# Trustworthy Direct Interval Propagation via Conformal Prediction in Neural Network Surrogates

Ghifari Adam Faza
*LMSD, Department of Mechanical Engineering, KU Leuven, Heverlee 3001, Belgium, adam.faza@kuleuven.be*

Hans Hallez
*M-Group research group, Department of Computer Science, KU Leuven, Brugge, 8200, Belgium, hans.hallez@kuleuven.be*

David Moens
*LMSD, Department of Mechanical Engineering, KU Leuven, Heverlee 3001, Belgium, david.moens@kuleuven.be*

**Abstract.** Uncertainty propagation is essential for reliable engineering design, but when input distributions are unknown, or data are scarce, probabilistic models risk misspecification, making interval representations a more robust alternative. However, interval propagation requires solving optimisation problems for each bound, leading to high computational cost, especially in downstream tasks such as design optimisation or reliability analysis. To mitigate this, surrogate models can be designed to provide interval predictions directly via interval-valued regression, supported by recent neural-network-based methods and data augmentation strategies for constructing interval datasets. Yet these approaches typically ignore surrogate approximation error and data noise. We address this by extending interval regression with a conformal prediction framework, which is model-agnostic and provides user-controlled confidence levels to capture both input and model uncertainty in a distribution-free manner. Experimental results on analytical and PDE benchmark problems demonstrate that the proposed conformalised interval propagation significantly improves coverage reliability compared to uncalibrated predictors, while maintaining practical prediction efficiency. Furthermore, we show that calibration using approximate interval data remains effective when the approximation error is limited, highlighting the practical applicability of the proposed framework in data-scarce settings.

**Keywords:** interval propagation, conformal prediction, uncertainty quantification

## 1. Introduction

Uncertainty propagation is fundamental to engineering design, where variations in geometry, material properties, or environmental loading must be tracked through computational models. When input distributions are known, classical methods such as Monte Carlo simulation (Robert and Casella, 2004; Brevault et al., 2020) and quasi-Monte Carlo methods (Dick et al., 2013; Lemieux, 2009) are widely used. When data are scarce, however, prescribing precise distributions risks mis-

specification; interval representations offer a more robust alternative. For interval-valued inputs $\boldsymbol{x}^\dagger = [\boldsymbol{x}_L, \boldsymbol{x}_U]$, propagation through a model $\mathcal{M}$ reduces to two optimisation problems:

$$y_L = \min_{\boldsymbol{x}_L \leq \boldsymbol{x} \leq \boldsymbol{x}_U} \mathcal{M}(\boldsymbol{x}), \qquad y_U = \max_{\boldsymbol{x}_L \leq \boldsymbol{x} \leq \boldsymbol{x}_U} \mathcal{M}(\boldsymbol{x}). \tag{1}$$

This becomes computationally prohibitive in downstream tasks such as design optimisation or reliability analysis. Substantial efficiency gains arise if a surrogate predicts output interval bounds directly, bypassing repeated optimisation entirely.

Direct interval propagation via interval-valued regression (Roque et al., 2007) can be realised with neural network approaches including regularised neural networks (RANN) (Yang et al., 2019), interval neural networks (INN) (Oala et al., 2021; Betancourt and Muhanna, 2022; Tretiak et al., 2023; Wang et al., 2025), and bound-propagation methods (Gowal et al., 2019; Zhang et al., 2018). A key limitation is that these methods require interval-labelled training data, which are rarely available in practice; data augmentation (Faza et al., 2026) addresses this by constructing synthetic interval datasets from pointwise observations. A fundamental gap remains: these approaches account only for input uncertainty, assuming exact surrogates and noise-free data, whereas real-world settings introduce both approximation error and measurement noise. We address this by extending interval propagation within a conformal prediction framework (Papadopoulos et al., 2002), which is model-agnostic, distribution-free, and provides user-controlled coverage guarantees.

We propose a conformalised direct interval propagation methodology and evaluate it on standard neural network surrogates and operator-learning architectures (DeepONet (Lu et al., 2021)), across one-dimensional regression and PDE benchmarks under both ideal and data-scarce calibration scenarios (Faza et al., 2026). The main contributions are:

1. A conformal prediction algorithm that enhances the robustness of direct interval propagation for both standard neural surrogates and operator-learning surrogates such as DeepONet.

2. A demonstration of the proposed method's effectiveness through numerical experiments on both standard and PDE-based surrogate models.

The remainder is organised as follows. Section 2 reviews direct interval propagation. Section 3 presents the conformalised formulation. Section 4 reports numerical experiments under ideal and augmented calibration scenarios. Section 5 concludes the paper.

## 2. Direct Interval Propagation

Rather than solving the optimisation in Equation (1), direct interval propagation (DIP) learns a mapping from interval-valued inputs to interval-valued outputs directly:

$$\hat{f} : [\boldsymbol{x}_L, \boldsymbol{x}_U] \mapsto [\boldsymbol{y}_L, \boldsymbol{y}_U]. \tag{2}$$

We focus on two DIP strategies, RANN (Yang et al., 2019) and CROWN bound propagation (Zhang et al., 2018), which are the best-performing methods across both ideal-interval and pointwise-data training settings (Faza et al., 2026).

### 2.1. Regularised Artificial Neural Network

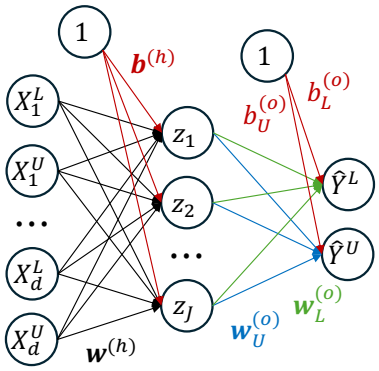



*Figure 1.* Regularised artificial neural network (RANN) architecture for Naïve direct interval propagation. The architecture consist of a standard neural network that doubles the number of input and output to accommodate interval-valued variable.

RANN uses the simple architecture illustrated in Figure 1. For interval-valued data $(\boldsymbol{X}^\dagger, Y^\dagger)$ with $d$-dimensional predictors in lower-upper form, the network has $2d$ inputs, $J$ hidden nodes, and 2 outputs, producing:

$$\hat{\boldsymbol{y}}_L = \sigma^{(o)}\left(\sum_{j=1}^{J} z_j w_{j,L}^{(o)} + b_L^{(o)}\right) \quad \textit{and} \quad \hat{\boldsymbol{y}}_U = \sigma^{(o)}\left(\sum_{j=1}^{J} z_j w_{j,U}^{(o)} + b_U^{(o)}\right), \tag{3}$$

where $z_j$ is the $j$th hidden-node output:

$$z_j = \sigma^{(h)}\left(\sum_{k=1}^{2d} X_k w_{k,j}^{(h)} + b_j^{(h)}\right). \tag{4}$$

Training uses MSE loss augmented with a non-crossing regulariser (Yang et al., 2019) to enforce $\hat{Y}_U \geq \hat{Y}_L$:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}(y_{i,L} - \hat{y}_{i,L})^2 + \frac{1}{N}\sum_{i=1}^{N}(y_{i,U} - \hat{y}_{i,U})^2 + \frac{\lambda}{N}\sum_{i=1}^{N}\left(\max\{0, \hat{y}_{i,L} - \hat{y}_{i,U}\}\right)^2, \tag{5}$$

where $\lambda \geq 0$ controls the non-crossing penalty strength; intermediate values balance crossing avoidance with prediction accuracy.

For DeepONet (Figure 2a), the solution operator $\mathcal{G}$ is:

$$\mathcal{G}(\boldsymbol{u})(\boldsymbol{x}) \approx \sum_{i=1}^{q} \beta_i(\boldsymbol{u}) \cdot \tau_i(\boldsymbol{x}), \tag{6}$$

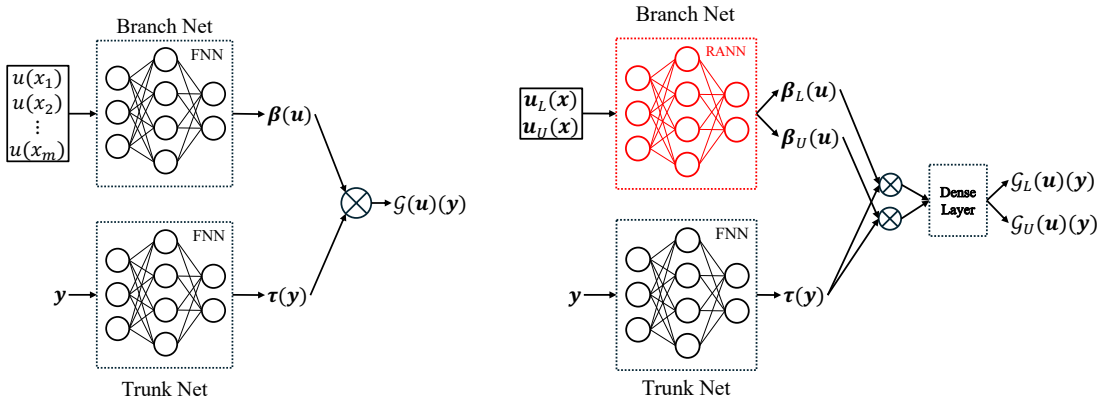



*Figure 2.* (a) Standard DeepONet architecture. (b) Modified DeepONet architecture with interval-valued branch net.

where $\boldsymbol{\tau} \in \mathbb{R}^q$ and $\boldsymbol{\beta} \in \mathbb{R}^q$ are the trunk-net and branch-net outputs.

Since the query location is known in practice, only the branch net is replaced by RANN while the trunk net remains a standard FNN (Figure 2b). The branch net latent representation becomes $2q$-dimensional: $\boldsymbol{\beta}^\dagger = [\beta_{1,L}, \beta_{1,U}, \ldots, \beta_{q,L}, \beta_{q,U}]$. With $\boldsymbol{\tau} \in \mathbb{R}^q$ and $\boldsymbol{\beta} \in \mathbb{R}^{2q}$, the outputs are combined via two dot products:

$$\boldsymbol{z}_L = \sum_{i=1}^{q} \beta_{i,L}(\boldsymbol{u}) \cdot \tau_i(\boldsymbol{x}), \quad \boldsymbol{z}_U = \sum_{i=1}^{q} \beta_{i,U}(\boldsymbol{u}) \cdot \tau_i(\boldsymbol{x}). \tag{7}$$

The results from the two multiplications are then concatenated, $\boldsymbol{z}^\dagger = [z_{1,L}, z_{1,U}, \ldots, z_{q,L}, z_{q,U}]$, and passed through a dense layer to produce output $\mathcal{G}_L(\boldsymbol{u})(\boldsymbol{y})$ and $\mathcal{G}_U(\boldsymbol{u})(\boldsymbol{y})$.

## 2.2. CROWN Bound Propagation

Computing certified output bounds under input perturbations is challenging because ReLU nonlinearities make exact bound propagation intractable for deep networks. The CROWN framework (Zhang et al., 2018) resolves this with a layer-wise linear relaxation that yields sound, tractable bounds, originally developed for adversarial robustness and applicable to uncertainty quantification in neural surrogate models.

Given an input domain $\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^{d_0} \mid \boldsymbol{x}_{0,L} \leq \boldsymbol{x} \leq \boldsymbol{x}_{0,U}\}$, CROWN propagates interval bounds through each layer of the network. For a layer $l$, assume the pre-activation satisfies $\hat{z}_{l,L} \leq \hat{z}_l \leq \hat{z}_{l,U}$. Because the ReLU activation $z_l = \max(0, \hat{z}_l)$ is nonlinear, CROWN replaces it with tight linear upper and lower relaxations over the interval $[\hat{z}_{l,L}, \hat{z}_{l,U}]$. The upper relaxation is given by

$$z_l \leq \frac{\hat{z}_{l,U}}{\hat{z}_{l,U} - \hat{z}_{l,L}} \left(\hat{z}_l - \hat{z}_{l,L}\right), \tag{8}$$

and the lower relaxation is

$$z_l \geq \boldsymbol{\alpha}_l \hat{z}_l, \qquad 0 \leq \boldsymbol{\alpha}_l \leq 1, \tag{9}$$

where $\boldsymbol{\alpha}_l$ defines the slope of the lower bound. Composing these relaxations across all layers yields certified global bounds on the network output:

$$f^L(\boldsymbol{x}) \leq \hat{f}(\boldsymbol{x}) \leq f^U(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \mathcal{X}, \tag{10}$$

where $f^L(\cdot)$ and $f^U(\cdot)$ are affine functions of $\boldsymbol{x}$, recursively defined by the relaxation parameters at each layer.

Adapting CROWN to DeepONet is straightforward via the `Auto_LiRPA` library (Xu et al., 2020), which provides a bound propagation wrapper compatible with the standard architecture without requiring any modifications.

Given $n$ interval-valued PDE input-output pairs $\{[\boldsymbol{u}_{i,L}, \boldsymbol{u}_{i,U}]\}_{i=1}^n$ and $\{[\boldsymbol{g}_{i,L}, \boldsymbol{g}_{i,U}]\}_{i=1}^n$, each input is converted to center-perturbation form for `Auto_LiRPA`:

$$\boldsymbol{u}_c = \frac{\boldsymbol{u}_U + \boldsymbol{u}_L}{2}, \quad \boldsymbol{u}_p = \frac{\boldsymbol{u}_U - \boldsymbol{u}_L}{2}, \tag{11}$$

We train the interval-aware DeepONet using two strategies, both receiving $[\boldsymbol{u}_c, \boldsymbol{u}_p]$ and outputting predicted bounds $[\hat{\boldsymbol{g}}_L, \hat{\boldsymbol{g}}_U]$.

The first strategy uses a *bound-based loss*, which directly fits the lower and upper interval endpoints:

$$\mathcal{L}_{bound} = MSE(\boldsymbol{g}_L, \hat{\boldsymbol{g}}_L) + MSE(\boldsymbol{g}_U, \hat{\boldsymbol{g}}_U) + \lambda \left(\max(0, \hat{\boldsymbol{g}}_L - \hat{\boldsymbol{g}}_U)^2\right), \tag{12}$$

where the final term enforces non-crossing of the predicted bounds.

The *midpoint-based loss* instead fits the interval midpoints $\boldsymbol{g}_M = \frac{1}{2}(\boldsymbol{g}_L + \boldsymbol{g}_U)$ and $\hat{\boldsymbol{g}}_M = \frac{1}{2}(\hat{\boldsymbol{g}}_L + \hat{\boldsymbol{g}}_U)$, offering more stability under noisy labels:

$$\mathcal{L}_{mid} = MSE(\boldsymbol{g}_M, \hat{\boldsymbol{g}}_M) + \lambda \left(\max(0, \hat{\boldsymbol{g}}_U - \hat{\boldsymbol{g}}_L)^2\right). \tag{13}$$

The bound-based loss targets tight endpoint matching, while the midpoint-based loss prioritises the central trend and is more stable under uncertain labels.

## 3. Conformal Prediction

Conformal prediction (CP) (Vovk et al., 1999; Vovk et al., 2005) is a model-agnostic, distribution-free framework that wraps any trained predictor to produce set-valued predictions with finite-sample coverage guarantees, under the exchangeability assumption. For $\boldsymbol{x} \in \mathbb{R}^d$ and $y \in \mathbb{R}$, it constructs a set-valued predictor

$$C : \boldsymbol{x} \to \{\text{subsets of } y\}, \tag{14}$$

which maps an input $\boldsymbol{x}$ to a subset of the response space. This prediction set is constructed such that, for a new unseen sample $(\boldsymbol{x}_*, y_*)$ drawn from the same distribution as the calibration data, the marginal coverage guarantee

$$\mathbb{P}(y_* \in C(\boldsymbol{x}_*)) \geq 1 - \alpha \tag{15}$$

holds for any finite sample size, where $\alpha \in (0, 1)$ is the user-defined miscoverage level. While the original *full* CP is computationally expensive (requiring repeated model retraining), *split* conformal

prediction (Papadopoulos et al., 2002; Papadopoulos, 2008) fits the model once on a training set and computes conformity scores on a held-out calibration set. We adopt the split framework throughout.

### 3.1. Split Conformal Prediction and Conformalised Quantile Regression

Data are split into $\mathcal{D}_t = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n_t}$ for training and $\mathcal{D}_c = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n_c}$ for calibration. A model $\hat{\mu}(\boldsymbol{x})$ is fitted on $\mathcal{D}_t$, and absolute-residual conformity scores are computed on $\mathcal{D}_c$:

$$r_i = |y_i - \hat{\mu}(\boldsymbol{x}_i)|. \tag{16}$$

For a given error level $\alpha$, we return the conformal correction $d$ as the $k$-th smallest value in $r_i$, where $k = \lceil (n/2+1)(1-\alpha) \rceil$. The conformal prediction of a new unobserved point $\boldsymbol{x}_*$ is then defined as:

$$C(\boldsymbol{x}_*) = \left[\hat{\mu}(\boldsymbol{x}_*) - d,\ \hat{\mu}(\boldsymbol{x}_*) + d\right]. \tag{17}$$

A variant of conformal prediction built upon quantile regression (Koenker and Bassett, 1978; Steinwart and Christmann, 2011), known as conformalised quantile regression (CQR) (Romano et al., 2019), follows a similar principle while leveraging conditional quantile estimates. Given a quantile regression model $\mathcal{M}_q$, two conditional quantile functions corresponding to lower and upper quantile levels $\alpha_{lo}$ and $\alpha_{hi}$ are first fitted using the training dataset:

$$\{\hat{q}_{\alpha_{lo}}, \hat{q}_{\alpha_{hi}}\} \leftarrow \mathcal{M}_q(\mathcal{D}_t). \tag{18}$$

Next, conformity scores are computed on the calibration dataset. For each calibration sample $(\boldsymbol{x}_i, y_i) \in \mathcal{D}_c$, the score is defined as

$$r_i = \max\left\{\hat{q}_{\alpha_{lo}}(\boldsymbol{x}_i) - y_i,\ y_i - \hat{q}_{\alpha_{hi}}(\boldsymbol{x}_i)\right\}. \tag{19}$$

This score is non-positive when $y_i$ lies within the predicted interval and equals the violation magnitude otherwise. The correction $d$ is the $k$-th smallest value of $\{r_i\}$, $k = \lceil (n/2+1)(1-\alpha) \rceil$, giving the conformal prediction interval:

$$C(\boldsymbol{x}_*) = \left[\hat{q}_{\alpha_{lo}}(\boldsymbol{x}_i) - d,\ \hat{q}_{\alpha_{hi}}(\boldsymbol{x}_i) + d\right]. \tag{20}$$

For operator learning (Moya et al., 2025), CQR extends to DeepONet by evaluating the score pointwise. Given quantile-DeepONet estimates $\hat{g}_{\alpha_{\mathrm{lo}}}$, $\hat{g}_{\alpha_{\mathrm{hi}}}$ and calibration set $\mathcal{D}_c = \{(u_i, g_i)\}_{i=1}^{n_c}$, the score at each domain location is:

$$r_i(x) = \max\left\{\hat{g}_{\alpha_{\mathrm{lo}}}(u_i)(x) - g_i(x),\ g_i(x) - \hat{g}_{\alpha_{\mathrm{hi}}}(u_i)(x)\right\}, \quad x \in \mathcal{X}. \tag{21}$$

The pointwise correction $d(x)$ is the empirical $(1-\alpha)$-quantile of $\{r_i(x)\}_{i=1}^{n_c}$, giving the conformal band for a new input $u_*$:

$$C(u_*)(x) = \left[\hat{g}_{\alpha_{\mathrm{lo}}}(u_*)(x) - d(x),\ \hat{g}_{\alpha_{\mathrm{hi}}}(u_*)(x) + d(x)\right], \quad x \in \mathcal{X}. \tag{22}$$

This formulation mirrors the standard CQR procedure while extending coverage guarantees to operator-valued predictions.

### 3.2. Conformalised Direct Interval Propagation

Applying CP to DIP follows the CQR formulation (Romano et al., 2019; Moya et al., 2025). Given a DIP model $\mathcal{M}_{\mathcal{I}}$ trained on interval-valued data $\mathcal{D}_t$ (Faza et al., 2026), the conformity score for each calibration sample $(\boldsymbol{x}_i^\dagger, y_i^\dagger) \in \mathcal{D}_c$ is:

$$r_i = \max\left\{\hat{y}_L(\boldsymbol{x}_i) - y_L,\ y_U - \hat{y}_U(\boldsymbol{x}_i)\right\}. \tag{23}$$

Then, the correction $d$ is the $k$-th smallest value in $r_i$, where $k = \lceil (n/2+1)(1-\alpha) \rceil$ The conformal prediction of a new unobserved point $\boldsymbol{x}_*^\dagger$ is then defined as:

$$C(\boldsymbol{x}_*^\dagger) = \left[\hat{y}_L(\boldsymbol{x}_i^\dagger) - d,\ \hat{y}_U(\boldsymbol{x}_i^\dagger) + d\right]. \tag{24}$$

Adapted from Equation 19, this score captures the worst-case interval coverage violation: it is positive when the predicted interval fails to enclose the true interval and non-positive when coverage is already satisfied. The correction $d$ is chosen as the $k$-th smallest conformity score, $k = \lceil (n/2+1)(1-\alpha) \rceil$, ensuring that at least a fraction $(1-\alpha)$ of calibration intervals are enclosed and guaranteeing marginal coverage for future predictions. Very small $\alpha$ (e.g., $\alpha = 0$) drives $d$ to the maximum score, producing overly conservative intervals; $\alpha = 0.05$ is a typical practical choice.

The same approach extends to interval DeepONets: the score in Equation 23 is computed pointwise over the domain and the correction $d$ is applied identically to the CQR-DeepONet case.

## 4. Numerical experiments

We extend the experimental design of (Faza et al., 2026) to reflect realistic data conditions: model training is restricted to intervals augmented from pointwise data, since ideal interval datasets are rarely available in practice. We evaluate two calibration scenarios — (i) ideal interval-valued calibration data (best-case) and (ii) augmented interval calibration data (fully practical) — and vary the calibration set size to assess its effect on prediction quality. Each scenario is evaluated on a one-dimensional regression task and a one-dimensional PDE.

Prediction quality is measured using predicted interval normalized average width (PINAW), predicted interval coverage probability (PICP), and the coverage width-based criterion (CWC):

$$\text{PINAW} = \frac{1}{N}\sum_{i=1}^{N}\frac{(\hat{y}_U - \hat{y}_L)}{(y_U - y_L)}, \tag{25}$$

$$\text{PICP} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{1}\left(\hat{y}_L \le y_i \le \hat{y}_U\right), \tag{26}$$

$$\text{CWC} = \text{PINAW}\left(1 + e^{\gamma \max(0,(1-\delta)-\text{PICP})}\right), \tag{27}$$

where $\mathbf{1}(\cdot)$ is an indicator function, $\delta$ is the desired coverage level, and $\gamma = 5$ balances the coverage-width trade-off. Since we evaluate against interval-valued targets, we replace the standard PICP with a continuous overlap-based version:

$$\text{PICP} = \frac{1}{N}\sum_{i=1}^{N} \frac{\max(0, \min(\hat{y}_U, y_U) - \max(\hat{y}_L, y_L))}{(y_U - y_L)}. \tag{28}$$

### 4.1. Calibration with ideal interval data

#### 4.1.1. *1-dimensional regression*

Consider a simple one-dimensional regression problem:

$$y = \sin(2x)e^{-x} + 1 + \varepsilon,\ x \in [0, \pi], \tag{29}$$

where $\varepsilon$ is a normally distributed aleatoric noise with mean $\mu = 0$ and standard deviation $\sigma = 0.025$. We create interval inputs with varying interval widths across the sample space. Since the actual function is cheap to evaluate, we propagate the inputs by evaluating on the actual function.

Table I. One-dimensional regression with ideal interval data calibration performance comparison across methods and calibration set sizes on three different metrics: PINAW, PICP, and CWC.

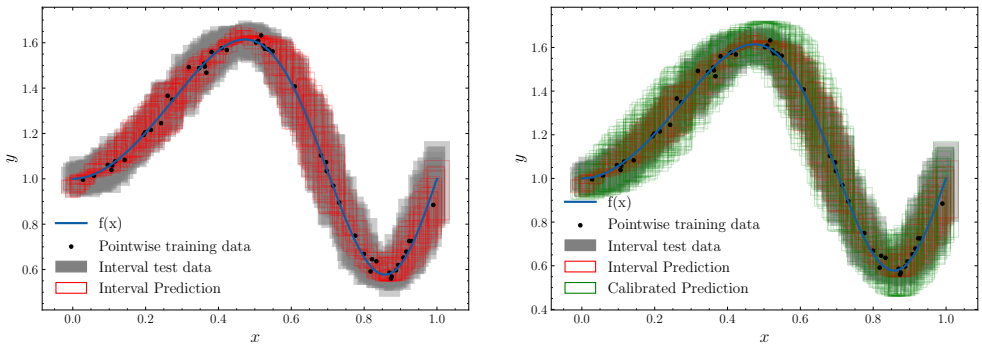
| $n_{\text{calib}}$ | **Metric** | **Naive** | **CROWN** | **Mid-CROWN** |
|---|---|---|---|---|
| 0 | PINAW | $0.587 \pm 0.111$ | $0.427 \pm 0.025$ | $0.533 \pm 0.026$ |
| | PICP | $0.551 \pm 0.068$ | $0.392 \pm 0.036$ | $0.515 \pm 0.027$ |
| | CWC | $6.117 \pm 0.901$ | $9.417 \pm 1.175$ | $6.572 \pm 0.558$ |
| 5 | PINAW | $1.767 \pm 0.511$ | $1.915 \pm 0.396$ | $1.682 \pm 0.355$ |
| | PICP | $0.959 \pm 0.033$ | $0.959 \pm 0.048$ | $0.986 \pm 0.019$ |
| | CWC | $3.899 \pm 0.895$ | $4.271 \pm 0.795$ | $3.468 \pm 0.608$ |
| 10 | PINAW | $2.252 \pm 0.476$ | $2.660 \pm 1.242$ | $1.600 \pm 0.245$ |
| | PICP | $0.989 \pm 0.010$ | $0.992 \pm 0.006$ | $0.991 \pm 0.010$ |
| | CWC | $4.625 \pm 0.910$ | $5.417 \pm 2.505$ | $3.267 \pm 0.443$ |
| 25 | PINAW | $2.837 \pm 0.817$ | $3.099 \pm 1.166$ | $2.070 \pm 0.733$ |
| | PICP | $0.997 \pm 0.003$ | $0.996 \pm 0.004$ | $0.997 \pm 0.004$ |
| | CWC | $5.707 \pm 1.635$ | $6.259 \pm 2.367$ | $4.180 \pm 1.544$ |
| 50 | PINAW | $2.339 \pm 0.675$ | $2.612 \pm 0.478$ | $1.854 \pm 0.359$ |
| | PICP | $0.996 \pm 0.005$ | $0.995 \pm 0.007$ | $0.998 \pm 0.002$ |
| | CWC | $4.723 \pm 1.330$ | $5.284 \pm 0.930$ | $3.727 \pm 0.713$ |

We fix the training set to 50 samples and vary the calibration set size, repeating each configuration ten times with different random seeds. Comparisons against optimisation-based propagation

are omitted, as (Faza et al., 2026) already establishes that DIP achieves comparable or better performance.

Table I shows a clear PICP improvement after calibration even with few calibration samples, confirming that conformal calibration compensates for the base predictor's undercoverage at the cost of increased PINAW. Conformal calibration guarantees coverage but not tightness: the interval width is set by the empirical quantile of calibration conformity scores, so outliers can inflate the correction substantially, an effect amplified by small calibration sets.

Figure 3a illustrates the undercoverage behaviour of the uncalibrated model, where the predicted intervals fail to fully enclose several ground-truth responses. In contrast, Figure 3b shows that, after conformal calibration, the prediction intervals are expanded sufficiently to enclose the previously missed observations, thereby correcting the undercoverage at the cost of increased interval width.




(a) (b)

*Figure 3.* Illustration of one-dimensional interval predictions comparing (a) the uncalibrated model and (b) the conformal-calibrated model.

With very small calibration sets (e.g., $n_{\text{calib}} = 5$), achievable coverage levels are restricted to multiples of $1/(n_{\text{calib}} + 1)$, making the 0.95 target unattainable; the procedure selects the next feasible level, yielding conservative intervals. Nevertheless, conformalisation consistently improves prediction reliability even with limited calibration data.

#### 4.1.2. *1-dimensional PDE*

We consider the second-order steady-state stochastic Poisson equation in one dimension as our first test case. The stochastic differential equation (SDE) is formulated as:

$$\begin{aligned} \tfrac{d^2 g(x)}{dx^2} &= 20u(x), \;\; x \in [0,1], \\ g(x=0) &= g(x=1) = 0, \\ u(x) &\sim \mathcal{GP}(m(x), k(x,x')), \end{aligned}$$

where $u(x)$ is the spatially random forcing function, modelled as a Gaussian process ($\mathcal{GP}$) with mean $m(x)$ set to be 0, and the radial basis function kernel $k(x,x') = \exp\left(-(x-x')^2/(2l^2)\right)$ where the kernel length scale of the GP is set to be $l = 0.1$. The HF dataset is generated using a grid size $\Delta x = 0.01$, while the LF dataset grid size is $\Delta x = 0.1$.

Table II. One-dimensional PDE problem with ideal interval data calibration performance comparison across methods and calibration set sizes.

| $n_{\text{calib}}$ | **Metric** | **Naive** | **CROWN** | **Mid-CROWN** |
|---|---|---|---|---|
| 0 | PINAW | $0.943 \pm 0.852$ | $0.227 \pm 0.227$ | $1.627 \pm 2.179$ |
| | PICP | $0.359 \pm 0.247$ | $0.049 \pm 0.077$ | $0.787 \pm 0.242$ |
| | CWC | $37.103 \pm 65.155$ | $23.965 \pm 24.916$ | $12.035 \pm 22.830$ |
| 10 | PINAW | $10.602 \pm 7.689$ | $40.337 \pm 30.477$ | $5.777 \pm 5.368$ |
| | PICP | $0.996 \pm 0.026$ | $0.983 \pm 0.074$ | $0.995 \pm 0.037$ |
| | CWC | $21.535 \pm 16.072$ | $88.779 \pm 84.591$ | $12.408 \pm 18.623$ |
| 25 | PINAW | $11.915 \pm 8.647$ | $47.454 \pm 35.038$ | $5.182 \pm 3.472$ |
| | PICP | $0.998 \pm 0.015$ | $0.996 \pm 0.031$ | $0.998 \pm 0.019$ |
| | CWC | $23.945 \pm 17.474$ | $96.542 \pm 73.896$ | $10.484 \pm 8.377$ |
| 50 | PINAW | $9.889 \pm 7.097$ | $40.576 \pm 29.677$ | $4.702 \pm 3.880$ |
| | PICP | $0.996 \pm 0.023$ | $0.989 \pm 0.056$ | $0.996 \pm 0.028$ |
| | CWC | $20.113 \pm 15.436$ | $86.272 \pm 74.724$ | $9.702 \pm 9.117$ |
| 100 | PINAW | $8.659 \pm 6.161$ | $35.926 \pm 26.073$ | $4.887 \pm 5.543$ |
| | PICP | $0.991 \pm 0.032$ | $0.977 \pm 0.083$ | $0.991 \pm 0.047$ |
| | CWC | $17.824 \pm 13.265$ | $83.020 \pm 88.582$ | $11.073 \pm 21.682$ |
| 250 | PINAW | $8.621 \pm 6.099$ | $35.475 \pm 25.712$ | $4.780 \pm 4.505$ |
| | PICP | $0.992 \pm 0.031$ | $0.976 \pm 0.084$ | $0.989 \pm 0.051$ |
| | CWC | $17.741 \pm 13.930$ | $82.224 \pm 88.481$ | $10.903 \pm 18.804$ |

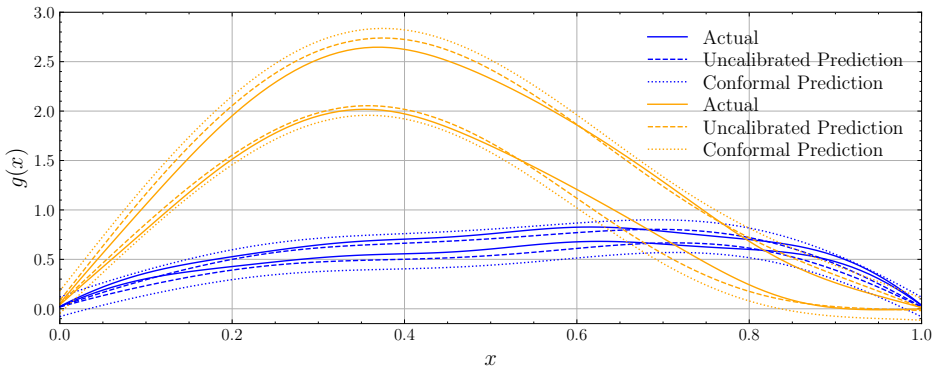



*Figure 4.* Illustration of one-dimensional PDE interval predictions for multiple problem instances. Each colour represents a different PDE instance characterised by a distinct input function $u(x)$.

Using 500 training samples (other settings follow (Faza et al., 2026)), Table II and Figure 4 confirm the same pattern as the regression case: conformalisation consistently improves PICP across all methods at the cost of increased PINAW.

Naive and CROWN exhibit poor uncalibrated performance, requiring large corrections that yield wide post-calibration intervals. Mid-CROWN starts with the highest initial coverage but also sees a substantial increase in interval width after calibration. Note that the overlap-based PICP (Equation 28) counts partial overlaps as coverage, so observed values can exceed the nominal 0.95 target; $\alpha$ can be tuned to balance coverage and width.

### 4.2. Calibration with augmented pointwise data

#### 4.2.1. *1-dimensional regression*

We repeat the setup of Section 4.1.1 (Equation 29), replacing ground-truth calibration data with augmented intervals from the same approximation procedure as training. Since both training and calibration use the same approximated source, the coverage guarantee holds with respect to the augmented distribution rather than the true one; systematic discrepancies between the two may cause under- or over-coverage.

Table III. One-dimensional regression with augmented interval data calibration performance comparison across methods and calibration set sizes.

| $n_{\text{calib}}$ | **Metric** | **Naive** | **CROWN** | **Mid-CROWN** |
|---|---|---|---|---|
| 0 | PINAW | $0.623 \pm 0.097$ | $0.430 \pm 0.014$ | $0.537 \pm 0.033$ |
| | PICP | $0.519 \pm 0.084$ | $0.398 \pm 0.022$ | $0.522 \pm 0.034$ |
| | CWC | $7.738 \pm 1.831$ | $9.210 \pm 0.930$ | $6.442 \pm 0.637$ |
| 5 | PINAW | $1.612 \pm 0.386$ | $1.417 \pm 0.743$ | $0.945 \pm 0.260$ |
| | PICP | $0.938 \pm 0.103$ | $0.832 \pm 0.126$ | $0.794 \pm 0.163$ |
| | CWC | $3.809 \pm 0.558$ | $4.575 \pm 1.382$ | $3.880 \pm 1.949$ |
| 10 | PINAW | $1.608 \pm 0.315$ | $1.222 \pm 0.265$ | $1.009 \pm 0.240$ |
| | PICP | $0.955 \pm 0.024$ | $0.865 \pm 0.062$ | $0.840 \pm 0.159$ |
| | CWC | $3.606 \pm 0.529$ | $3.601 \pm 0.360$ | $3.550 \pm 1.650$ |
| 25 | PINAW | $2.107 \pm 0.553$ | $1.447 \pm 0.203$ | $1.181 \pm 0.170$ |
| | PICP | $0.979 \pm 0.016$ | $0.930 \pm 0.026$ | $0.924 \pm 0.049$ |
| | CWC | $4.412 \pm 1.027$ | $3.496 \pm 0.358$ | $2.916 \pm 0.302$ |
| 50 | PINAW | $1.941 \pm 0.363$ | $1.635 \pm 0.604$ | $1.273 \pm 0.213$ |
| | PICP | $0.974 \pm 0.013$ | $0.934 \pm 0.026$ | $0.939 \pm 0.033$ |
| | CWC | $4.142 \pm 0.710$ | $3.869 \pm 1.207$ | $2.975 \pm 0.257$ |

Table III shows that augmented calibration yields slightly lower PICP than ideal calibration, but the degradation remains limited. The method's effectiveness depends on how closely the augmented intervals approximate the true ones: small discrepancies allow the augmented correction to serve as a reliable proxy, while large systematic bias can cause coverage failures.

#### 4.2.2. *1-dimensional PDE*

Following the setup of Section 4.1.2 with augmented calibration data, Table IV shows a divergence between methods: Mid-CROWN's interval width decreases after calibration, reducing PICP, while Naive and CROWN receive larger corrections and become more conservative. As shown in Figure 5, augmented output intervals underestimate the true interval width, so conformity scores are computed against a narrower reference — insufficient correction for Mid-CROWN but still adequate for the weaker methods. These results confirm that augmented calibration inherits the bias of the augmented distribution and may not reliably guarantee true-interval coverage when systematic discrepancies are present.

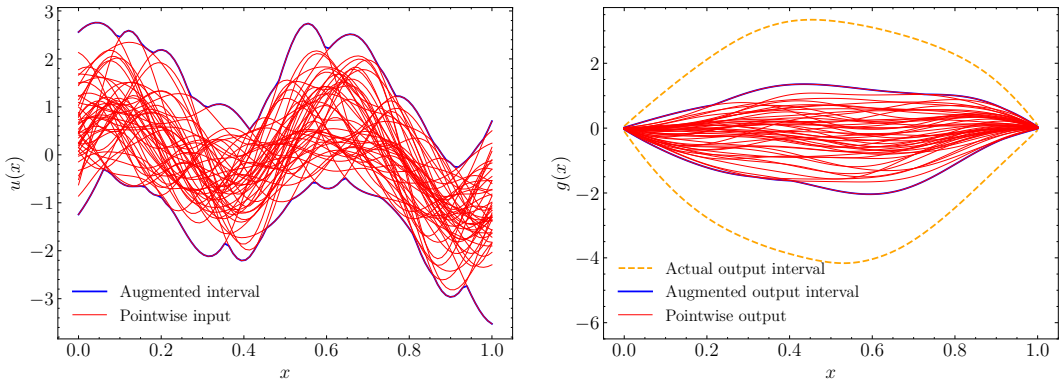



*Figure 5.* Augmented input function (left) and comparison between the augmented output and the ground truth output propagated through PDE solver.

## 5. Conclusion

We propose a conformalised direct interval propagation framework that combines DIP — which efficiently predicts output interval bounds directly — with split conformal prediction to provide finite-sample coverage guarantees without modifying the underlying model or assuming a data distribution. By calibrating conformity scores on interval-valued data, the method achieves reliable marginal coverage while preserving computational efficiency.

Evaluated on regression and PDE surrogate tasks using both ideal and augmented interval calibration data, conformal calibration consistently improves coverage across DIP methods, accompanied by increased interval width due to the coverage-efficiency trade-off. Augmented calibration remains effective when the discrepancy from true intervals is small, but systematic bias can cause undercoverage or overly conservative predictions. These findings establish conformalised DIP as a practical and theoretically grounded framework for uncertainty quantification in interval-valued neural network surrogates, particularly under data-scarce conditions.

Table IV. One-dimensional PDE problem with augmented interval data calibration performance comparison across methods and calibration set sizes.

| $n_{\text{calib}}$ | **Metric** | **Naive** | **CROWN** | **Mid-CROWN** |
|---|---|---|---|---|
| 0 | PINAW | $1.032 \pm 1.010$ | $0.236 \pm 0.225$ | $3.163 \pm 8.135$ |
| | PICP | $0.328 \pm 0.260$ | $0.053 \pm 0.084$ | $0.817 \pm 0.224$ |
| | CWC | $52.083 \pm 94.153$ | $23.569 \pm 22.457$ | $13.937 \pm 32.887$ |
| 10 | PINAW | $8.907 \pm 7.040$ | $36.995 \pm 28.804$ | $1.419 \pm 0.885$ |
| | PICP | $0.976 \pm 0.072$ | $0.969 \pm 0.103$ | $0.774 \pm 0.268$ |
| | CWC | $19.910 \pm 21.883$ | $89.016 \pm 101.695$ | $37.004 \pm 156.746$ |
| 25 | PINAW | $11.308 \pm 9.108$ | $44.356 \pm 33.681$ | $2.647 \pm 3.779$ |
| | PICP | $0.990 \pm 0.043$ | $0.990 \pm 0.056$ | $0.912 \pm 0.141$ |
| | CWC | $23.524 \pm 20.934$ | $93.429 \pm 79.918$ | $9.249 \pm 22.976$ |
| 50 | PINAW | $8.691 \pm 6.334$ | $36.938 \pm 28.041$ | $1.123 \pm 0.202$ |
| | PICP | $0.984 \pm 0.054$ | $0.973 \pm 0.096$ | $0.687 \pm 0.258$ |
| | CWC | $18.549 \pm 16.576$ | $87.277 \pm 97.712$ | $19.310 \pm 44.099$ |
| 100 | PINAW | $7.651 \pm 5.503$ | $32.086 \pm 23.578$ | $2.229 \pm 3.042$ |
| | PICP | $0.975 \pm 0.069$ | $0.959 \pm 0.116$ | $0.491 \pm 0.285$ |
| | CWC | $17.107 \pm 16.888$ | $84.255 \pm 109.276$ | $98.950 \pm 194.710$ |
| 250 | PINAW | $7.401 \pm 5.277$ | $30.945 \pm 22.569$ | $1.363 \pm 0.820$ |
| | PICP | $0.971 \pm 0.078$ | $0.954 \pm 0.122$ | $0.477 \pm 0.284$ |
| | CWC | $17.332 \pm 22.383$ | $83.917 \pm 111.922$ | $83.304 \pm 163.789$ |

## Data Availability

The code are available at: `https://github.com/fazaghifari/IntervalOperatorLearning`. Although, the provided example in the repository are not the exact experiments presented in this paper, they illustrate the implementation of the conformal calibration discussed herein.

## Acknowledgements

This work was supported by the granted H2020 FETOPEN-2018-2019-2020-01 European project, *Epistemic AI* under grant agreement No. 964505 (E-pi). The authors acknowledge the use of AI tools, such as ChatGPT and Grammarly, to assist with grammar refinement and improve the clarity of wording in the manuscript. All substantive content, analysis, and conclusions are the original work of the authors.

## References


Betancourt, D. and R. L. Muhanna: 2022, 'Interval deep learning for computational mechanics problems under input uncertainty'. *Probabilistic Engineering Mechanics* **70**, 103370.

Brevault, L., M. Balesdent, and J. Morio: 2020, *Aerospace System Analysis and Optimization in Uncertainty*. Springer International Publishing.

Dick, J., F. Y. Kuo, and I. H. Sloan: 2013, 'High-dimensional integration: The quasi-Monte Carlo way'. *Acta Numerica* **22**, 133–288.

Faza, G. A., J. Wauters, F. Cuzzolin, H. Hallez, and D. Moens: 2026, 'Direct interval propagation methods using neural-network surrogates for uncertainty quantification in physical systems surrogate model'. *Knowledge-Based Systems* **341**, 115824.

Gowal, S., K. Dvijotham, R. Stanforth, R. Bunel, C. Qin, J. Uesato, R. Arandjelovic, T. Mann, and P. Kohli: 2019, 'On the Effectiveness of Interval Bound Propagation for Training Verifiably Robust Models'.

Koenker, R. and G. Bassett: 1978, 'Regression Quantiles'. *Econometrica* **46**(1), 33.

Lemieux, C.: 2009, *Monte Carlo and Quasi-Monte Carlo Sampling*. Springer New York.

Lu, L., P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis: 2021, 'Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators'. *Nature Machine Intelligence* **3**(3), 218–229.

Moya, C., A. Mollaali, Z. Zhang, L. Lu, and G. Lin: 2025, 'Conformalized-DeepONet: A distribution-free framework for uncertainty quantification in deep operator networks'. *Physica D: Nonlinear Phenomena* **471**, 134418.

Oala, L., C. Heiß, J. Macdonald, M. März, G. Kutyniok, and W. Samek: 2021, 'Detecting failure modes in image reconstructions with interval neural network uncertainty'. *International Journal of Computer Assisted Radiology and Surgery* **16**(12), 2089–2097.

Papadopoulos, H.: 2008, *Inductive Conformal Prediction: Theory and Application to Neural Networks*, Chapt. 1. InTech.

Papadopoulos, H., K. Proedrou, V. Vovk, and A. Gammerman: 2002, *Inductive Confidence Machines for Regression*, p. 345–356. Springer Berlin Heidelberg.

Robert, C. P. and G. Casella: 2004, *Monte Carlo Statistical Methods*. Springer New York.

Romano, Y., E. Patterson, and E. J. Candès: 2019, 'Conformalized Quantile Regression'.

Roque, A. M. S., C. Mate, J. Arroyo, and A. Sarabia: 2007, 'iMLP: Applying Multi-Layer Perceptrons to Interval-Valued Data'. *Neural Processing Letters* **25**(2), 157–169.

Steinwart, I. and A. Christmann: 2011, 'Estimating conditional quantiles with the help of the pinball loss'. *Bernoulli* **17**(1).

Tretiak, K., G. Schollmeyer, and S. Ferson: 2023, 'Neural network model for imprecise regression with interval dependent variables'. *Neural Networks* **161**, 550–564.

Vovk, V., A. Gammerman, and C. Saunders: 1999, 'Machine-Learning Applications of Algorithmic Randomness'. In: *Proceedings of the Sixteenth International Conference on Machine Learning*. San Francisco, CA, USA, p. 444–453, Morgan Kaufmann Publishers Inc.

Vovk, V., A. Gammerman, and G. Shafer: 2005, *Algorithmic learning in a random world*. Springer Science+Business Media.

Wang, K., K. Shariatmadar, S. K. Manchingal, F. Cuzzolin, D. Moens, and H. Hallez: 2025, 'CreINNs: Credal-Set Interval Neural Networks for Uncertainty Estimation in Classification Tasks'. *Neural Networks* **185**, 107198.

Xu, K., Z. Shi, H. Zhang, Y. Wang, K.-W. Chang, M. Huang, B. Kailkhura, X. Lin, and C.-J. Hsieh: 2020, 'Automatic Perturbation Analysis for Scalable Certified Robustness and Beyond'.

Yang, Z., D. K. Lin, and A. Zhang: 2019, 'Interval-valued data prediction via regularized artificial neural network'. *Neurocomputing* **331**, 336–345.

Zhang, H., T.-W. Weng, P.-Y. Chen, C.-J. Hsieh, and L. Daniel: 2018, 'Efficient Neural Network Robustness Certification with General Activation Functions'.