| Setting | Model | MAE | MSE | RMSE | MPIW | PICP (%) |
|---|---|---|---|---|---|---|
| Noise-free | ANN | 0.0219 | 0.0007 | 0.0267 | 2.1846 | 100 |
| | PINN | 0.0260 | 0.0009 | 0.0312 | 2.6153 | 100 |
| | Bagging-PINN | 0.0224 | 0.0007 | 0.0277 | 1.2241 | 100 |
| | Init-Ensemble PINNs | 0.0195 | 0.0006 | 0.0241 | 1.2106 | 100 |
| Impulse noise ($CR = 0.05, IF = 3$) | ANN | 0.2253 | 0.1659 | 0.4073 | 23.0970 | 100 |
| | PINN | 0.1061 | 0.0163 | 0.1276 | 6.3677 | 100 |
| | Bagging-PINN | 0.0902 | 0.0126 | 0.1121 | 1.2388 | 99.97 |
| | Init-Ensemble PINNs | 0.1106 | 0.0194 | 0.1391 | 2.7788 | 99.97 |

The stability of the method is further illustrated in Fig. 3. The spatio-temporal heatmap of the prediction interval width shows that Bagging-PINN maintains a narrow and uniform uncertainty band across the entire domain, even in regions near extreme impulse noise points (marked with crosses). Moreover, the 1D spatial cross-section at $t = 0.5$ further confirms the robustness of the model: the mean prediction closely follows the ground truth, and the confidence band fully covers the true solution without distortion or unreasonable inflation under the influence of local perturbations.

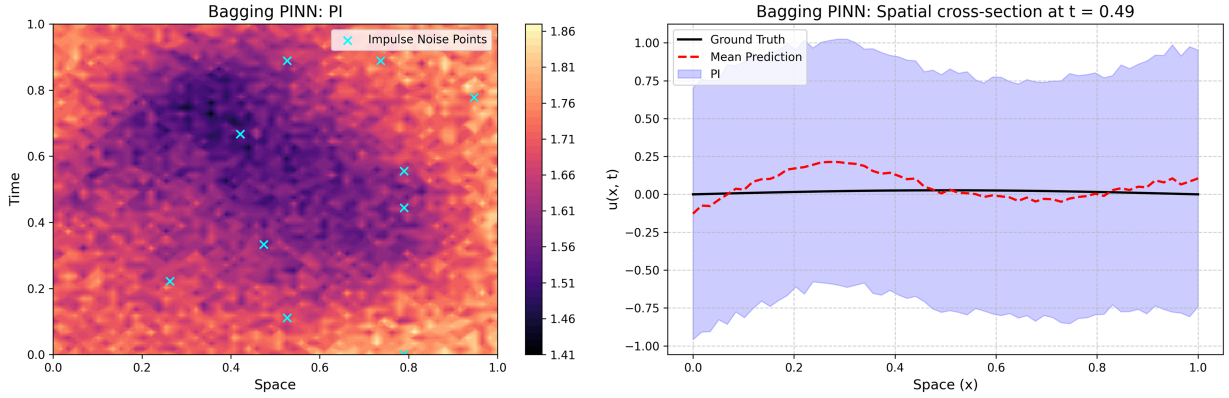


**Figure 3:** Heatmap of prediction interval width and 1D spatial cross-section of Bagging-PINN.

### *4.3 Nonlinear Burgers' Equation*

To demonstrate the generalization capability of the proposed method for complex nonlinear problems, we apply the model to the one-dimensional Burgers equation. Specifically, the problem is defined as follows.

$$u_t + uu_x - \frac{0.01}{\pi} u_{xx} = f(t,x), \quad x \in [-1,1], \ t \geq 0, \tag{6}$$

where the kinematic viscosity coefficient is set to $\nu = \frac{0.01}{\pi}$, and the source term $f(t,x)$ is analytically constructed such that the exact solution is $u(t,x) = e^{-t}\sin(\pi x)$.

Despite the nonlinear nature of the equation, Bagging-PINN continues to outperform the other methods. In the noise-free case, Table 4 shows that Bagging-PINN achieves the lowest MAE (0.0143), together with the best UQ performance, as indicated by the very small MPIW value (0.6786). Under impulse noise ($CR = 0.02$, $IF = 5$), the standard PINN exhibits significant instability in its confidence estimation,

with a large MPIW of 5.3102. In contrast, both ensemble methods remain robust. Compared to Init-Ensemble PINNs, Bagging-PINN results in a lower MAE (0.0188 vs. 0.0190) and a narrower prediction interval (MPIW 0.6843 vs. 0.6939). This method also ensures a full coverage with PICP = 100%.

**Table 4:** Test-set error metrics for Burgers equation under different noise settings.

| Setting | Model | MAE | MSE | RMSE | MPIW | PICP (%) |
|---|---|---|---|---|---|---|
| Noise-free | ANN | 0.1485 | 0.0423 | 0.2057 | 4.0780 | 100 |
| | PINN | 0.0287 | 0.0003 | 0.0030 | 5.1900 | 100 |
| | Bagging-PINN | 0.0143 | 0.0001 | 0.0077 | 0.6786 | 100 |
| | Init-Ensemble PINNs | 0.0159 | 0.0004 | 0.0096 | 0.6826 | 100 |
| Impulse noise $(CR = 0.2, IF = 5)$ | ANN | 0.0889 | 0.0191 | 0.1382 | 4.3667 | 100 |
| | PINN | 0.0211 | 0.0080 | 0.0250 | 5.3102 | 100 |
| | Bagging-PINN | 0.0188 | 0.0006 | 0.0247 | 0.6843 | 100 |
| | Init-Ensemble PINNs | 0.0190 | 0.0006 | 0.0250 | 0.6939 | 100 |

### *4.4 Sensitivity Analysis*

In order to set an appropriate value of $M$, we conduct a sensitivity analysis of $M$ with respect to model performance under noisy data across three considered equations.

The sensitivity analysis in Table 5 shows that the computational time increases approximately linearly with the ensemble size $M$. As $M$ increases, the deterministic errors (MAE, RMSE) tend to decrease and then stabilize. In terms of UQ, when $M$ is small (e.g., $M = 1$), PICP reaches 100%, but MPIW is excessively large (for example, in the ODE case, MPIW decreases from 1.8278 at $M = 1$ to 0.4064 at $M = 20$). This indicates that a sufficiently large $M$ leads to sharper and more reliable prediction intervals, avoiding overestimation of uncertainty.

**Table 5:** Summary of sensitivity analysis of ensemble size $M$ on model performance under noisy data.

| Equation | M | Time (s) | MAE | RMSE | PICP (%) | MPIW |
|---|---|---|---|---|---|---|
| ODE | 1 | 6.59 | 0.0677 | 0.0769 | 100.00 | 1.8278 |
| | 5 | 10.68 | 0.0633 | 0.0692 | 100.00 | 0.8147 |
| | 20 | 24.37 | 0.0606 | 0.0654 | 100.00 | 0.4064 |
| | 100 | 98.53 | 0.0636 | 0.0683 | 58.00 | 0.1847 |
| | 200 | 193.20 | 0.0633 | 0.0680 | 19.50 | 0.1316 |
| Wave | 1 | 7.83 | 0.1678 | 0.1958 | 100.00 | 1.0762 |
| | 5 | 18.30 | 0.0927 | 0.1151 | 100.00 | 0.9424 |
| | 20 | 41.50 | 0.1169 | 0.1445 | 80.08 | 0.4156 |
| | 100 | 127.87 | 0.1437 | 0.1767 | 32.28 | 0.1570 |
| Burgers | 1 | 2.50 | 0.0266 | 0.0344 | 100.00 | 4.5626 |
| | 5 | 11.54 | 0.0185 | 0.0257 | 100.00 | 2.2367 |
| | 50 | 109.27 | 0.0172 | 0.0229 | 100.00 | 0.6918 |
| | 200 | 418.98 | 0.0193 | 0.0254 | 100.00 | 0.3471 |

However, when $M$ becomes too large (e.g., $M = 100$ or 200), the prediction intervals become overly narrow, leading to overconfidence and a significant drop in coverage probability (e.g., PICP decreases to 19.50% at $M = 200$ for the ODE problem). This trade-off is further illustrated in Fig. 4, which shows the variation of PICP and MPIW with respect to $M$ on a logarithmic scale.

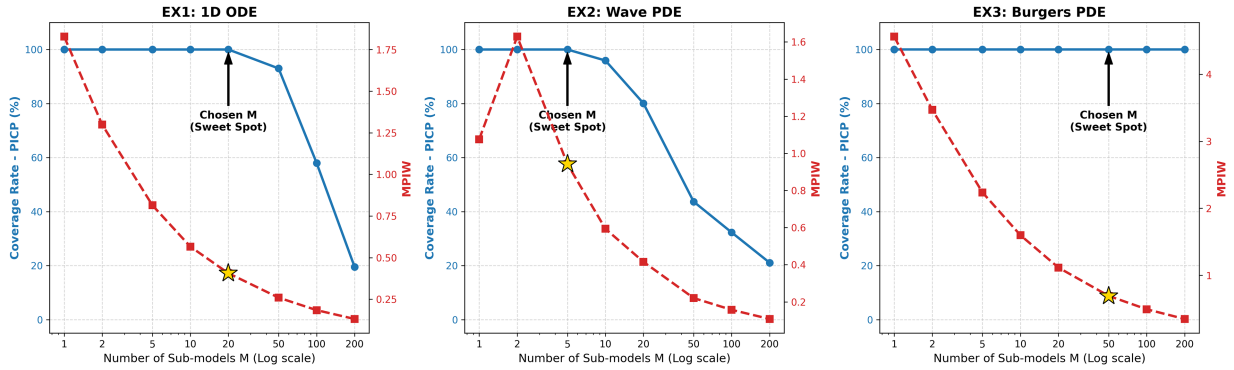


**Figure 4:** The variation of PICP and MPIW with respect to $M$ on a logarithmic scale.

Based on the above analysis, to balance training time, predictive accuracy, and the quality of prediction intervals, the optimal ensemble sizes (highlighted in the figure) are selected as $M = 20$, $M = 5$, and $M = 50$ for the three considered examples, respectively.

In summary, by employing UQ metrics (PICP and MPIW), together with thorough comparisons against a traditional ensemble baseline (Init-Ensemble PINNs) on both linear and nonlinear equations, we have experimentally demonstrated that Bagging-PINN provides more stability and UQ capability, particularly in scenarios involving anomalous noisy data.

## 5 Conclusions and Future Works

This study introduced and evaluated Bagging-PINN, a double-stochastic ensemble framework for solving ODEs and PDEs. The numerical examples on both ODEs and nonlinear PDEs under extreme impulse noise demonstrate that Bagging-PINN effectively mitigates intrinsic instability, provides highly accurate predictions without overfitting to data outliers. Furthermore, the results on PICP and MPIW confirm that our optimized ensemble framework delivers sharp, well-calibrated prediction intervals, establishing it as a highly reliable and computationally feasible tool for real-world engineering problems.

However, the scalability of the method for higher-dimensional problems or more complex systems of equations remains a challenge because the computational cost of Bagging-PINN increases linearly with the ensemble size $M$. Nevertheless, since the base models are trained completely independently on different bootstrap datasets, the approach exhibits an embarrassingly parallel property. Therefore, this limitation also suggests a future research direction that leverages parallel computing to implement Bagging-PINN for three-dimensional (3D) problems and complex systems such as the Navier–Stokes equations.

**Acknowledgement:** Not applicable.

**Funding Statement:** The authors received no specific funding for this study.

**Author Contributions:** Conceptualization, Thao Nguyen-Trang; methodology, Thao Nguyen-Trang and Hiep Ha-Hoang; software, Hiep Ha-Hoang; validation, Thao Nguyen-Trang; formal analysis, Thao Nguyen-Trang; investigation,

Thao Nguyen-Trang; resources, Hiep Ha-Hoang; data curation, Thao Nguyen-Trang and Hiep Ha-Hoang; writing—original draft preparation, Hiep Ha-Hoang; writing—review and editing, Thao Nguyen-Trang and Hiep Ha-Hoang; visualization, Thao Nguyen-Trang and Hiep Ha; supervision, Thao Nguyen-Trang. All authors reviewed and approved the final version of the manuscript.

**Availability of Data and Materials:** The source codes and data of the paper are available on GitHub under MIT licence at https://github.com/hhiep1504/SE-PINNs.

**Ethics Approval:** Not applicable.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References


1. Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys. 2019;378:686–707. doi:10.1016/j.jcp.2018.10.045.
2. Anderson JD. Computational fluid dynamics: the basics with applications. Vol. 206. Berlin/Heidelberg, Germany: Springer; 1995.
3. Liu WK, Li S, Park HS. Eighty years of the finite element method: birth, evolution, and future. Arch Comput Methods Eng. 2022;29(6):4431–53. doi:10.1007/s11831-022-09740-9.
4. Ren Z, Zhou S, Liu D, Liu Q. Physics-informed neural networks: a review of methodological evolution, theoretical foundations, and interdisciplinary frontiers toward next-generation scientific computing. Appl Sci. 2025;15(14):8092. doi:10.3390/app15148092.
5. Strang G, Fix GJ. An analysis of the finite element method. Vol. 212. Englewood Cliffs, NJ, USA: Prentice-Hall; 1973.
6. Karniadakis GE, Kevrekidis IG, Lu L, Perdikaris P, Wang S, Yang L. Physics-informed machine learning. Nat Rev Phys. 2021;3(6):422–40. doi:10.1038/s42254-021-00314-5.
7. Quarteroni A, Valli A. Domain decomposition methods for partial differential equations. Oxford, UK: Oxford University Press; 1999.
8. Hughes TJ, Cottrell JA, Bazilevs Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Comput Methods Appl Mech Eng. 2005;194(39–41):4135–95. doi:10.1016/j.cma.2004.10.008.
9. Aliakbari M, Soltany Sadrabadi M, Vadasz P, Arzani A. Ensemble physics informed neural networks: a framework to improve inverse transport modeling in heterogeneous domains. Phys Fluids. 2023;35(5):053616. doi:10.1063/5.0150016.
10. Cuomo S, Di Cola VS, Giampaolo F, Rozza G, Raissi M, Piccialli F. Scientific machine learning through physics-informed neural networks: where we are and what's next. J Sci Comput. 2022;92(3):88. doi:10.1007/s10915-022-01939-z.
11. Liu Z, Archibong G, Waheed UB, Wang S, Song C. Physics-informed Fourier-DeepONet for a generalized eikonal solution. Comput Geosci. 2026;206(1):106026. doi:10.1016/j.cageo.2025.106026.
12. Lu L, Pestourie R, Yao W, Wang Z, Verdugo F, Johnson SG. Physics-informed neural networks with hard constraints for inverse design. SIAM J Sci Comput. 2021;43(6):B1105–32. doi:10.1137/21M1397908.
13. Mishra S, Molinaro R. Estimates on the generalization error of physics-informed neural networks for approximating a class of inverse problems for PDEs. IMA J Num Anal. 2022;42(2):981–1022. doi:10.1093/imanum/drab093.
14. Luo K, Zhao J, Wang Y, Li J, Wen J, Liang J, et al. Physics-informed neural networks for PDE problems: a comprehensive review. Artif Intell Rev. 2025;58(10):1–43. doi:10.1007/s10462-025-11322-7.
15. Haitsiukevich K, Ilin A. Improved training of physics-informed neural networks with model ensembles. In: 2023 International Joint Conference on Neural Networks (IJCNN). Piscataway, NJ, USA: IEEE; 2023. p. 1–8. doi:10.1109/IJCNN54540.2023.10191822.

16. Srikitrungruang T, Aghaee Dabaghan Fard S, Lemon M, Lee J, Zhou Y. Robust physics-informed neural network approach for estimating heterogeneous elastic properties from noisy displacement data. Adv Sci. 2025;12(48):e08445. doi:10.1002/advs.202508445.
17. Cai S, Mao Z, Wang Z, Yin M, Karniadakis GE. Physics-informed neural networks (PINNs) for fluid mechanics: a review. Acta Mech Sin. 2021;37(12):1727–38. doi:10.1007/s10409-021-01148-1.
18. Le HG, Dao TP, Dang MP, Nguyen-Trang T. Behavior modeling for a new flexure-based mechanism by Hunger Game Search and physics-guided artificial neural network. Sci Rep. 2025;15(1):2015. doi:10.1038/s41598-025-85724-6.
19. Wang Y, Zhong L. NAS-PINN: neural architecture search-guided physics-informed neural network for solving PDEs. J Comput Phys. 2024;496:112603. doi:10.1016/j.jcp.2023.112603.
20. Breiman L. Bagging predictors. Mach Learn. 1996;24(2):123–40. doi:10.1007/BF00058655.
21. Krishnapriyan A, Gholami A, Zhe S, Kirby R, Mahoney MW. Characterizing possible failure modes in physics-informed neural networks. Adv Neural Inform Process Syst. 2021;34:26548–60.
22. Wang S, Teng Y, Perdikaris P. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. SIAM J Sci Comput. 2021;43(5):A3055–81. doi:10.1137/20m1318043.
23. Gal Y, Ghahramani Z. Dropout as a bayesian approximation: representing model uncertainty in deep learning. In: International Conference on Machine Learning. London, UK: PMLR; 2016. p. 1050–9.