OPEN ACCESS

Computational Mathematics EDITORIAL

The Innovation Informatics

Bayesian physics-informed neural networks to solve the partial differential equations with noise and incomplete constraints

Xi'an Li,[1,2] Jinran Wu,[3] Xiao Ning,[4] Yangshuai Wang,[5] and Lei Zhang[6,*]

[1]Ceyear Technologies Co., Ltd, Qingdao 266555, China

[2]School of Information Science and Engineering, Shandong University, Qingdao 266237, China

[3]School of Mathematics and Physics, The University of Queensland, St Lucia 4072, Australia

[4]School of Sciences, China Pharmaceutical University, Nanjing 210009, China

[5]Department of Mathematics, Faculty of Science, National University of Singapore, Singapore 119077, Singapore

[6]School of Mathematical Sciences, Institute of Natural Sciences and MOE-LSC, Shanghai Jiao Tong University, Shanghai 200240, China

*Correspondence: lzhang2012@sjtu.edu.cn

Received: January 23, 2026; Accepted: April 13, 2026; Published Online: April 14, 2026; https://doi.org/10.59717/j.xinn-inform.2026.100039

Citation: Li X., Wu J., Ning X., et al. (2026). Bayesian physics-informed neural networks to solve the partial differential equations with noise and incomplete constraints. *The Innovation Informatics* 2:100039.

Partial differential equations (PDEs) have extensive applications across various fields, as they can model the physical properties and behaviours of complex systems, but their analytical solutions are seldom available. Classical numerical methods, such as FDM and FEM,[1] can provide solutions to given PDEs, but they may face challenges for irregular domains, high dimensions, and inverse-problem cases due to their computational complexity and mesh-dependent properties. In recent years, Physics-informed neural networks (PINNs) have sprung up in scientific machine learning and attracted lots of researchers' attention.[2] Theoretically, this method casts the solution of a given PDE into an optimisation problem by incorporating physical laws into the neural networks via automatic differentiation. In form, deep neural networks (DNNs) are configured as their solvers, and the target loss function comprises the residual terms arising from enforcing physical constraints and the discrepancy between the observed data and the neural network predictions. While effective for well-posed problems with high-quality data, the general PINNs lack mechanisms to quantify uncertainty, making them ill-suited for real-world applications where measurements are inherently noisy, physical parameters are uncertain, and constraints are incomplete. To address these limitations, researchers have developed Bayesian PINNs (B-PINNs) by combining the PINNs with Bayesian inference that is robust for uncertainty quantification in noisy observations and limited constraints.[3] Subsequently, it is extended to address a broad class of PDEs, such as the phonon Boltzmann transport equation and the governing equations for hybrid nanofluid systems. Moreover, application-oriented variants such as inverse BPINNs have been proposed for engineering problems, notably seismic inversion in geologically complex settings.[4]

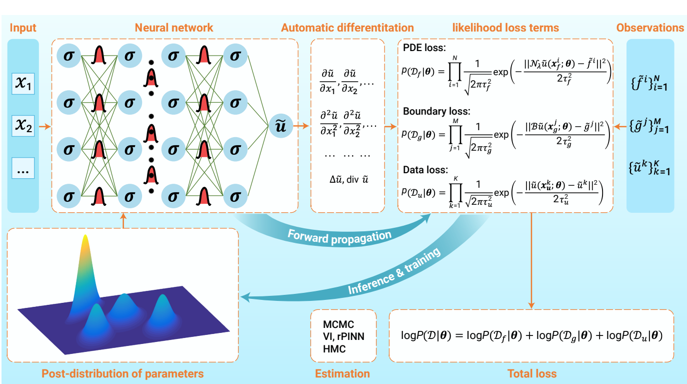

Figure 1. The semantic diagram of B-PINNs for solving PDEs with noise or incomplete constraints Input points are fed into a normal or multi-scale DNN to generate predictions. By incorporating physical laws via automatic differentiation of the predictions, the log-likelihood loss is derived. Finally, the posterior distribution is obtained through training and inference.

UNIFIED ARCHITECTURE OF B-PINNS

For systematically introducing the concept and framework of B-PINNs, we consider the following PDE of the form:

$$\begin{aligned} \mathcal{N}_{\lambda} u(x) &= f(x),\ x \in \Omega \\ \mathcal{B} u(x) &= g(x),\ x \in \Gamma \end{aligned} \tag{1}$$

where $\mathcal{N}_{\lambda}$ is a differential operator, u is the solution, λ represents the physical parameters, f is a forcing term, Ω is a interested domain in d-dimensional Euclidean space, $\mathcal{B}$ is a boundary operator, and Γ stands for the boundary of Ω.

In B-PINNs, the solution u is represented by a Bayesian neural network $u(x;\theta)$ with parameters θ following a prior distribution $P(\theta)$. Bayesian neural networks (BNNs) are the probabilistic extension of conventional neural networks, and the network parameters are modelled as a probability distribution.[2,5] Specifically, the weights and biases in the l_{th} hidden layer of BNNs are treated as random variables, such as assumed to follow a zero-mean Gaussian distribution or a Laplace distribution. In practice, B-PINNs apply Bayes' theorem to update the network parameters based on observed data, and it is

$$P(\theta|\mathcal{D}) = P(\theta)P(\mathcal{D}|\theta)/P(\mathcal{D}) \tag{2}$$

where $P(\theta|\mathcal{D})$ is the posterior distribution that captures all uncertainties about the network parameters for given available data, $P(\theta)$ is the prior distribution, $P(\mathcal{D})$ represents the distribution of given available data, and $P(\mathcal{D}|\theta)$ is the likelihood function used to combine the physical law and observed data with the network parameters. Therefore, we can estimate the posterior distribution of network parameters and provide an approximation of the solution according to a given training dataset. Then, under the physical constraints of Eq. (1), a log-likelihood loss is required and expressed as:

$$\log P(\mathcal{D}|\theta) = \log P(\mathcal{D}_f|\theta) + \log P(\mathcal{D}_g|\theta) \tag{3}$$

with

$$P(\mathcal{D}_f|\theta) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\tau_f^2}} \exp\left(-\frac{\left\|\mathcal{N}_{\lambda}\tilde{u}\left(x_f^i;\theta\right)-\tilde{f}^i\right\|^2}{2\tau_f^2}\right) \tag{4}$$

and

$$P(\mathcal{D}_g|\theta) = \prod_{j=1}^{M} \frac{1}{\sqrt{2\pi\tau_g^2}} \exp\left(-\frac{\left\|\mathcal{B}\tilde{u}\left(x_g^j;\theta\right)-\tilde{g}^j\right\|^2}{2\tau_g^2}\right) \tag{5}$$

where the $\mathcal{D}$ stands for the observed dataset that composed of the sparse collocation points and the corresponding evaluation of f and g, i.e., $\mathcal{D} = \mathcal{D}_f \cup \mathcal{D}_g$ with $\mathcal{D}_f = \{(x_f^i, \tilde{f}^i)\}_{i=1}^{N}$ and $\mathcal{D}_g = \{(x_g^j, \tilde{g}^j)\}_{j=1}^{M}$. These observations are i.i.d. Gaussian random variables, i.e.,

$$\tilde{f}^i = f(x_f^i) + \varepsilon_f^i, i = 1,2,\cdots,N \text{ and } \tilde{g}^j = g(x_f^j) + \varepsilon_g^j, j = 1,2,\cdots,M. \tag{6}$$

In which $\varepsilon_f = \{\varepsilon_f^1, \varepsilon_f^2, \cdots\}$ and $\varepsilon_g = \{\varepsilon_g^1, \varepsilon_g^2, \cdots\}$ are independent mean-zero Gaussia n noise with standard deviations τ_f and τ_g, respectively. If there are supplementary observed data of solution, i.e., $\mathcal{D}_u = \{(x_u^k, \tilde{u}^k)\}_{k=1}^{K}$ with $\tilde{u}^k = u(x_u^k) + \varepsilon_u^k$, $k = 1,2,\ldots,K$, then a likelihood term is provided to quantifies the mismatch between B-PINN's predictions and empirical measurements. It is

$$P(\mathcal{D}_u|\theta) = \prod_{k=1}^{K} \frac{1}{\sqrt{2\pi\tau_u^2}} \exp\left(-\frac{\left\|\tilde{u}\left(x_u^k;\theta\right) - \tilde{u}^k\right\|^2}{2\tau_u^2}\right) \tag{7}$$

where τ_u is the standard deviations for mean-zero Gaussian noise $\boldsymbol{\varepsilon}_u = \{\boldsymbol{\varepsilon}_u^1, \boldsymbol{\varepsilon}_u^2, \ldots\}$. At the end, the schematic of the B-PINNs architecture is described and plotted in Figure 1.

In B-PINNs, uncertainty is typically viewed as having two complementary components: aleatoric uncertainty, which represents inherent randomness in the data-generation process and is modelled through the likelihood, and epistemic uncertainty, which reflects model uncertainty due to limited data and is captured by the posterior distribution over parameters. This posterior uncertainty propagates to predictions at a new input x^* via the Bayesian predictive distribution

$$P(u^* \mid x^*; \mathcal{D}) = \int P(u^* \mid x^*; \theta)\, P(\theta; \mathcal{D})\, d\theta, \tag{8}$$

where u^* denotes the predicted solution at x^*. As a result, B-PINNs produce predictive distributions rather than point estimates, enabling uncertainty bands to be placed around the predicted solution.

ESTIMATION APPROACHES FOR B-PINNS

Generally, the network parameters θ for BNNs are a high-dimensional random variable, and the corresponding posterior distribution is very tough to obtain through direct calculation or effective sampling. Alternatively, approximation methods such as Variational Inference (VI) and Hamiltonian Monte Carlo (HMC) can be used for posterior inference.

VI assumes that the posterior distribution belongs to a family of parameterised distributions and transforms Bayesian inference into an optimisation problem.[5] It approximates the true posterior $P(\theta|\mathcal{D})$ with a simpler parametric distribution $Q_\phi(\theta)$ (the variational family), then optimizes the parameters ϕ to minimize the Kullback-Leibler (KL) divergence of $Q_\phi(\theta)$ and $P(\theta|\mathcal{D})$. It is

$$D_{KL}[Q_\phi(\theta)||P(\theta|\mathcal{D})] = \int Q_\phi(\theta) \log \frac{Q_\phi(\theta)}{P(\theta|\mathcal{D})}. \tag{9}$$

For B-PINNs, common variational families include mean-field Gaussian distributions or more sophisticated approximations like normalising flows. Its advantages include three aspects: (1)its drawing sample is inexpensive compared to MCMC in the training process; (2) it handles large datasets and models better than MCMC methods; and (3) it can leverage existing deep learning optimisation tools and infrastructure.

HMC represents a gold standard for posterior inference due to its ability to efficiently explore high-dimensional parameter spaces.[5] It utilises Hamiltonian dynamics to propose distant states with high acceptance probability, overcoming the random walk behaviour of simpler MCMC methods. In the context of B-PINNs, HMC samples from the posterior distribution $P(\theta|\mathcal{D})$ by simulating Hamiltonian dynamics in the parameter space; this mode of movement is jumping, which enables the sampling process to traverse different parameter regions and approach the true posterior distribution more quickly. In addition, to accelerate parameter estimation and Bayesian inference, advanced methods such as Natural Gradient VI, Low-Rank VI, Operator VI, Stochastic Gradient HMC, SGLD, NUTS, and Vectorised HMC have been developed. They are particularly well-suited for high-dimensional PDEs and large-scale training datasets.[5] Moreover, a hybrid approach is usually effective for real-time problems: initialising parameters via SGD optimisers (e.g., Adam) followed by posterior sampling using the above advanced algorithms. These strategies can well balance the fidelity and the computational efficiency.

APPLICATIONS OF B-PINNS FOR SOLVING PDES

B-PINNs have been shown to deliver robust PDEs' solutions even when observational data are corrupted by noise and the underlying physical constraints are incomplete or uncertain. Because real-world science and engineering almost always involve making inferences from imperfect information, B-PINNs are broadly applicable: they can handle forward problems with noisy boundary or initial conditions by propagating measurement uncertainty and producing confidence intervals for predictions in unobserved regions; they support inverse problems and parameter discovery by estimating unknown PDEs parameters (such as material properties, reaction rates, or diffusion coefficients) from sparse, noisy measurements; they accommodate heteroscedastic settings by allowing noise levels to vary across data points via per-sample variance in the likelihood; and they remain effective when constraints are only partially known, for instance by learning governing equations from historical data and embedding them into a B-PINNs framework for prediction with uncertainty quantification. Moreover, B-PINNs extend naturally to stochastic PDEs by modelling uncertain coefficients or boundary conditions as random variables with prior distributions, enabling uncertainty-aware solutions for cases like diffusion with random coefficients, wave propagation in heterogeneous media, and fluid flow under uncertain boundary conditions.

DISCUSSION

B-PINNs deliver reliable PDE solutions under noisy or incomplete constraints by integrating physics-based modeling with Bayesian inference. Rather than serving solely as numerical solvers, they offer a probabilistic framework for scientific inference that respects physical laws while quantifying uncertainty. Despite strong empirical performance, their theoretical foundations remain underdeveloped, with open questions on posterior consistency, error bounds, convergence and generalization. Recent numerical analysis further identifies training error as a key bottleneck, motivating deeper investigation into the optimization landscape of physics-informed losses and the design of more effective training strategies. Relevant theoretical tools include PCA-based Bayesian inference, Sobolev space and functional approximation theories, the neural tangent kernel and optimal transport.

Promising future applications span multiscale and multiphysics systems, digital twins, real-time monitoring, experimental design and data-driven discovery of governing laws. Broader industrial adoption will require integration with existing simulation workflows, certification for safety-critical contexts and user-friendly interfaces. Meanwhile, advances in neural architectures—such as physics-informed Kolmogorov–Arnold Networks, geometric deep learning, and operator learning in noisy settings—offer opportunities to enhance accuracy, efficiency and interpretability while preserving the uncertainty quantification that distinguishes B-PINNs from deterministic approaches.

REFERENCES

1. Li Z., Qiao Z. and Tang T. (2017). Numerical solution of differential equations: Introduction to finite difference and finite element methods. Cambridge University Press. DOI: 10.1017/9781316678725
2. Raissi M., Perdikaris P. and Karniadakis G. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **378**:686–707. DOI:10.1016/j.jcp.2018.10.045
3. Yang L., Meng X. and Karniadakis G. (2022). B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data. *J. Comput. Phys.* **425**:109913. DOI:10.1016/j.jcp.2020.109913
4. Liu Z., Zhang J., Chen Y., et al. (2025). From physics constraints to trustworthy Bayesian reasoning: Synergies of PINN, iPINN, and iBPINN in Prestack AVO inversion. *IEEE Geosci. Remote.* **63**:1–24. DOI:10.1109/TGRS.2025.3636416
5. Zyphur M.J. and Oswald F.L. (2015). Bayesian estimation and inference: A user's guide *J. Manage.* **41**:390–420. DOI:10.1177/0149206313501200

FUNDING AND ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (Grant No. 12501601). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

AUTHOR CONTRIBUTIONS

Xi'an Li drafted the manuscript; Xiao Ning and Yangshuai Wang reviewed and improved this manuscript; Jinran Wu and Lei Zhang provided critical revisions and intellectual guidance. All authors contributed to the manuscript and approved the final version.

DECLARATION OF INTERESTS

The authors declare no competing interests.