



Evidential Physics-Informed Neural Networks

Hai Siong Tan¹, Kuancheng Wang², and Rafe McBeth³

¹Gryphon Center for Artificial Intelligence and Theoretical Sciences, Singapore

²Georgia Institute of Technology, Atlanta, GA, USA

³University of Pennsylvania, Perelman School of Medicine, Department of Radiation Oncology, PA, USA
haisiong.tan@gryphonai.com.sg

Abstract

We present a novel class of Physics-Informed Neural Networks that is formulated based on the principles of Evidential Deep Learning, where the model incorporates uncertainty quantification by learning parameters of a higher-order distribution. The dependent and trainable variables of the PDE residual loss and data-fitting loss terms are recast as functions of the hyperparameters of an evidential prior distribution. Our model is equipped with an information-theoretic regularizer that contains the Kullback-Leibler divergence between two inverse-gamma distributions characterizing predictive uncertainty. Relative to Bayesian-Physics-Informed-Neural-Networks, our framework appeared to exhibit higher sensitivity to data noise, preserve boundary conditions more faithfully and yield empirical coverage probabilities closer to nominal ones. Toward examining its relevance for data mining in scientific discoveries, we demonstrate how to apply our model to inverse problems involving 1D and 2D nonlinear differential equations.

Keywords Physics-Informed-Neural-Networks · uncertainty quantification

1. Introduction

Physics-Informed Neural Networks (PINNs) furnish a class of scientific machine learning frameworks that can harness observational data and the universal approximation capability of neural networks [16] to solve partial differential equations (PDE) and their related inverse problems. It is essentially characterized by a linear combination of loss functions formulated such that they drive the model towards fitting the empirical datasets while also adhering to the governing PDEs (equipped with some set of boundary/initial conditions). Over the years, PINNs have shown excellent results in terms of solving not only forward problems [22, 25, 26], but also inverse ones [8, 14, 17, 19, 30] where we use PINNs to determine unknown scientific parameters in the PDEs that are assumed to model the observational data, enabling deep learning

techniques to be used as a powerful tool for scientific discovery.

In this paper, we present a novel form of PINNs designed such that uncertainty quantification [1] is naturally embedded within the model architecture and its loss function. In our formulation, uncertainty estimates of both the dependent variables of the PDE and its unknown parameters are obtained immediately upon completion of model training which is implemented in a typical gradient-descent based approach. This is in contrast to other forms of methods such as Bayesian-based techniques which rely on Monte-Carlo sampling to estimate output variance. Our model is largely inspired by the principles of *Evidential Deep Learning* (EDL) introduced in the seminal works [2, 27] where an *evidential*, higher-order set of priors are placed over likelihood functions for the outputs, and model training is then used to in-

for hyperparameters of this distribution. Uncertainty estimates can be deduced straightforwardly as they are closed-form functions of the learned hyperparameters. To our knowledge, although EDL has been implemented in various types of model architectures (e.g. U-Nets [18, 31], FNN ([2]), LeNET [27]) it hasn't been synthesized with PINNs. In our work here, we propose how a hybrid of PINN and EDL principles can be performed, and present a novel information-theoretic regularizer for EDL that is also applicable for more general regression tasks.

As reviewed in [8], one of the most recent state-of-the-art proposals for uncertainty quantification in PINNs is the highly interesting Bayesian-PINN (B-PINN) of Yang et al. in [29]. Thus, for comparison, we validated our model against the inverse PDE problems studied in [29]. Specifically, we applied our model to 1D and 2D nonlinear reaction-diffusion-type systems with the training datasets being doped with heterogeneous noise. We found that relative to B-PINN, our model inherited uncertainty estimates which were significantly more sensitive to the added data noise, and correlated well with prediction errors for out-of-distribution data. It adhered to the boundary conditions more faithfully in the examples we examined relative to other frameworks. We call our class of models *Evidential PINN* (or E-PINN for short), and hope that it adds to the uncertainty quantification toolkit for scientific machine learning at large.

2. Theoretical basis of our model

A distinctive feature of PINNs is that the PDE residual is included as an essential component of the loss function, guiding the model training towards adhering to the PDE in an unsupervised manner. Consider the case where the PDE contains an unknown parameter κ . We lift it to be a random variable equipped with a density function $\mathcal{P}(\kappa; \vec{\mu}_\kappa)$ where $\vec{\mu}_\kappa$ collectively denotes its moments. In the most general case, κ is not separable from the PDE residual which we denote as

$$\mathcal{L}(u(\vec{x}), \partial u, \vec{x}, \kappa) = 0, \quad (1)$$

where ∂u denotes all differential operators acting on the dependent variable u . Typical PINN formalism asserts \mathcal{L}^2 to be part of the overall loss function. In our model, we replace it by a marginalized version,

integrating the square of (1) over the domain of κ (\mathcal{D}_κ) to obtain

$$\mathcal{G}(u, \partial u, \vec{x}, \vec{\mu}_\kappa) \equiv \int_{\mathcal{D}_\kappa} d\kappa \mathcal{L}^2 \cdot \mathcal{P}(\kappa; \vec{\mu}_\kappa) = 0, \quad (2)$$

with \mathcal{G} being the weighted PDE residual, and $\vec{\mu}_\kappa$ are now learnable parameters instead of κ . The usual PINN formalism proceeds from here, yet an explicit form for $\mathcal{P}(\kappa; \vec{\mu}_\kappa)$ has to be specified, and \mathcal{G} is generally dependent on the chosen κ prior. For concreteness, in this paper, we consider the case where κ is separable from other terms within \mathcal{L} in the following sense.

$$\mathcal{L} = \mathcal{L}_1(u, \partial u, \vec{x}) + \kappa \mathcal{L}_2(u, \partial u, \vec{x}). \quad (3)$$

Marginalizing over κ via (2), we then obtain

$$\mathcal{G}(u, \partial u, \vec{x}, \vec{\mu}_\kappa) = \mathcal{L}_1^2 + \mathcal{L}_2^2 (\text{Var}(\kappa) + \bar{\kappa}^2) + \bar{\kappa} (2\mathcal{L}_1\mathcal{L}_2), \quad (4)$$

where $\vec{\mu}_\kappa = \{\bar{\kappa}, \text{Var}(\kappa)\}$ is part of the list of learnable parameters. Note that (4) is agnostic to the exact form of the prior $\mathcal{P}(\kappa; \vec{\mu}_\kappa)$, and that it reduces to the usual PDE residual loss of PINNs in the limit $\text{Var}(\kappa) \rightarrow 0$ with $\bar{\kappa} \rightarrow \kappa$. Despite its simplicity, the PDE residual (4) for \mathcal{L} is applicable for many inverse PINN problems studied in previous literature (e.g. [4, 5, 29]). It can be encapsulated within traditional Monte-Carlo Dropout [10] and Deep Ensemble [9, 11, 23] methods to yield uncertainty estimates for $u(\vec{x})$ apart from κ .

We now incorporate the fundamental principles of EDL [2, 27] in PINN and the residual loss (4). We first assume that the observed datapoints of u have been drawn i.i.d. from Gaussian profiles. Placing Gaussian and inverse-gamma priors on the means and variances respectively, the posterior distribution adopts the form of a 4-parameter normal-inverse-gamma distribution.

$$\tilde{p}(\bar{u}, \sigma_u^2 | \gamma, \nu, \alpha, \beta) = \frac{\beta^\alpha \sqrt{\nu}}{\Gamma(\alpha) \sqrt{2\pi\sigma_u^2}} e^{-\frac{2\beta + \nu(\gamma - \bar{u})^2}{2\sigma_u^2}}. \quad (5)$$

This assertion leads to simple closed-form formulas for the prediction means and uncertainties as follows.

$$\mathbb{E}[\bar{u}] = \gamma, \quad \sigma_p^2 = \frac{\beta}{\alpha - 1} \left(1 + \frac{1}{\nu}\right), \quad (6)$$

where we note that the expectation value \mathbb{E} is defined w.r.t. \tilde{p} in (5), and that the first and second components within the bracket are the aleatoric and epistemic uncertainties defined as $\mathbb{E}[\sigma_u^2]$, $\text{Var}[\bar{u}]$ respectively. In EDL, the learnable parameters are the hyperparameters $\{\gamma, \alpha, \beta, \nu\}$. Fitting data to the evidential

model requires not the usual mean-squared-error loss term but rather the negative logarithm of the marginal likelihood obtained after integrating the first-order Gaussian profiles over the measure described by \tilde{p} in (5).

$$\mathcal{L}_{edl} = \log((u_i - \gamma)^2 \nu + \Omega)^{(\alpha + \frac{1}{2})} + \log \frac{\sqrt{\pi/\nu} \Gamma(\alpha)}{\Omega^\alpha \Gamma(\alpha + \frac{1}{2})}, \quad (7)$$

where $\Omega \equiv 2\beta(1 + \nu)$ and u_i refer to the observed datapoints for the dependent variable u . In [2], the authors proposed a regularizer designed to minimize evidence on incorrect predictions that reads $\mathcal{L}_R = |u_i - \gamma|(2\nu + \alpha)$. This was shown in [2, 28] to work well, yet it is independent of β which controls the overall uncertainty scale. Here we propose a refinement of this original EDL regularizer by multiplying it to the Kullback-Leibler divergence [7, 21] between the distribution described by (5) and a weakly-informative prior, its final form being (see Appendix A for a detailed derivation)

$$\mathcal{L}_{kl} = |u_i - \gamma|(2\nu + \alpha) \left(\log \frac{\beta_r^\alpha \Gamma(\alpha)}{\beta^\alpha} + \gamma_e(\alpha - 1) + \frac{\beta - \beta_r}{\beta_r} \right), \quad (8)$$

where γ_e is the Euler-Mascheroni constant. In [8], β_r is a reference β value which is assumed to describe a weakly informative prior for larger deviations from the observed dataset. We set it to be of the same order-of-magnitude as the maximal target variable over the PDE domain. Empirically, we found this regularizer to perform better than \mathcal{L}_R that was used in [2, 28]. Theoretically, it aligns with the goal of guiding distributions for erroneous predictions to inherit weak priors with higher uncertainties.

To include a suitable PDE residual in the EDL framework, we take \mathcal{G} as defined in (4), but with $u \rightarrow \gamma$. Physically, this can be interpreted as the dependent variable being intrinsically stochastic in nature and the PDE should be understood as modeling the dynamics of its mean γ . The base model output variables are now represented by $\{\gamma, \alpha, \beta, \nu\}$ (instead of u) and thus the Dirichlet/Neumann boundary conditions should be expressed in terms of the mean γ . For the case studies here, we adopted the soft constraint approach for boundary conditions which were thus included as part of the training dataset. Our overall loss function for E-PINN reads

$$\mathcal{L}_{epinn} = \lambda_d \mathcal{L}_{edl} + \lambda_{kl} \mathcal{L}_{kl} + \lambda_r \mathcal{G}(\gamma, \partial\gamma, \vec{x}, \bar{\kappa}, \text{Var}(\kappa)), \quad (9)$$

where $\lambda_d, \lambda_{kl}, \lambda_r$ are the weight coefficients for the data (EDL) loss term, the regularizer term and PDE residual term respectively. In [9], we restore the arguments of \mathcal{G} to emphasize that the unknown parameter κ is now represented by the trainable parameters $\bar{\kappa}, \text{Var}(\kappa)$. Their learning is coupled to the output variables $\{\gamma, \alpha, \beta, \nu\}$ which yield the model's prediction and its associated uncertainty via (6).

3. Methodology

We validated our model against the same 1D and 2D inverse problems presented in the seminal paper [29] where B-PINNs were formulated. They are nonlinear differential equations with source terms designed such that the exact solutions are simple products of sinusoidal functions, enabling a more convenient analysis of the neural networks' effectiveness. In contrast to [29] however, we doped the training dataset with a more heterogeneous noise as we wish to scrutinize the degree to which the uncertainty distributions align with those of the inserted noise. The differential equations are of the same class as diffusion-reaction equations with non-linear source terms.

- [1] 1D nonlinear Poisson equation:

$$\lambda \frac{d^2 u}{dx^2} + \kappa \tanh(u) = f_{1D}(x), \quad u_e = \sin^3(6x).$$

- [2] 2D nonlinear diffusion-reaction PDE:

$$\lambda \nabla^2 u + \kappa u^2 = f_{2D}(x), \quad u_e = \sin(\pi x) \sin(\pi y).$$

Like in [29], we set $\lambda = 0.01$ and κ to be the unknown PDE parameter to be inferred. In [1] and [2], the exact solutions u_e and their corresponding source terms $f_{1D,2D}(x)$ are such that we have $\kappa = \{0.7, 1\}$ for the 1D and 2D equations respectively. For these problems, taking fully-connected multilayer perceptron (with a few hidden layers) to be the base models turned out to be sufficient for our case studies. The 1D and 2D problems required only three and two hidden layers respectively. The total number of iterations was 10^5 epochs, and Adam optimizer [20] was used with a learning rate of 10^{-3} . All E-PINN models were implemented with PyTorch [3] using its *Autograd* engine [24]. For the 1D problem, the optimal loss function parameters were found to be $\lambda_d = 1, \lambda_{kl} = 0.01, \lambda_r = 0.1$, whereas for the 2D problem, we used $\lambda_d = 1, \lambda_{kl} = 0.005, \lambda_r = 0.05$. As

for Bayesian-PINNs, the algorithm of [29] was implemented in PyTorch using the Hamiltonian-Monte-Carlo method [6, 29] to sample the posterior distributions. The number of samples used was of the order of 10^4 with a burn-in ratio of about 0.2.

For the 1D problem, we took the number of observational and collocation data points to be 200 (random) and 500 (regularly spaced) respectively. The training dataset domain was $x \in (-0.8, 0.7)$, yet domain for testing data was extended to $x = 1.2$ for assessing various models on out-of-distribution data. The data was doped with a standard Gaussian noise with a spatially varying amplitude, i.e. $\mathcal{A}(x)\mathcal{N}(0, 1)$ as depicted in Fig 1.

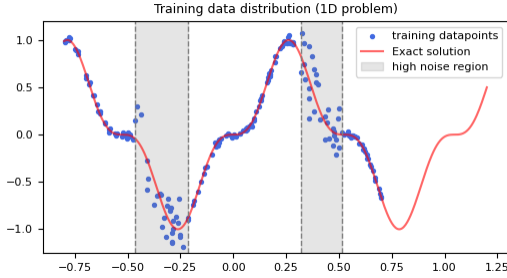


Figure 1: Training data distribution for the 1D problem. The noise amplitude $\mathcal{A}(x)$ was set to be 0.2 for the shaded ‘high-noise’ regions and 0.02 otherwise.

For the 2D problem, we also took the number of observational and collocation data points to be 200 and 500 respectively, with the domain being a square centered at the origin with length 2. In addition, we imposed a soft constraint realizing the Dirichlet condition along each of the boundary segments $\{x = \pm 1, y = \pm 1\}$ by including 50 points for each side of the square. We doped the training data with a standard Gaussian noise with a spatially varying amplitude $\mathcal{A}(x, y)$ that was maximal at the origin and vanishing along the boundary segments.

$$\mathcal{A}(x, y) = 0.1 \cos\left(\frac{\pi x}{2}\right) \cos\left(\frac{\pi y}{2}\right) e^{-(x^2+y^2)}. \quad (10)$$

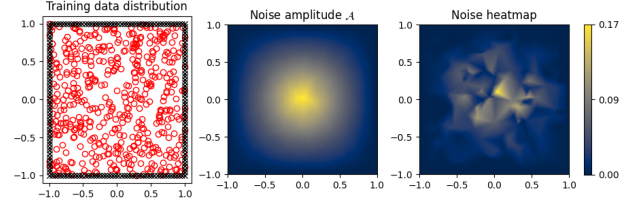


Figure 2: Data and noise distributions for the 2D problem. The rightmost diagram show the noise generated by the stochastic function $\mathcal{A}(x, y)\mathcal{N}(0, 1)$.

As a metric for assessing uncertainty quantification, we monitored the *empirical coverage probability* (ECP) [12, 13] defined as the proportion of samples of which deviations from the exact solution fell within the confidence band (we took it to be $\pm 1.96\sigma_p$ for a 0.95 CI). We also kept track of the Spearman’s coefficients between σ_p and (i) prediction errors (ρ_e) (ii) noise-induced errors in the training data (ρ_n). We compared our E-PINN model against B-PINN, Deep Ensemble (DE) method (defined by a set of 20 PINNs with different initial weights) and E-PINN using the original EDL regularizer of [2] rather than [8] (denoted as EPINN(V) henceforth).

4. Results

For the 1D problem, as visible in Fig. 3 the predictive uncertainty σ_p was larger in the high-noise regions and beyond the training domain, with the 0.95 confidence band encompassing both exact solution and model-predicted curve. Taking $\sigma_\kappa \equiv \sqrt{\text{Var}(\kappa)}$ to be the standard error, E-PINN predicted $\kappa \approx 0.71 \pm 0.03$, close to the actual value of 0.7. The ECP value was 0.96 which was closest to the nominal value of 0.95 among all models. Compared to E-PINN(V), our model inferred a more accurate κ and was better calibrated as measured by its ECP. Relative to B-PINN, it inherited stronger correlations with prediction errors (ρ_e), noise-induced deviations of the training data from exact solution (ρ_n) and had a better ECP score. The Deep Ensemble inferred κ with a much smaller uncertainty than other models, but its ECP appeared to be problematic indicating a poor degree of calibration. Like B-PINN, the correlation of its uncertainty distribution with the noise of the data was weak ($\rho_n \sim 0.1$).

Table 1: Results for 1D ODE where $\kappa_{true} = 0.7$

	E-PINN	E-PINN(V)	B-PINN	DE
κ	0.71	0.74	0.69	0.6976
σ_κ	0.03	0.03	0.03	10^{-4}
ρ_e	0.7	0.6	0.5	0.6
ρ_n	0.7	0.7	0.2	0.1
ECP	0.96	0.79	0.87	0.38

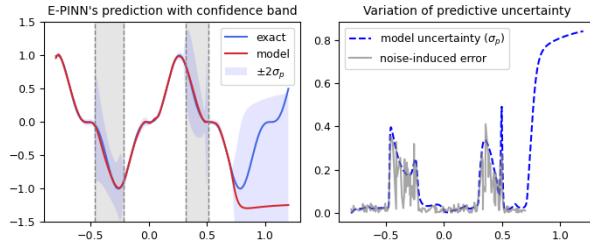


Figure 3: E-PINN's predictions and uncertainty distributions (1D problem).

For the 2D problem, we found E-PINN to exhibit similar advantages relative to other frameworks. It inferred κ to be 1.00 ± 0.01 , with a ECP value of 0.89, and Spearman's coefficients $\rho_e \sim 0.7$, $\rho_n \sim 0.8$. The mean model error was ~ 0.02 while it was ~ 0.002 when restricted to boundary, showing strong adherence to the Dirichlet boundary condition.

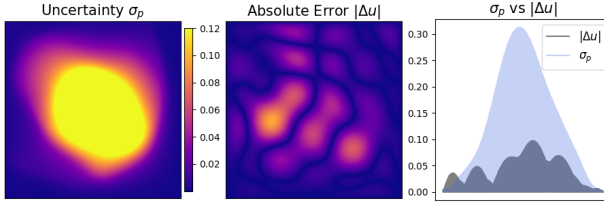


Figure 4: E-PINN's predictions and uncertainty distribution heatmaps (2D problem).

For B-PINN, the mean model error was ~ 0.006 while it was ~ 0.01 when restricted to boundary, showing weaker adherence to boundary conditions. E-PINN(V)'s prediction for $\kappa \approx 1.02 \pm 0.01$ did not quite cover the actual value of 1.00, and was weaker in its ECP value compared to E-PINN. The Deep Ensemble

inferred κ with an uncertainty much larger than other models, and its ECP value deviated most significantly from the nominal value. Uncertainty heatmaps and error plots for these models are relegated to Appendix C.

Table 2: Results for the 2D PDE where $\kappa_{true} = 1$. *Abbrev:* Δu_b = mean model error evaluated along domain's boundary

	E-PINN	E-PINN(V)	B-PINN	DE
κ	1.00	1.02	1.01	0.99
σ_κ	0.01	0.01	0.04	0.09
ρ_e	0.7	0.7	0.1	0.5
ρ_n	0.8	0.6	-0.2	0.5
ECP	0.93	0.89	1.00	0.63
Δu_b	0.002	0.006	0.01	0.001

5. Conclusion

We have constructed a novel uncertainty-aware PINN-like model synergizing the fundamental principles of EDL and PINN. It provides another data-driven framework in scientific machine learning that is suited for solving inverse problems involving differential equations. E-PINN generates uncertainty estimates for *both* the target variable and unknown PDE parameters upon completion of model training. The novel information-theoretic regularizer we proposed in eqn. (8) appeared to enhance the model's effectiveness relative to the original version first proposed in [2]. For the 1D and 2D problems examined here, we found that E-PINNs superseded B-PINNs and other models in terms of sensitivity of uncertainties towards data noise and predictive errors. Its empirical coverage probability values were close to nominal ones indicating a good degree of uncertainty calibration. The 1D problem showed E-PINN to be effective in interpreting out-of-domain data, while the 2D problem illustrated the model's ability to adhere to boundary conditions. An important future direction would be to validate E-PINN against a wider set of PDEs used in various scientific and engineering fields, such as those proposed in *PINNacle* [15].

References

- [1] ABDAR, M., POURPANAH, F., HUSSAIN, S., REZAZADEGAN, D., LIU, L., GHAVAMZADEH, M., FIEGUTH, P., CAO, X., KHOSRAVI, A., ACHARYA, U. R., MAKARENKOV, V., AND NAHAVANDI, S. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion* 76 (dec 2021), 243–297.
- [2] AMINI, A., SCHWARTING, W., SOLEIMANY, A., AND RUS, D. Deep evidential regression. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (Red Hook, NY, USA, 2020), NIPS '20, Curran Associates Inc.
- [3] BAFGHI, R. A., AND RAISSI, M. Comparing PINNs across frameworks: JAX, tensorflow, and pytorch. In *ICLR 2024 Workshop on AI4DifferentialEquations In Science* (2024).
- [4] BATY, H. A hands-on introduction to physics-informed neural networks for solving partial differential equations with benchmark tests taken from astrophysics and plasma physics, 2024.
- [5] CHEN, Y., LU, L., KARNIADAKIS, G. E., AND DAL NEGRO, L. Physics-informed neural networks for inverse problems in nano-optics and metamaterials. *Optics Express* 28, 8 (Apr. 2020), 11618.
- [6] COBB, A. D., AND JALAIAN, B. Scaling hamiltonian monte carlo inference for bayesian neural networks with symmetric splitting. *Uncertainty in Artificial Intelligence* (2021).
- [7] COVER, T. M., AND THOMAS, J. A. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA, 2006.
- [8] CUOMO, S., DI COLA, V. S., GIAMPAOLO, F., ROZZA, G., RAISSI, M., AND PICCIALLI, F. Scientific machine learning through physics-informed neural networks: Where we are and what's next, 2022.
- [9] EGELE, R., MAULIK, R., RAGHAVAN, K., LUSCH, B., GUYON, I., AND BALAPRAKASH, P. Autodeuq: Automated deep ensemble with uncertainty quantification. In *2022 26th International Conference on Pattern Recognition (ICPR)* (Los Alamitos, CA, USA, aug 2022), IEEE Computer Society, pp. 1908–1914.
- [10] GAL, Y., AND GHAHRAMANI, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning* (New York, New York, USA, 20–22 Jun 2016), M. F. Balcan and K. Q. Weinberger, Eds., vol. 48 of *Proceedings of Machine Learning Research*, PMLR, pp. 1050–1059.
- [11] GANAIE, M., HU, M., MALIK, A., TANVEER, M., AND SUGANTHAN, P. Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence* 115 (Oct. 2022), 105151.
- [12] GNEITING, T., BALABDAOUI, F., AND RAFTERY, A. E. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 69, 2 (03 2007), 243–268.
- [13] GNEITING, T., AND RAFTERY, A. E. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association* 102, 477 (03 2007), 359–378.
- [14] HAO, Z., LIU, S., ZHANG, Y., YING, C., FENG, Y., SU, H., AND ZHU, J. Physics-informed machine learning: A survey on problems, methods and applications, 2023.
- [15] HAO, Z., YAO, J., SU, C., SU, H., WANG, Z., LU, F., XIA, Z., ZHANG, Y., LIU, S., LU, L., AND ZHU, J. Pinnacle: A comprehensive benchmark of physics-informed neural networks for solving pdes, 2023.
- [16] HORNIK, K., STINCHCOMBE, M., AND WHITE, H. Multilayer feedforward networks are universal approximators. *Neural Networks* 2, 5 (1989), 359–366.
- [17] JAGTAP, A. D., MAO, Z., ADAMS, N., AND KARNIADAKIS, G. E. Physics-informed neural

- networks for inverse problems in supersonic flows. *Journal of Computational Physics* 466 (2022), 111402.
- [18] JONES, C. K., WANG, G., YEDAVALLI, V., AND SAIR, H. Direct quantification of epistemic and aleatoric uncertainty in 3d u-net segmentation. *J Med Imaging (Bellingham)* 9, 3 (May 2022), 034002.
- [19] KIM, D., AND LEE, J. A review of physics informed neural networks for multiscale analysis and inverse problems. *Multiscale Science and Engineering* 6 (02 2024).
- [20] KINGMA, D. P., AND BA, J. Adam: A method for stochastic optimization, 2017.
- [21] KULLBACK, S., AND LEIBLER, R. A. On Information and Sufficiency. *The Annals of Mathematical Statistics* 22, 1 (1951), 79 – 86.
- [22] LAGARIS, I., LIKAS, A., AND FOTIADIS, D. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks* 9, 5 (1998), 987–1000.
- [23] LAKSHMINARAYANAN, B., PRITZEL, A., AND BLUNDELL, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Red Hook, NY, USA, 2017), NIPS’17, Curran Associates Inc., p. 6405–6416.
- [24] PASZKE, A., GROSS, S., CHINTALA, S., CHANAN, G., YANG, E., DEVITO, Z., LIN, Z., DESMAISON, A., ANTIGA, L., AND LERER, A. Automatic differentiation in pytorch. In *NIPS 2017 Workshop on Autodiff* (2017).
- [25] RAISSI, M., PERDIKARIS, P., AND KARNIADAKIS, G. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics* 378 (2019), 686–707.
- [26] RAISSI, M., PERDIKARIS, P., AND KARNIADAKIS, G. E. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations, 2017.
- [27] SENSOY, M., KAPLAN, L., AND KANDEMIR, M. Evidential deep learning to quantify classification uncertainty, 2018.
- [28] TAN, H. S., WANG, K., AND MCBETH, R. Deep evidential learning for radiotherapy dose prediction. *Computers in Biology and Medicine* 182 (Nov. 2024), 109172.
- [29] YANG, L., MENG, X., AND KARNIADAKIS, G. E. B-pinns: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data. *Journal of Computational Physics* 425 (Jan. 2021), 109913.
- [30] ZHOU, W., AND XU, Y. F. Data-guided physics-informed neural networks for solving inverse problems in partial differential equations, 2024.
- [31] ZOU, K., YUAN, X., SHEN, X., WANG, M., AND FU, H. Tbrats: Trusted brain tumor segmentation. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part VIII* (Berlin, Heidelberg, 2022), Springer-Verlag, p. 503–513.

A. Supplementary derivations for Section 2.

In this appendix, we collect various mathematical details omitted in Section 2 for additional clarity.

To derive the residual function \mathcal{G} in eqn. (4), we note that

$$\begin{aligned}\mathcal{G} &= \int_{\mathcal{D}_{\kappa}} d\kappa \mathcal{P}(\kappa; \vec{\mu}_{\kappa}) [\mathcal{L}_1^2 + \mathcal{L}_2^2 \kappa^2 + 2\kappa \mathcal{L}_1 \mathcal{L}_2] \\ &= \mathcal{L}_1^2 + \mathcal{L}_2^2 \overline{\kappa^2} + 2\overline{\kappa} \mathcal{L}_1 \mathcal{L}_2 \\ &\equiv \mathcal{L}_1^2 + \mathcal{L}_2^2 (\text{Var}(\kappa) + \overline{\kappa}^2) + 2\overline{\kappa} \mathcal{L}_1 \mathcal{L}_2.\end{aligned}\quad (11)$$

When \mathcal{G} is incorporated in Monte-Carlo Dropout for PINN, the uncertainty in κ is not affected by the stochastic dropout units that are switched on during each forward pass, but has already been learned during model training. On the other hand, for Deep Ensemble, one averages over the collection of models to deduce both $\overline{\kappa}$ and $\text{Var}(\kappa)$.

With regards to our regularizer loss term \mathcal{L}_{kl} , we had obtained it by taking the KL divergence between the normal-inverse-gamma (NIG) distribution

parametrized by $\{\gamma, \alpha, \beta, \nu\}$ and another reference NIG distribution with $\{\gamma_r, \alpha_r, \beta_r, \nu_r\}$ that presumably characterizes a non-/weakly informative prior. When multiplied to a term such as $|u_i - \gamma|$ in (8), this means that for more significantly large errors, the loss term guides the distributions for such samples to approach a weakly-informative one yielding higher uncertainty. By definition, the KL divergence between two NIG distributions with density functions (\tilde{p}_r, \tilde{p}) is

$$D_{KL}(\tilde{p}_r || \tilde{p}) = \iint d\bar{u} d\sigma_u^2 \tilde{p}_r \log \frac{\tilde{p}_r}{\tilde{p}}. \quad (12)$$

The NIG distribution can be equivalently understood as the product

$$\bar{u} \sim \mathcal{N}(\gamma, \frac{\sigma_u^2}{\nu}), \quad \sigma_u^2 \sim \Gamma^{-1}(\alpha, \beta).$$

We would like the reference prior to be a non- or weakly-informative prior. Taking the limit of $\nu_r \rightarrow \infty$ yields a uniform distribution over \mathbb{R} which is unfortunately unnormalizable. To avoid this pathology, we take the reference NIG to be parametrized by values of γ, ν identical to those of the EDL model outputs' distributions, and instead set the reference inverse-gamma distribution to be of relatively higher entropy, taking $\alpha_r = 1$ and letting β_r be task-dependent. Since the variance σ_u^2 is not physically expected to be larger than the upper bound for u , a natural candidate for β_r is the order-of-magnitude of u_{max} – the maximum value of the dependent variable as deduced from physical principles. For the differential equations examined in our work here, we take $\beta_r = 1$. Generally, it should be set to the same order-of-magnitude as u_{max} . This definition of \tilde{p}_r leads to the KL divergence term being

$$\mathcal{D}_{KL} = \left(\log \frac{\beta_r^\alpha \Gamma(\alpha)}{\beta^\alpha} + \gamma_e(\alpha - 1) + \frac{\beta - \beta_r}{\beta_r} \right). \quad (13)$$

In this form, we note that there is no dependence on ν . In [2], the authors argued that the NIG distribution admits the interpretation of having its variance estimated from α ‘virtual observations’ with sample mean γ and sum of squared deviations 2ν , and one can thus interpret $2\nu + \alpha$ as a measure of evidence for the EDL distributions. Together with a linear term $|u_i - \gamma|$ that aims to increase the loss for predictions with larger errors, the authors of [2] then took $|u_i - \gamma|(2\nu + \alpha)$ to be the regularizer term. Since previous works [2, 28] has shown it to be effective in a number of contexts

and it carries a dependence on ν , we multiply it to the KL term in (13) to obtain our final expression for the regularizer loss term

$$\mathcal{L}_{kl} = |u_i - \gamma|(2\nu + \alpha)D_{kl}. \quad (14)$$

B. Learning curves for E-PINN training

Model training for E-PINN involves choices of loss function weight coefficients $\lambda_d, \lambda_r, \lambda_{kl}$. Apart from the individual loss terms, they should yield learning curves which show stable convergent behavior for $\bar{\kappa}, \sigma_\kappa$. Fig. 5 depicts the evolutions of $\bar{\kappa}, \sigma_\kappa$ for the 2D problem. While the latter decreased steadily, the former experienced a transient oscillatory phase before becoming stable near the theoretical κ value. This phase mirrored the dynamical balance between the PDE residual loss \mathcal{G} and the EDL-type loss terms $\lambda_d \mathcal{L}_{edl} + \lambda_{kl} \mathcal{L}_{kl}$.

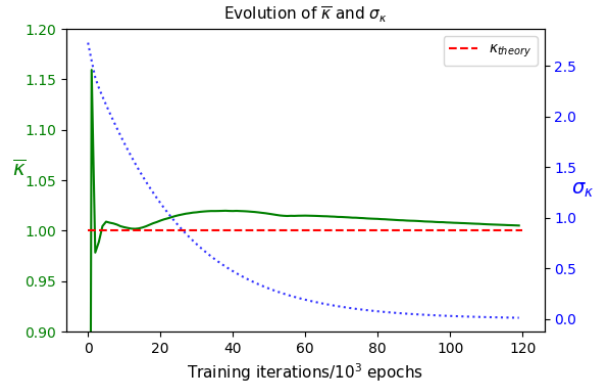


Figure 5: Evolution of $\bar{\kappa}, \sigma_\kappa$ superposed on the same graph for the 2D case.

In Fig. 6, we observe that the early sharp rise in the PDE residual loss parallels the transient oscillatory learning curve of $\bar{\kappa}$ present in Fig. 5. The EDL data loss term $\lambda_d \mathcal{L}_{edl} + \lambda_{kl} \mathcal{L}_{kl}$ evolved similarly to the total loss function shown in Fig. 5. Similar trends for these training curves and evolutions of $\bar{\kappa}, \sigma_\kappa$ were observed for the 1D case too.

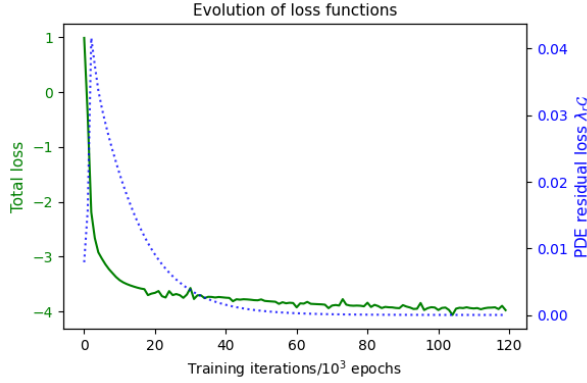


Figure 6: Evolution of the total and PDE loss functions for the 2D case.

Setting $\lambda_d = 1$, we took $\lambda_{kl} = 0.01, \lambda_r = 0.1$ for the 1D problem and $\lambda_{kl} = 0.05, \lambda_r = 0.5$ for the 2D problem. Like the usual PINN framework, E-PINN training is fundamentally semi-supervised in nature, blending a data-based component with a PDE-residual term, with no labels in the training dataset for κ, σ_κ nor the uncertainty estimates. Yet we can use the degree of convergent behaviors of the each loss function component and evolution of κ, σ_κ to guide us in excluding unsuitable values of $\lambda_d, \lambda_{kl}, \lambda_r$. In more realistic application settings, the uncertainties in the dependent variable u and unknown parameter κ may be linked to physical principles (such as the precision of sensors measuring κ or u) which would give an order-of-magnitude guide for σ_κ, σ_p . This in turn can be useful for picking suitable weight coefficients for each loss term.

C. Additional details on the results for other uncertainty-aware PINN models

In this Appendix, we provide some additional details on the simulation and training results for

- [A] B-PINN with Hamiltonian-Monte-Carlo for posterior sampling [29].
- [B] E-PINN(V): E-PINN equipped not with (8) but with the original regularizer of [2] : $\mathcal{L}_{kl} =$

$$|u_i - \gamma|(2\nu + \alpha),$$

- [C] Deep Ensemble (DE) method using a set of 20 PINNs with distinct initial weights.

[A] B-PINN models exhibited weaker correlations with noise and errors compared to our E-PINN models. In Fig. 7, one can see that the confidence band was not particularly enlarged within the high-noise regions, in contrast to our E-PINN model (Fig. 3). It yielded weak correlations between σ_p and model errors ($\rho_e \sim 0.5$) or noise-induced errors ($\rho_n \sim 0.2$). The model's empirical coverage probability was 0.87. Predicted value for κ with its uncertainty was $\kappa = 0.69 \pm 0.03$, while its mean predictive uncertainty was $\sigma_p \sim 0.03$.

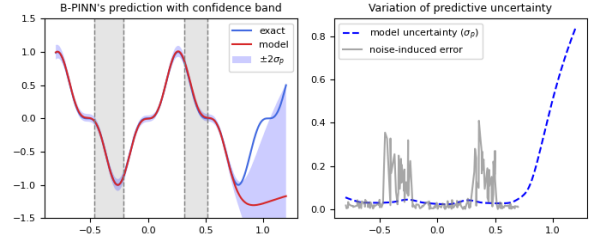


Figure 7: B-PINN's predictions and uncertainty distributions (1D problem).

For the 2D problem, B-PINN model yielded uncertainty estimates that tended to be inflated relative to magnitudes of model errors (ECP was thus unity) as depicted in Fig. 7, with vague correlations between σ_p and model errors ($\rho_e \sim 0.1$) or noise-induced errors ($\rho_n \sim -0.2$). The mean uncertainty $\bar{\sigma}_p \sim 0.02$. For the PDE parameter, the model inferred $\kappa \sim 1.01 \pm 0.04$.

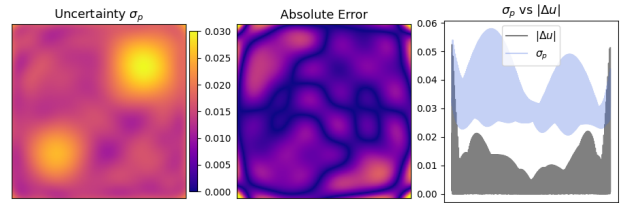


Figure 8: B-PINN's predictions and uncertainty distribution heatmaps (2D problem).

[B] For E-PINN(V), we found that the main disadvantage relative to our proposed \mathcal{L}_{kl} was a weaker response to out-of-distribution data. In Fig. 9, one can see that although the confidence band was enlarged within the high-noise regions, in contrast to our E-PINN model (Fig. 3), the predictive uncertainty σ_p was relatively much lower and not encompassing the deviations from the exact solution. The Spearman’s coefficients ρ_s were $\{0.6, 0.7\}$ for uncertainty correlations with error and noise respectively. Its empirical coverage probability was 0.79. Predicted value for κ with its uncertainty was $\kappa = 0.74 \pm 0.03$, while its mean predictive uncertainty was $\sigma_p \sim 0.02$.

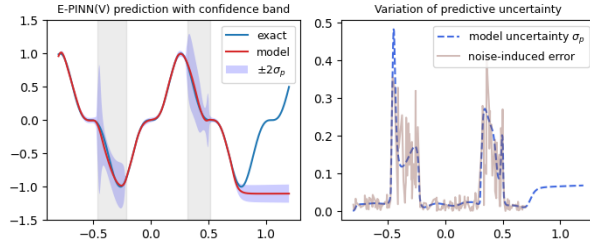


Figure 9: Predictions and uncertainty distributions for E-PINN(V) (1D problem).

For the 2D problem, E-PINN(V) was weaker than our model in the aspect of the inferred parameter’s accuracy, ECP value and mean error along boundary, yet was comparable in other aspects tabulated in Table 2.

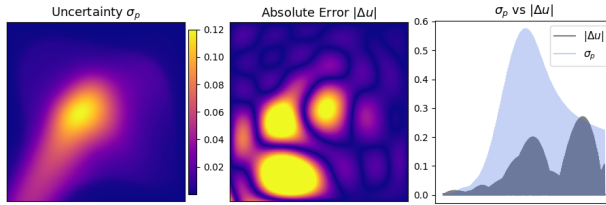


Figure 10: E-PINN(V)’s predictions and uncertainty distribution heatmaps (2D problem).

[C] We implemented DE using a set of 20 PINNs with κ as the only trainable parameter in the PDE residual loss), with distinct initial random weights

set in Pytorch via Kaiming initialization. Training was completed with 10^5 epochs with Adam optimizer under a constant learning rate of 10^{-3} . For the 1D problem DE generated an uncertainty spectrum that did not appear to be sensitive to noise at all, though its confidence band for out-of-domain data was better calibrated compared to B-PINN and E-PINN(V). Predicted value for κ with its uncertainty was $\kappa = 0.6976 \pm 10^{-4}$.

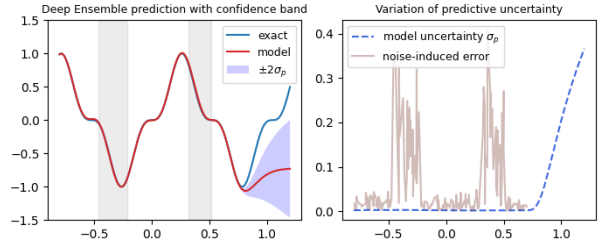


Figure 11: Predictions and uncertainty distributions for the Deep Ensemble method (1D problem).

For the 2D problem, DE inferred the PDE parameter $\kappa \sim 0.99 \pm 0.09$, and relative to its performance in the 1D case, yielded a better ECP = 0.63. The mean uncertainty was $\bar{\sigma}_p \sim 0.003$, with the mean error ~ 0.008 error restricted to boundary being ~ 0.001 . Relative to B-PINN, it notably had better uncertainty-error correlations, with ρ_e, ρ_n both being ~ 0.5 . From both 1D and 2D problems, the main weakness of this method seemed to be that the uncertainty values were too low giving rise to a rather problematic ECP. This implied that DE did not yield well-calibrated uncertainty estimates, as was also apparent in both uncertainty distribution plots.

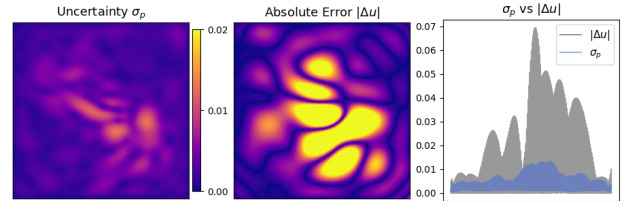


Figure 12: Predictions and uncertainty heatmaps for the Deep Ensemble method (2D problem).