# Quantification of modeling uncertainty in physics-informed neural networks by interval arithmetic

Zlatan Dimitrov, Petar O. Hristov
[1)] *GATE Institute, 1164 Sofia, Bulgaria, {zlatan.dimitrov, petar.hristov}@gate-ai.eu*

**Abstract.** The seemingly unrestrained advancements in machine learning technologies drive engineers to increasingly often rely on data-driven approaches in the design of complex systems. This not only requires extensive data to capture the underlying behavior, but it effectively ascribes complete trustworthiness to data. However, real-world scenarios frequently provide only sparse, noisy, or otherwise corrupted measurements. Compounding this challenge is the need to balance resource utilization and performance.

To alleviate some of the burden related to the models exclusively relying on data, the class of physics-informed machine learning emerged, which integrates knowledge about the governing physical laws, with data from the system. The majority of physics-based machine learning models, however, remain black boxes and the epistemic uncertainty stemming from the model itself remains unquantified or quantified in a biased way.

This work explores the extension of physics-informed neural networks with an interval description of the epistemic uncertainty, integral to the model, thereby avoiding subjective assignments of prior beliefs to components of the model. In the proposed implementation, data processing is done through numerical weights and two separate fully connected layers are used to estimate the center and interval uncertainty of the prediction, to alleviate the problem of interval repeated variables. In addition, the architecture leverages a physics-informed loss function considering whether the output of the model satisfies the underlying partial differential equation and boundary conditions.

We validate the performance of the architecture on simplified synthetic data, representing a structural dynamics problem, in particular a noisy dataset generated using an analytical solution to the Euler-Lagrange equation. The interval PINN, proposed in this paper, provides a way for evaluating a cantilever beam's free oscillations in a way that considers both the underlying structural mechanics knowledge and the available measurement data. The architecture is more explainable than purely data-based models, while avoiding applying prior bias on the quantification epistemic uncertainty.

**Keywords:** physics-informed machine learning, epistemic uncertainty, interval arithmetic, cantilever beam, partial differential equation (PDE)

## 1. Introduction

The reliable modeling of physical systems is important during tasks like engineering design, system identification, and digital twin development. This modeling is traditionally done using a numerical partial differential equation (PDE) method, such as the finite element method (FEM), which solves

a discretised version of the underlying mathematical equations on a computational mesh. Such approaches offer limited capability for the integration of observations, since they rely only on physical laws in an idealized environment. In addition, physics-based models require the full simulations to be run end to end for each new parameter combination, making their use in design and virtual experimentation computationally expensive (Taflove et al., 2005; Rylander et al., 2013).

In a somewhat parallel thread, data-based machine learning approaches (ML) have recently grown in popularity as a tool for solving physical problems. These methods perform well as surrogates, interpolating between different observations or parameter configurations, or when the available physics is not sufficient to describe the problem under investigation and the observation data is abundant, as empirical models. The limitation of these models stems from the fact that they are not grounded in natural laws, which may cause their results to become non-physical (Bishop, 2009; Raissi et al., 2019). Physics-informed machine learning bridges the gap between pure data-based modeling and idealized physics, by integrating the physical knowledge into parts of the ML model, which allows the latter to respect known conservation properties and invariants. Physics-informed neural networks (PINN) are a prominent example of such models for which the loss function is augmented with physical terms relevant to the problem at hand (Raissi et al., 2019). This approach generates a data-based surrogate, which is trained on fewer data than traditional ML models, while adhering to the underlying physical laws, at a cost lower than traditional physics-based models (Lee et al., 2024; Babaee et al., 2020; Lino et al., 2023).

A limitation of the original PINN architecture, carried over from classical neural networks is that it produces deterministic values for each input combination it is evaluated with, limiting the information about the uncertainty that is due to the use of an approximation, in place of the original model or instead of measurements.

Quantifying the predictive uncertainty of surrogate models is a classical problem that has seen many advancements in the last three decades. For example, the widely-used class of Gaussian process (GP) regression models provides a closed-form expression for the variance at each output, based either on a frequentist maximum-likelihood construction (Forrester et al., 2008) or on a Bayesian prior-to-posterior analysis (Rasmussen and Williams, 2006). Similar attempts at equipping neural networks with uncertainty quantification (UQ) capabilities have resulted in using dropout and constructing Bayesian neural networks. Gal and Ghahramani (2015) show that using dropout is equivalent to model averaging in mean predictions, but the results can also be interpreted in a Bayesian context as providing a UQ similar to that of a deep GP model. Bayesian neural networks (BNN) (Arbel et al., 2023), preserve the structure of the model, but endow its weights and biases with prior probability distribution which are propagated through the Bayesian pipeline to a predictive posterior.

The most important uncertainty component to be quantified in a surrogate modelling prediction is that due to lack of model or observational realizations at the requested input combination. This uncertainty is epistemic and, as argued in literature, it should be treated differently to random variations (Ferson and Ginzburg, 1996). However, both dropout and BNN use the concepts of probability theory to quantify it. To address the matter, research has been conducted into *interval neural networks* (INN), which use the mathematical interval as a primitive for characterizing epistemic uncertainty. The most common strategies for constructing an INN include propagating full interval weights and biases (Garczarczyk, 2000), representing the intervals as a center and radius

(Sadeghi et al., 2019), and training a pair of neural networks, one for each bound of the interval predictions (Huang et al., 1998).

In an attempt to provide a quantification of their predictive uncertainty, PINN have inherited the Bayesian treatments for NN UQ (Zhang et al., 2019; Yang et al., 2021). However, to the best of the authors knowledge the extension of INN to PINN architectures has not been addressed comprehensively. This paper introduces an interval-based PINN architecture, which quantifies the predictive epistemic uncertainty, by systematically treating all components of the PINN loss function. The performance of the model in terms of achieving a calibrated, yet practical UQ is demonstrated on a dynamics-rich problem of a freely vibrating cantilever beam, through the modelling of its first three eigenmodes. Some practical considerations for modelling unsteady problems are also discussed.

The remainder of the paper is organised as follow. Section 2 reviews the methodologies relevant to the problem and introduces the proposed interval-PINN model. Section 3 presents the experiments conducted to evaluate the performance of the proposed model and Section 4 draws some conclusions from the conducted work.

## 2. Methodology

Throughout this paper, a freely-oscillating cantilever beam is used as a running example to demonstrate the proposed interval PINN architecture. Specifically, the second and third eignemodes of the beam are analyzed, as they could be more difficult to capture by simpler models.

### 2.1. Cantilever beam setup

The free oscillation of a cantilever beam is described by the Euler-Lagrange equation:

$$\begin{aligned} & EI\ \partial_x^4 w(x,t) + \mu\ \partial_t^2 w(x,t) = 0 \ \text{ for } x \in (0,L),\, t \geq 0 \\ & \text{subject to :} \\ & \quad w(0,t) = \partial_x w(x,t)|_{x=0} = \partial_x^2 w(x,t)|_{x=L} = \partial_x^3 w(x,t)|_{x=L} = 0 \\ & \quad w(x,0) = w_0 \\ & \quad \partial_t w(x,t)|_{t=0} = 0 \end{aligned} \tag{1}$$

where $x$ is the spatial and $t$ the temporal coordinate, $w(x,t)$ is the beam displacement as a function of space and time, $E$ is the material Young's modulus, $I$ is the moment of area of the beam, and $\mu$ is its mass per unit length (Hjelmstad, 2025).

Eq. (1) has an analytical solution of the form:

$$w(x,t) = \sum_{i=1}^{\infty} A_i\, \phi_i(x) \cos(\omega_i t) \tag{2}$$

which states that the displacement, $w(x,t)$ is a superposition of all mode shapes, $\phi_i(x)$:

$$\phi_i(x) = \left\{ \cosh(\beta_i x) - \cos(\beta_i x) + \frac{\cosh(\beta_i L) + \cos(\beta_i L)}{\sinh(\beta_i L) + \sin(\beta_i L)} \left(\sin(\beta_i x) - \sinh(\beta_i x)\right) \right\} \tag{3}$$

where $\beta_i$ are functions of the natural frequencies of the beam.

### 2.2. Physics-informed neural networks

Physics-informed neural networks are regularized using known PDEs, which are applied to the output of the NN and represent the expected behavior of the system, as defined by the idealized laws encoding the relevant physics. In the case of the cantilever beam, the system of equations in Eq. (1) is applied on the network outputs. The NN is trained using double precision and a combination of the Adam and L-BFGS optimizers, as using second-order optimizers has been show to achieve better convergence during training (Kiyani et al., 2025).

### 2.3. Interval neural networks

The interval neural network (INN) architecture quantifies uncertainty by returning an upper and lower bounds at each predictive site. This interval does not imply any probability distribution of the values lying therein. The architecture implemented in this paper represents the interval as a center, $c(x,t)$ and radius, $\delta(x,t)$. In general, the INN is penalized for large width (high $\delta(x,t)$) and for training outputs outside the interval. In the particular case of an interval PINN, the center and width are subject to different penalties: the center is based on the PDE, the boundary conditions (BC), the initial conditions (IC), and the data, while the width depends only on the data. Deterministic weights have been chosen, due to their better numerical stability (Sadeghi et al., 2019).

### 2.4. Interval PINN for cantilever modeling

The fourth order PDE in Eq. (1) can be solved using automatic differentiation in a deep learning framework such as PyTorch (Paszke et al., 2019). A "time-marching" strategy has been implemented which separates the solution in time into $N$ periods with the length of half an oscillation as proposed in (Krishnapriyan et al., 2021; Lee et al., 2024). There the authors model the first eigenmode of a cantilever beam using a PINN and show that training a separate neural network for each period allows the model to focus on local features and avoid the ill-conditioning, stemming from the soft constraints of the PINN architecture. The method proposed in this paper combines the splitting in time with an INN architecture. The loss function of the $n$-th ($n = \{1, \ldots, N\}$) PINN is given by:

$$\mathcal{L}^{(n)} = \lambda_{\text{PDE}}\,\mathcal{L}^{(n)}_{\text{PDE}} + \lambda_{\text{BC}}\,\mathcal{L}^{(n)}_{\text{BC}} + \lambda_{\text{IC}}\,\mathcal{L}^{(n)}_{\text{IC/transfer}} + \lambda_D\,\mathcal{L}^{(n)}_{D} \tag{4}$$

where each subscript denotes terms pertaining to the relevant information. Since the temporal and spatial scales are different and diverge further when using higher-order derivatives, a set of normalizing constants, $\lambda_x$, is used (Lee et al., 2024). The physics part of Eq. (4) is given by:

$$\lambda_{\text{PDE}}\,\mathcal{L}^{(n)}_{\text{PDE}} = \frac{L^8}{EI^2\,Y^2_{\text{ref}}} \left\langle \left( EI\,\partial^4_x c^{(n)} + \mu\,\partial^2_t c^{(n)} \right)^2 \right\rangle_{\Omega^{(n)}} \tag{5}$$

where $c^{(n)} \equiv c^{(n)}(x,t)$ is the center of the $n$-th interval prediction, $L$ is the cantilever beam length, and $\Omega^{(n)}$ is the 2D collocation point domain where the PDE is evaluated for the $n$-th PINN. Mean calculation over collocation points is indicated by $\langle\cdot\rangle$. The term $Y_{\text{ref}} = F_{\text{tip}}L^3/EI$ is the reference

displacement based on the initial cantilever tip position, which is used to scale the displacement in the loss functions for numerical stability reasons.

The part of Eq. (4), which enforces the fixed boundary conditions ($w = 0$, $w' = 0$) and the free-end conditions (zero moment and shear force) of the cantilever beam, is given by:

$$\lambda_{\mathrm{BC}}\,\mathcal{L}_{\mathrm{BC}}^{(n)} = \frac{1}{Y_{\mathrm{ref}}^2}\Big[\underbrace{\langle (c^{(n)})^2\rangle}_{\text{fixed: } w=0} + L^2\,\underbrace{\langle (\partial_x c^{(n)})^2\rangle}_{\text{fixed: } w'=0} + L^4\,\underbrace{\langle (\partial_x^2 c^{(n)})^2\rangle}_{\text{free: moment}} + L^6\,\underbrace{\langle (\partial_x^3 c^{(n)})^2\rangle}_{\text{free: shear}}\Big] \tag{6}$$

where the $k$-th derivative is multiplied by $L^{2k}$, in order to account for the scale in each.

The data available about the initial state of the cantilever beam, which is supplied to the first PINN ($n = 1$) is constructed as:

$$\lambda_{\mathrm{IC}}\,\mathcal{L}_{\mathrm{IC}}^{(n)} = \frac{1}{Y_{\mathrm{ref}}^2}\Big[\underbrace{r_c\,\langle \max(0, |w_d - c^{(1)}| - \delta^{(1)})^2\rangle}_{\text{containment}} + \underbrace{r_t\langle (\delta^{(1)})^2\rangle}_{\text{tightness}} + \tau_c^2\,\underbrace{\langle (\partial_t c^{(1)} - \partial_t w_d)^2\rangle}_{\text{velocity IC}}\Big]_{t=t_0} \tag{7}$$

where $\delta^{(n)} \equiv \delta^{(n)}(x,t)$ is the interval radius, $w_d$ is the data, $\tau_c = (\mu L^4/EI)^{1/2}$ is a characteristic evolutionary time scale (Lee et al., 2024), and $r_c$ and $r_t$ are the containment and tightness loss coefficients, respectively.

For subsequent interval PINN ($n > 1$), the temporal dynamics is captured through the transfer loss component, which enforces continuity of the interval center, width, and velocity between consecutive PINNs at the transition time $t = t_n$:

$$\lambda_{\mathrm{IC}}\,\mathcal{L}_{\mathrm{transfer}}^{(n)} = \frac{1}{Y_{\mathrm{ref}}^2}\Big[\underbrace{\langle (c^{(n)} - c^{(n-1)})^2\rangle}_{\text{centre continuity}} + r_w\,\underbrace{\langle (\delta^{(n)} - \delta^{(n-1)})^2\rangle}_{\text{width continuity}} + \tau_c^2\,\underbrace{\langle (\partial_t c^{(n)} - \partial_t c^{(n-1)})^2\rangle}_{\text{velocity continuity}}\Big]_{t=t_n} \tag{8}$$

where $r_w$ is the width continuity coefficient.

The final component of the loss function in Eq. (4) is the one defining the interval width to contain the available data:

$$\lambda_{\mathrm{D}}\,\mathcal{L}_{D}^{(n)} = \frac{1}{Y_{\mathrm{ref}}^2}\Big[\underbrace{r_c\langle \max(0,\, |w_d - c^{(n)}| - \delta^{(n)})^2\rangle}_{\text{containment}} + \underbrace{r_t\langle (\delta^{(n)})^2\rangle}_{\text{tightness}} + \underbrace{r_d\langle (w_d - c^{(n)})^2\rangle}_{\text{centre}}\Big]_{\Omega^{(n)}} \tag{9}$$

where $r_d$ is the center loss coefficient.

## 3. Numerical experiments

The problem was simulated for parameters $L = 1$ m, $E = 1$ GPa, $I = 1$ m$^4$ a duration of 1.5 oscillations, and the domain was split in $N = 3$ periods, for each of which a separate PINN is trained.

### 3.1. Training data

Each interval PINN was trained on synthetic data, generated through a the analytical displacement solution in Eq. (2), with added noise of the form

$$\tilde{\phi}_i(x) = \phi_i(x)(1 + \epsilon_i) \tag{10}$$

where the noise, $\epsilon$, was given three distinct forms to test the performance of the model:

$$\epsilon_i \sim \sigma \cdot \mathcal{E}, \qquad \mathcal{E} \in \begin{cases} \mathcal{N}(0,\, 1) \\ \mathcal{U}(-1,\, 1) \\ 2\,\mathrm{Beta}(5,\, 2) - 1 \end{cases} \tag{11}$$

The noise amplitude was scaled with the displacement, which represents a sensor with an accurate, well-calibrated zero offset, and measurement uncertainty proportional to the displacement. Test cases were generated for three levels of noise, $\sigma \in \{0.05, 0.15, 0.25\}$.This process produced 9 test cases for each mode at which free oscillations are simulated (mode 2 and 3), totaling 18 test cases. Training data then consists of 100 points in space ($x$) at time $t = 0$, representing initial conditions. In addition, for mode 2, measurement data consisting of six equally-spaced points over the length of the beam and 20 equally-spaced points in time, over the three half-periods analyzed, was used to train the interval PINN. For mode 3, a similar arrangement of measurements, with 11 points over the beam and 20 points in time was used.

### 3.2. Model training

The hyperparameters $r_c$, $r_t$, $r_d$ introduced in Eqs. 7 and 9 and $r_w$, which appears in Eq. 8 are calibrated using a single representative case per mode, subject to a noise level $\sigma = 0.15$. Those cases were simulated with a constant noise level, to evaluate performance across the entire span of the beam. The model is subsequently validated on the full set of 18 test cases. To account for the stochastic nature of neural network parameter initialization, each configuration is trained three times independently.

### 3.3. Model evaluation

Model performance was assessed according to the indicators specified in Tab. I. These measures have been selected as they quantify the lack of model reliability in terms of the proportion of (FracOut), and distance to (Dist2Bnd), values outside of the interval, averaged over space and time. The "Overshoot" indicator additionally evaluates the average level to which the model overshoots the observations. Comparison is done across different levels of noise, as the measures used are $\sigma$-invariant.

The results presented in Fig. 1 indicate that the interval PINN encompasses the majority of observed data points within its predictive uncertainty bounds across all considered noise types, with FracOut values remaining below 0.25 for most test cases. In instances where data points lie outside the predicted interval, the mean distance to the nearest bound remains small (Dist2Bnd $< 0.005$m), suggesting that such points are situated in close proximity to the interval boundary.

Table I. Description of evaluation measures used to assess model performance. Here $y_i$ is the value of the observation.

| Measure | Formulation | Description |
|---|---|---|
| Dist2Bnd | $\langle \max(\lvert y_i - c \rvert - \delta, 0) \rangle$ | Mean distance to points outside the prediction interval |
| Overshoot | $\langle \max(\delta - \lvert y_i - c \rvert, 0) \rangle$ | Mean margin for points inside the interval |
| FracOut | $\langle \mathbf{1}(\lvert y_i - c \rvert > \delta) \rangle$ | Fraction of test points falling outside the predicted interval |

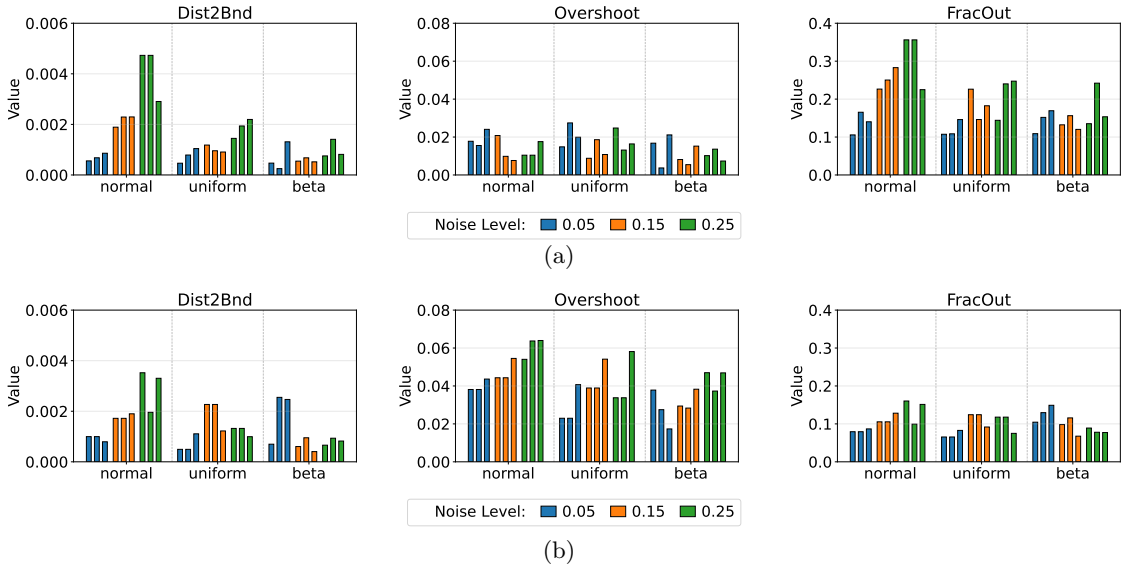



*Figure 1.* Interval PINN performance over 3 runs on 18 test cases for: (a) mode 2 and (b) mode 3.

The "Overshoot" indicator further reveals that the interval width remains consistent across the different types of noise both for mode 2 and for mode 3. From the results, it can be seen, that the interval PINN captures relatively more of the mode 3 data (lower "FracOut"), but suffers from underfitting, due to the more complicated displacement distribution over time and space (high "Overshoot"). Those observations are also evident in the following Figures and will be discussed further.

To explore the performance of the interval PINN in space and time, and especially the performance of the partitioning strategy, a series of displacement heatmaps were constructed. Fig. 2 displays the center and width of the prediction, and the noisy training data for mode 2 with $\sigma = 0.25$ and different noise models. The heatmaps show that the center of the interval PINN predictions successfully reconstructs the displacement of the beam, with minimal discontinuities across partitions. The beta noise model shown in Fig. 2a exhibits the best performance in this sense.

Finally, the reliability of the model in capturing the beam dynamics and its predictive uncertainty are shown in Figs. 3 and 4 for modes 2 and 3, respectively. The interval PINN prediction centers follow the underlying physics while its predictive uncertainty bound the test data, shown as red dots. The wider intervals observed for PINN 1 can be attributed to the fact that scatter in the initial conditions data (blue dots) away from the idealized physics is larger. This also explains the larger interval widths in Fig. 2b and c are wider, which are caused by the same phenomenon. Despite only sparse data being available beyond the initial condition $t = 0$ (green squares), the interval PINN propagates appropriate interval width to subsequent time steps, enabling it to capture the underlying data scatter.

Fig. 4 shows an identical depiction for mode 3 with $\sigma = 0.15$. The Interval PINN successfully captures the dynamics of the more complex mode, but predicts relatively high interval widths. Increasing the noise leads to degrading performance, which can be addressed by increasing the number of training data points fed to the model through time. Additionally, the more complex spatial dynamics lead to difficult convergence resulting in some of the interval PINN models, such as PINN 2 in Fig. 4c, to have large prediction intervals.

## 4. Conclusion

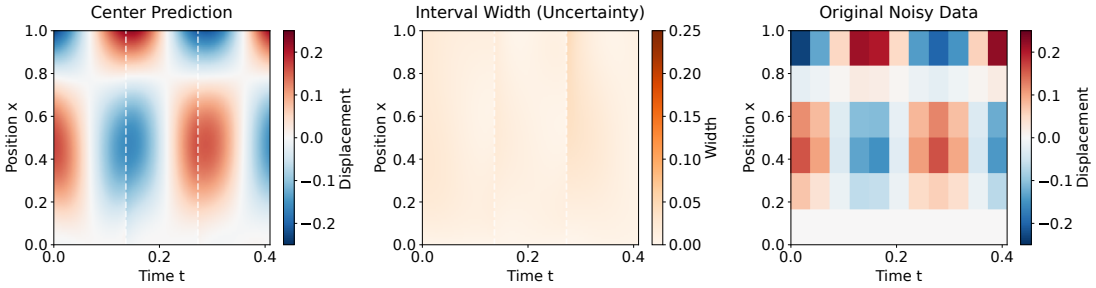
A loss function formulation to train an interval PINN was proposed, which quantifies the model uncertainty in the prediction of the surrogate model without resorting to probabilistic construction and subjective prior information. The model uses a center-radius representation with physics-informed losses that enforce the underlying PDE and boundary conditions while fitting noisy data through interval containment and tightness terms.

The proposed architecture was evaluated using a fourth-order PDE describing a the free-oscillation dynamics of a cantilever beam, tested across 18 scenarios with three noise distributions (normal, uniform, beta) and three noise levels. The results demonstrate that the interval PINN shows robustness towards numerical instabilities and captures the majority of data points within its uncertainty bounds while maintaining physically consistent center predictions across all noise types.

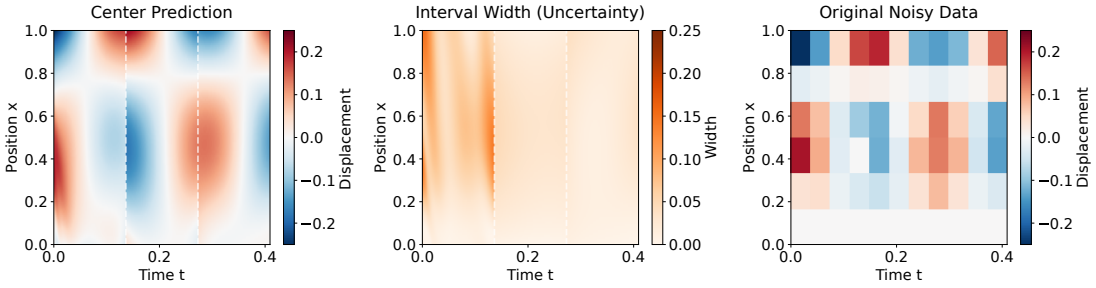
Further work will focus on implementing interval weights and biases within the network, developing parallel training of the separate PINN models to allow bidirectional information propagation, and validating the architecture on additional PINN problems across different fields.

## Acknowledgements

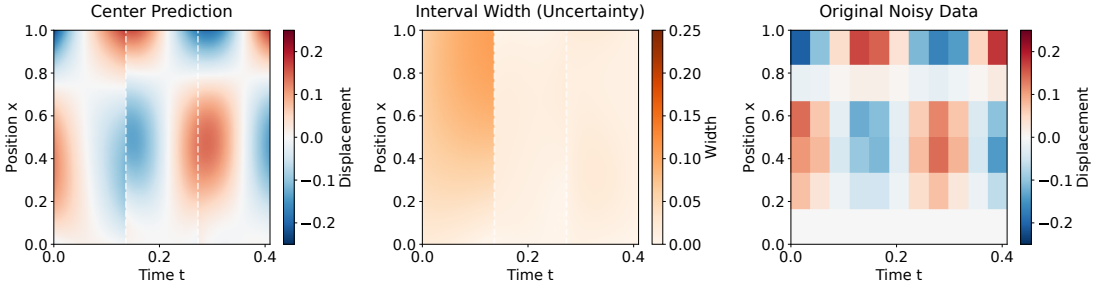
This research was conducted with the support of the Swiss National Science Foundation and the Bulgarian Ministry of Education and Science under Grant Agreement No. IZPYZ0_228861 for the project “Automatic maximally rigorous uncertainty quantification to enable design for certification” (unCertify). The work also received funding from the GATE project funded by the Horizon 2020 WIDESPREAD-2018-2020 TEAMING Phase 2 programme under grant agreement no. 857155 and the programme ”Research, Innovation and Digitalization for Smart Transformation” 2021-2027 (PRIDST) under grant agreement no. BG16RFPR002-1.014-0010-C01.




(a)

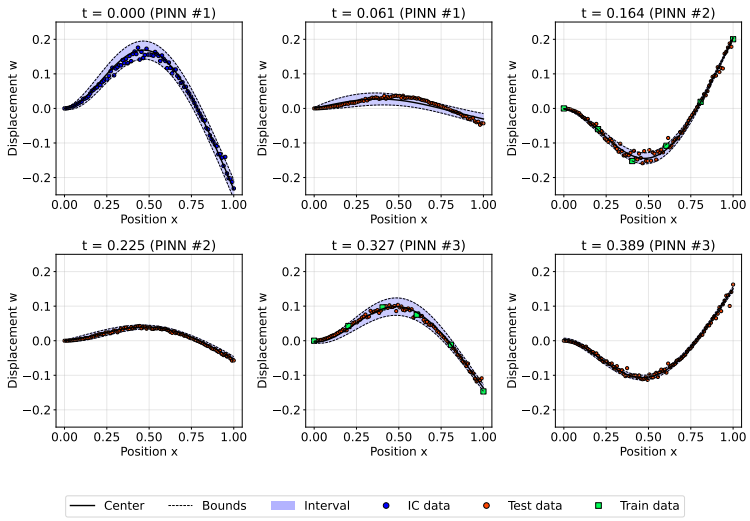



(b)

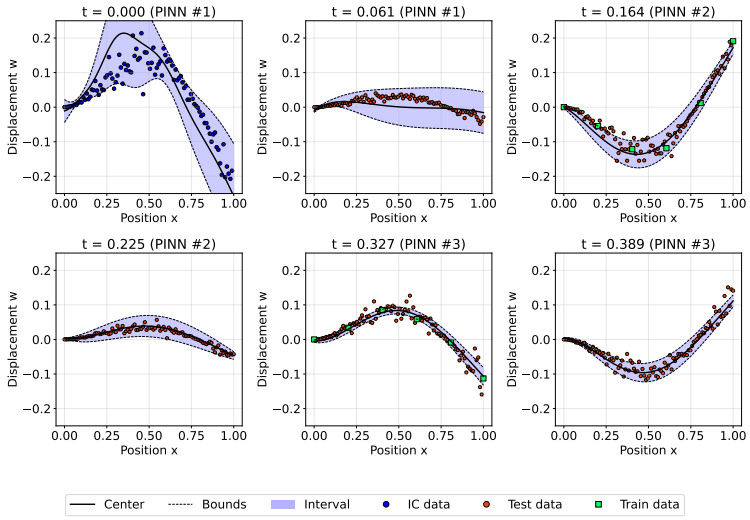



(c)

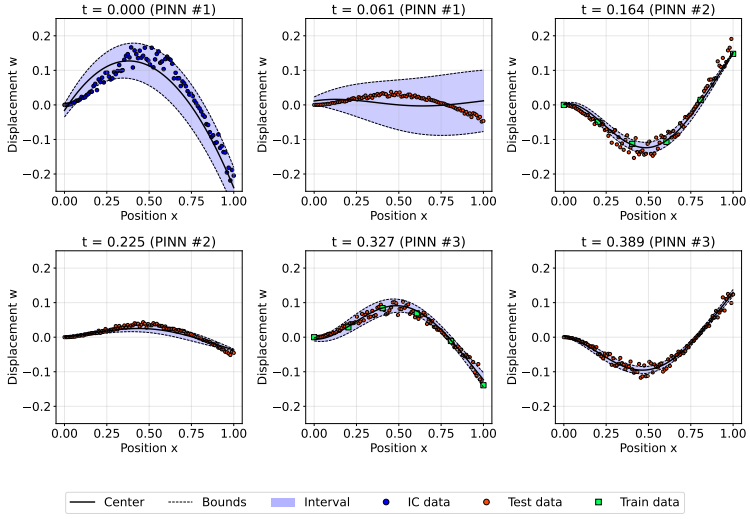
*Figure 2.* Heat map of the displacement, uncertainty, and training data for Interval PINN for mode 2 at noise level $\sigma = 0.25$ for noise sampled from (a) beta, (b) normal, and (c) uniform distribution.




(a)

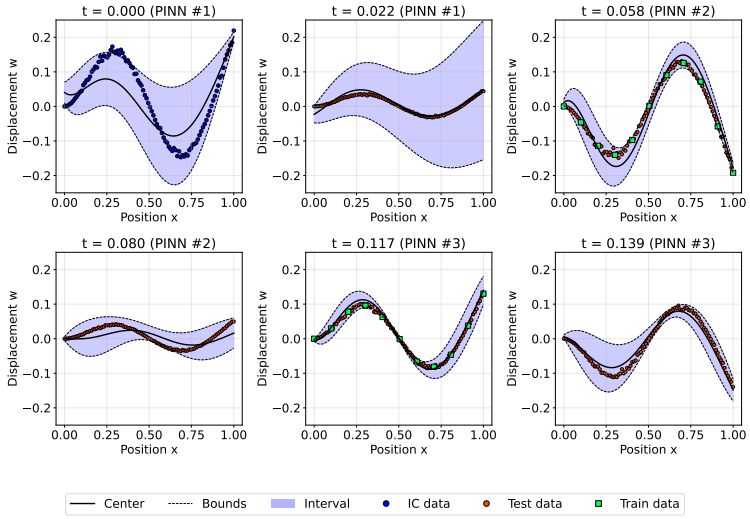



(b)

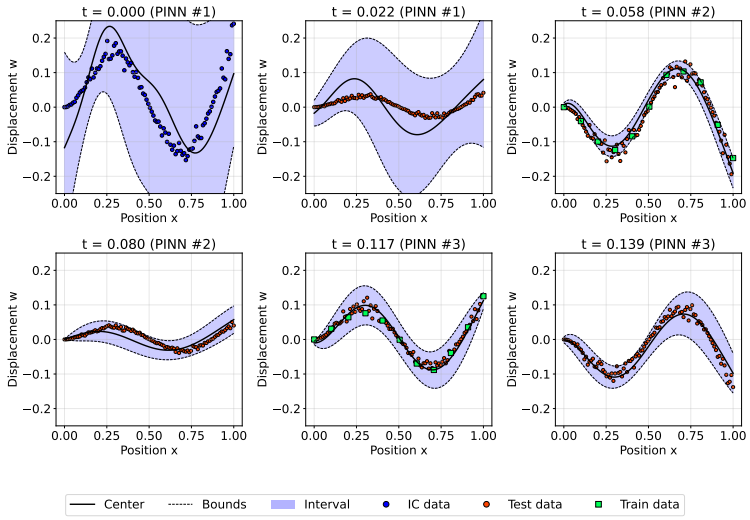



(c)

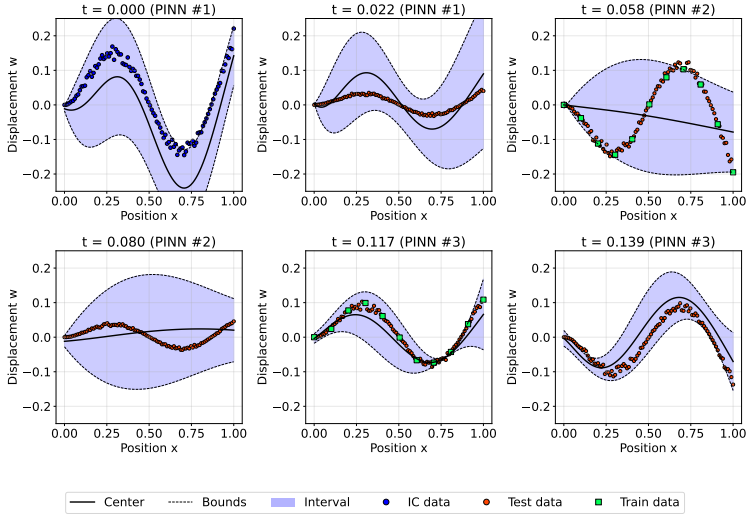
*Figure 3.* Interval centre and radius for Interval PINN for mode 2 at noise level $\sigma = 0.25$ for noise sampled from (a) beta, (b) normal, and (c) uniform distribution.




(a)




(b)




(c)

*Figure 4.* Interval centre and radius for Interval PINN for mode 3 at noise level $\sigma = 0.15$ for noise sampled from (a) beta, (b) normal, and (c) uniform distribution.

## References


Arbel, J., K. Pitas, M. Vladimirova, and V. Fortuin: 2023, 'A Primer on Bayesian Neural Networks: Review and Debates'. *ArXiv* **abs/2309.16314**.

Babaee, H., C. Bastidas, M. DeFilippo, C. Chryssostomidis, and G. E. Karniadakis: 2020, 'A Multifidelity Framework and Uncertainty Quantification for Sea Surface Temperature in the Massachusetts and Cape Cod Bays'. *Earth and Space Science* **7**, e2019EA000954.

Bishop, C. M.: 2009, *Pattern recognition and machine learning*. Springer Science + Business Media.

Ferson, S. and L. R. Ginzburg: 1996, 'Different methods are needed to propagate ignorance and variability'. *Reliability Engineering & System Safety* **54**(2), 133–144.

Forrester, A. I. J., A. Sobester, and A. J. Keane: 2008, *Engineering Design Via Surrogate Modelling: A Practical Guide*. J. Wiley.

Gal, Y. and Z. Ghahramani: 2015, 'Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning'. *33rd International Conference on Machine Learning, ICML 2016* **3**, 1651–1660.

Garczarczyk, Z.: 2000, 'Interval neural networks'. In: *2000 IEEE International Symposium on Circuits and Systems (ISCAS)*, Vol. 3. pp. 567–570 vol.3.

Hjelmstad, K. D.: 2025, *Euler–Lagrange Equations*, pp. 251–274. Cham: Springer Nature Switzerland.

Huang, L., B.-L. Zhang, and Q. Huang: 1998, 'Robust interval regression analysis using neural networks'. *Fuzzy Sets and Systems* **97**(3), 337–347.

Kiyani, E., K. Shukla, J. F. Urbán, J. Darbon, and G. E. Karniadakis: 2025, 'Optimizing the optimizer for physics-informed neural networks and Kolmogorov-Arnold networks'. *Computer Methods in Applied Mechanics and Engineering* **446**, 118308.

Krishnapriyan, A., A. Gholami, S. Zhe, R. Kirby, and M. W. Mahoney: 2021, 'Characterizing possible failure modes in physics-informed neural networks'. In: M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan (eds.): *Advances in Neural Information Processing Systems*, Vol. 34. pp. 26548–26560, Curran Associates, Inc.

Lee, J., K. Park, and W. Jung: 2024, 'Physics-Informed Neural Networks for Cantilever Dynamics and Fluid-Induced Excitation'. *Applied Sciences 2024, Vol. 14, Page 7002* **14**, 7002.

Lino, M., S. Fotiadis, A. A. Bharath, and C. D. Cantwell: 2023, 'Current and emerging deep-learning methods for the simulation of fluid dynamics'. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **479**.

Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala: 2019, 'PyTorch: An Imperative Style, High-Performance Deep Learning Library'. In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., pp. 8024–8035.

Raissi, M., P. Perdikaris, and G. Karniadakis: 2019, 'Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations'. *Journal of Computational Physics* **378**, 686–707.

Rasmussen, C. E. and C. K. I. Williams: 2006, *Gaussian processes for machine learning*. MIT Press.

Rylander, T., P. Ingelstrom, and A. Bondeson: 2013, *Computational Electromagnetics*. Springer.

Sadeghi, J., M. de Angelis, and E. Patelli: 2019, 'Efficient training of interval Neural Networks for imprecise training data'. *Neural Networks* **118**, 338–351.

Taflove, A., S. C. Hagness, and M. Piket-May: 2005, '9 - Computational Electromagnetics: The Finite-Difference Time-Domain Method'. In: W.-K. CHEN (ed.): *The Electrical Engineering Handbook*. Burlington: Academic Press, pp. 629–670.

Yang, L., X. Meng, and G. E. Karniadakis: 2021, 'B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data'. *Journal of Computational Physics* **425**, 109913.

Zhang, D., L. Lu, L. Guo, and G. E. Karniadakis: 2019, 'Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems'. *Journal of Computational Physics* **397**, 108850.