$PINN - A DOMAIN DECOMPOSITION METHOD FOR BAYESIAN PHYSICS-INFORMED NEURAL NETWORKS

Júlia Vicens-Figueres*
Universitat Pompeu Fabra

Juliette Vanderhaeghen*
Université Catholique de Louvain

Federica Bragone
KTH Royal Institute of Technology

Kateryna Morozovska
Imperial College London

Khemraj Shukla
Brown University

ABSTRACT

Physics-Informed Neural Networks (PINNs) are a novel computational approach for solving partial differential equations (PDEs) with noisy and sparse initial and boundary data. However, efficient quantification of epistemic and aleatoric uncertainties in big multi-scale problems remains challenging. We propose $PINN, a novel method of computing global uncertainty in PDEs using a Bayesian framework, by combining local Bayesian PINNs (BPINNs) with domain decomposition. The solution continuity across subdomains is obtained by imposing flux continuity across the interface of neighboring subdomains. Although we have adopted conservative PINNs (cPINNs), the method can be seamlessly extended to other domain decomposition techniques. The results show that the proposed method recovers the global uncertainty by computing the local uncertainty exactly more efficiently, as the uncertainty in each subdomain can be computed concurrently. The robustness of $PINN is verified by adding uncorrelated random noise to the training data up to 15% and testing for different domain sizes.

1 INTRODUCTION

During recent years, deep learning techniques have progressed towards scalable tools for solving complex scientific problems. Physics-Informed Neural Networks (PINNs) Raissi et al. (2019) embed physics described by ODEs and PDEs into the learning process, enabling both solution and discovery of equations. The ability of PINNs to combine existing knowledge with observed data can improve computational efficiency Cuomo et al. (2022); Markidis (2021).

Despite their success, PINNs have limitations in uncertainty quantification (UQ) and scalability Psaros et al. (2023). Traditionally, data-driven models experience two types of uncertainties: epistemic and aleatoric. Bayesian inference has quantified uncertainties in PDEs Li & Marzouk (2014); Yan & Zhou (2019). BPINN Yang et al. (2021) integrates PINN with Bayesian optimization to handle noisy data, using priors on weights/biases and sampling (e.g., Hamiltonian Monte-Carlo HMC Radivojević & Akhmatskaya (2020) or variational inference Blei et al. (2017)) for posterior estimation. Thus BPINN improves handling of uncertainty, but increases costs due to iterative sampling.

A second challenge is scalability for larger problems (complex domains, 3D, space-time). Domain decomposition helps by splitting into subdomains for parallel training Shukla et al. (2021). Methods vary: overlapping/non-overlapping/adaptive domains Klawonn et al. (2024), with challenges in interface loss and time-domain handling (e.g., cPINN Jagtap et al. (2020) excels spatially but constrains time; XPINN Shukla et al. (2021) generalizes but risks overfitting Hu et al. (2022)).

Combining Bayesian inference with domain decomposition enables UQ where subdomains have varying noise. We propose $PINN, combining BPINN advantages for uncertainties with cPINN for multi-scale problems. We test our method over different criteria: different PDEs; variable noise levels; variable noise per domain; domain sizes; forward/inverse; 1D/2D multi-scale.

*Equal contribution.

2 METHODOLOGY

We propose $PINN, combining Bayesian Physics-Informed Neural Networks (BPINN) with domain decomposition via conservative PINNs (cPINN).

BPINN Yang et al. (2021) extends PINNs by assigning prior distributions to network parameters $\boldsymbol{\theta}$ and PDE parameters $\boldsymbol{\lambda}$, enabling uncertainty quantification. Given noisy data $\mathcal{D} = \mathcal{D}_u \cup \mathcal{D}_\phi \cup \mathcal{D}_b$ (solution, residual, boundary/initial conditions), the likelihood is:

$$P(\mathcal{D}\,|\,\boldsymbol{\theta}) = P(\mathcal{D}_u\,|\,\boldsymbol{\theta})P(\mathcal{D}_\phi\,|\,\boldsymbol{\theta})P(\mathcal{D}_b\,|\,\boldsymbol{\theta}), \tag{1}$$

where each term follows a Gaussian distribution and the posterior $P(\boldsymbol{\theta}, \boldsymbol{\lambda} \mid \mathcal{D}) \propto P(\mathcal{D} \mid \boldsymbol{\theta}, \boldsymbol{\lambda})P(\boldsymbol{\theta})P(\boldsymbol{\lambda})$ is sampled via HMC Radivojević & Akhmatskaya (2020).

cPINN Jagtap et al. (2020) decomposes the domain into subdomains for training and uses interface conditions to enforce flux continuity. For subdomains q and q^+ with common interface, the loss is:

$$\text{Interface Loss} = \text{MSE}_{u_{avg}} + \text{MSE}_{flux}, \tag{2}$$

$$\text{MSE}_{u_{avg}} = \frac{1}{N_{I_q}} \sum_{i=1}^{N_{I_q}} |\hat{u_q}(x^i_{I_q}, t^i_{I_q}) - \{\{u(x^i_{I_q}, t^i_{I_q})\}\}|^2, \tag{3}$$

$$\text{MSE}_{flux} = \frac{1}{N_{I_q}} \sum_{i=1}^{N_{I_q}} |f_q(u(x^i_{I_q}, t^i_{I_q})) \cdot \mathbf{n} - f_{q^+}(u(x^i_{I_q}, t^i_{I_q})) \cdot \mathbf{n})|^2, \tag{4}$$

where $f_q \cdot \mathbf{n}$ are interface fluxes and $\{\{u\}\} = (u_q + u_{q^+})/2$ is the average solution.

$PINN (shown in Figure 1) combines BPINN's uncertainty quantification with cPINN's scalability.

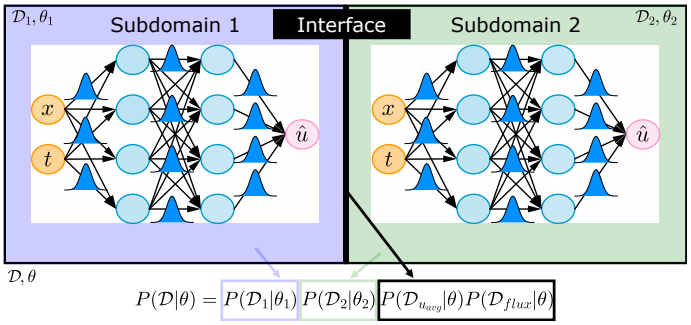

Figure 1: Structure of $PINN with two subdomains and shared interface.

The likelihood extends BPINN by adding interface terms:

$$P(\mathcal{D}\,|\,\boldsymbol{\theta}) = P(\mathcal{D}_u)P(\mathcal{D}_\phi)P(\mathcal{D}_{IC})P(\mathcal{D}_{BC})P(\mathcal{D}_{u_{avg}})P(\mathcal{D}_{flux}), \tag{5}$$

where $P(\mathcal{D}_u)$, $P(\mathcal{D}_\phi)$, $P(\mathcal{D}_{IC})$, $P(\mathcal{D}_{BC})$ are standard BPINN terms (conditioning on $\boldsymbol{\theta}$ omitted for brevity), and the interface likelihoods are:

$$P(\mathcal{D}_{u_{avg}}\,|\,\boldsymbol{\theta}) = \prod_{i=1}^{N_{CDC}} \frac{1}{\sqrt{2\pi\sigma^{(i)2}_{u_{avg}}}} \exp\left[-\frac{\left(|\,\hat{u}_1(x^{(i)}_{CDC}, t^{(i)}_{CDC}; \boldsymbol{\theta}) - \mathcal{I}_{u_{avg}}\,| \right)^2}{2\sigma^{(i)2}_{u_{avg}}} \right], \tag{6}$$

$$P(\mathcal{D}_{flux}|\boldsymbol{\theta}) = \prod_{i=1}^{N_{CDC}} \frac{1}{\sqrt{2\pi\sigma^{(i)2}_{flux}}} \exp\left[-\frac{\left(|\,f_1(\hat{u}_1(x^{(i)}_{CDC}, t^{(i)}_{CDC}; \boldsymbol{\theta})) \cdot \mathbf{n} - \mathcal{I}_{flux}\,| \right)^2}{2\sigma^{(i)2}_{flux}} \right], \tag{7}$$

where N_{CDC} is the number of interface collocation points, and $\mathcal{I}_{u_{avg}}$, $\mathcal{I}_{flux}$ ensure continuity of both solution and flux at the interface, key for conservative PDEs.

$$\mathcal{I}_{u_{avg}} = \frac{\hat{u}_1(x^{(i)}_{CDC}, t^{(i)}_{CDC}; \boldsymbol{\theta}) + \hat{u}_2(x^{(i)}_{CDC}, t^{(i)}_{CDC}; \boldsymbol{\theta})}{2}, \quad \mathcal{I}_{flux} = \frac{f_1(\hat{u}_1) \cdot \boldsymbol{n} + f_2(\hat{u}_2) \cdot \boldsymbol{n}}{2}. \tag{8}$$

Inverse Problem. For unknown PDE parameters $\boldsymbol{\lambda}$, we test three formulations. First, separate parameters per subdomain ($\boldsymbol{\lambda}_1$, $\boldsymbol{\lambda}_2$) with residual likelihood $P(\mathcal{D}_\phi \,|\, \boldsymbol{\theta}; \boldsymbol{\lambda}) = P(\mathcal{D}_{\phi_1} \,|\, \boldsymbol{\theta}; \boldsymbol{\lambda}_1) P(\mathcal{D}_{\phi_2} \,|\, \boldsymbol{\theta}; \boldsymbol{\lambda}_2)$. Then, to enforce consistency, we add a **soft constraint**:

$$P(\mathcal{D}_\lambda \,|\, \boldsymbol{\theta}) = \prod_{i=1}^{N_{CDC}} \frac{1}{\sqrt{2\pi\sigma_\lambda^{(i)^2}}} \exp\left[-\frac{\left(\boldsymbol{\lambda}_1 - \frac{\boldsymbol{\lambda}_1 + \boldsymbol{\lambda}_2}{2}\right)^2}{2\sigma_\lambda^{(i)^2}} \right]. \tag{9}$$

However, this still allows mismatch. To reduce this discrepancy, we introduce a **hard constraint** by enforcing a single $\boldsymbol{\lambda}$ for both subdomains $P(\mathcal{D}_\phi \,|\, \boldsymbol{\theta}; \boldsymbol{\lambda}) = P(\mathcal{D}_{\phi_1} \,|\, \boldsymbol{\theta}; \boldsymbol{\lambda}) P(\mathcal{D}_{\phi_2} \,|\, \boldsymbol{\theta}; \boldsymbol{\lambda})$ where each $P(\mathcal{D}_{\phi_j})$ follows the standard BPINN form conditioned on the shared $\boldsymbol{\lambda}$.

3 Results and Implementation

\$PINN is validated on four benchmark PDEs: Burgers, Fisher-KPP, Fokker-Planck (Appendix A.1), and Allen-Cahn equations, using three training scenarios: (BI) initial/boundary conditions with collocation points; (BIC) adds interface data to BI to enforce continuity across the interface; (RD) uses randomly sampled domain data. We show both forward (recovering $u(x,t)$ from IC/BC) and inverse problems (inferring PDE parameters $\boldsymbol{\lambda}$), noise robustness (up to 15%), and sub-domain scalability.

Network Architecture. Each subdomain uses a fully connected neural network with 3 hidden layers (50 neurons each), tanh activation. We use HMC Radivojević & Akhmatskaya (2020) with 500 burn-in samples, 1000 posterior samples (leapfrog steps: 50, step size: 10^{-3}). Training data: $N_u = 50$ (IC/BC), $N_\phi = 500$ (collocation), $N_{CDC} = 20$ (interface).

Experimental Suite. We conduct forward and inverse tests for all four PDEs in 1D with 2 subdomains under boundary-informed (BC) and boundary-initial (BIC) conditions, testing noise levels from 0% to 15%. For Allen-Cahn, we extend to 3-4 subdomain decomposition. For Fokker-Planck, we also test the 2D forward problem (Appendix A.1). Inverse problems infer viscosity ν (Burgers), growth rate ρ (Fisher-KPP, Allen-Cahn), and diffusion/drift coefficients D, V (Fokker-Planck).

Allen-Cahn Forward Problem. Figure 2 compares BPINN (top) with \$PINN BI (middle) and BIC (bottom) across noise levels. Both methods maintain accuracy throughout 0-15% noise. Adding interface data (BIC) significantly reduces epistemic uncertainty, particularly near domain boundaries. The RD case (not shown) exhibits increased overfitting at 15% noise but maintains comparable accuracy to BPINN. Tests with asymmetric noise (one clean domain, one noisy) show that localized noise does not compromise clean regions, suggesting benefits of incorporating variable-quality data.

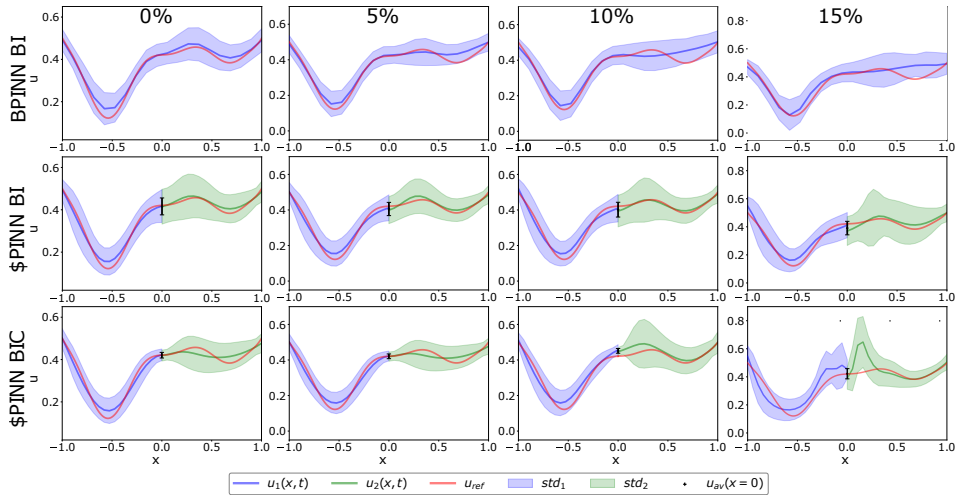

Figure 2: Allen-Cahn with BPINN and \$PINN using IC, BC and CDC.

Cross-PDE Validation Tests on Burgers, Fokker-Planck, and Fisher-KPP confirm consistent behavior. $PINN achieves comparable accuracy to BPINN with fewer collocation points. BIC consistently reduces interface uncertainty across all PDEs. Variable domain sizes (tested with 2/3 vs. 1/3 splits) do not degrade performance. At 15% noise, all methods show degradation, but $PINN maintains lower epistemic uncertainty overall.

Scalability to Multiple Subdomains Figure 3 demonstrates Allen-Cahn with 3 and 4 subdomains. Interface uncertainty is slightly elevated but remains comparable to BPINN and 2-subdomain cases. Standard deviations are acceptable; performance can be enhanced with denser interface sampling. For the current small-scale problem, computational benefits are modest, but larger problems would benefit from parallelization across more subdomains.

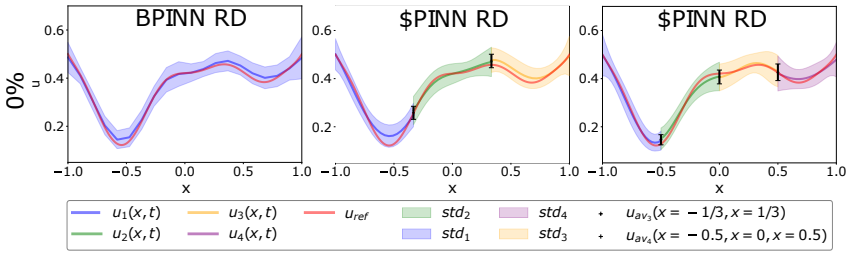

Figure 3: Allen-Cahn for BPINN and $PINN with 3 and 4 subdomains.

Inverse Problem We infer the diffusion coefficient $D = 0.01$ for Allen-Cahn using three approaches: no constraints, soft constraints, and hard constraints. Without constraints, subdomain parameters D_1 and D_2 diverge, especially at high noise, and $PINN underperforms BPINN. Soft constraints improve parameter consistency but remain insufficient. Hard constraints (single shared parameter) dramatically improve results: Figure 4 shows hard-constrained $PINN (pink) matches BPINN (orange) with 15% error across noise levels.

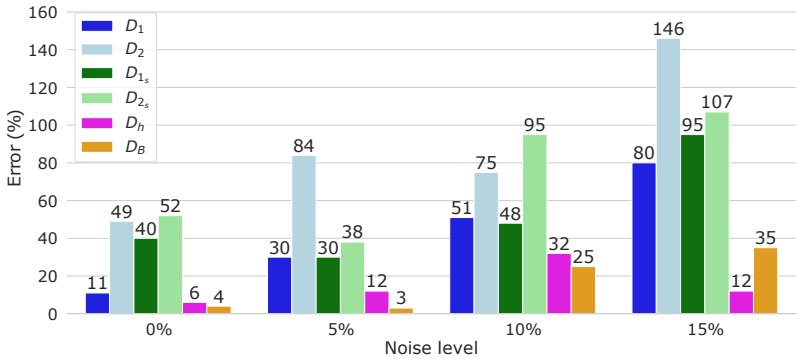

Figure 4: Absolute error (%) of diffusion coefficient recovery for Allen-Cahn: $PINN without constraints (blue), with soft constraints (green), with hard constraint (pink), and BPINN (orange).

3.1 Conclusion

These results demonstrate that $PINN successfully extends Bayesian PINNs to domain decomposition without sacrificing accuracy. Forward problems across all tested PDEs show that $PINN matches or exceeds BPINN performance while using fewer collocation points, with the BIC configuration consistently reducing interface uncertainty. Scalability to 3-4 subdomains is validated, though interface sampling density must scale accordingly. For inverse problems, hard constraints are essential—enabling parameter recovery comparable to BPINN with 12-15% error across noise levels. The method remains robust to variable domain sizes and asymmetric noise distributions, suggesting practical applicability to heterogeneous data scenarios.

ACKNOWLEDGMENTS

The work of Juliette Vanderhaeghen was supported by the ARC Project BiDiTwins of the Fédération Wallonie-Bruxelles (Belgium), under Grant 23/28-129, under the supervision of Sébastien Jodogne.

REFERENCES

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

Salvatore Cuomo, Vincenzo Schiano Di Cola, Fabio Giampaolo, Gianluigi Rozza, Maziar Raissi, and Francesco Piccialli. Scientific machine learning through physics–informed neural networks: Where we are and what's next. *Journal of Scientific Computing*, 92(3):88, 2022.

Zheyuan Hu, Ameya D. Jagtap, George Em Karniadakis, and Kenji Kawaguchi. When do extended physics-informed neural networks (XPINNs) improve generalization? *SIAM Journal on Scientific Computing*, 44(5):A3158–A3182, 2022. doi: 10.1137/21M1447039.

Ameya D. Jagtap, Ehsan Kharazmi, and George Em Karniadakis. Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering*, 365:113028, 2020. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2020.113028.

Axel Klawonn, Martin Lanser, and Janine Weber. Machine learning and domain decomposition methods-a survey. *Computational Science and Engineering*, 1(1):2, 2024.

Jinglai Li and Youssef M. Marzouk. Adaptive construction of surrogates for the Bayesian solution of inverse problems. *SIAM Journal on Scientific Computing*, 36(3):A1163–A1186, 2014. doi: 10.1137/130938189.

Stefano Markidis. The old and the new: Can physics-informed deep-learning replace traditional linear solvers? *Frontiers in big Data*, 4:669097, 2021.

Apostolos F. Psaros, Xuhui Meng, Zongren Zou, Ling Guo, and George Em Karniadakis. Uncertainty quantification in scientific machine learning: Methods, metrics, and comparisons. *Journal of Computational Physics*, 477:111902, 2023. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2022.111902.

Tijana Radivojević and Elena Akhmatskaya. Modified Hamiltonian Monte Carlo for Bayesian inference. *Statistics and Computing*, 30(2):377–404, 2020.

M. Raissi, P. Perdikaris, and G.E. Karniadakis. 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, 2019. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2018.10.045.

Khemraj Shukla, Ameya D. Jagtap, and George Em Karniadakis. Parallel physics-informed neural networks via domain decomposition. *Journal of Computational Physics*, 447:110683, 2021. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2021.110683.

Liang Yan and Tao Zhou. Adaptive multi-fidelity polynomial chaos approach to bayesian inference in inverse problems. *Journal of Computational Physics*, 381:110–128, 2019. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2018.12.025.

Liu Yang, Xuhui Meng, and George Em Karniadakis. B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data. *Journal of Computational Physics*, 425:109913, 2021.

A Appendix

A.1 Supplementary Experiments

The performance of $PINN is evaluated on three additional PDEs under varying noise levels.

Burgers Forward Problem. Figure 5 shows that $PINN achieves good approximations while using fewer training points, whereas BPINN requires more IC/BC and collocation points to approximate the solution accurately. Incorporating boundary, initial-condition, and interface data slightly increases overfitting but reduces uncertainty near the interface. At higher noise levels (e.g., 15%), BPINN becomes more sensitive, exhibiting increased variance and deviation, while $PINN remains more robust with lower uncertainty and closer agreement with the reference solution. When distributing random data points throughout the domain (not shown), both BPINN and $PINN improve their approximations.

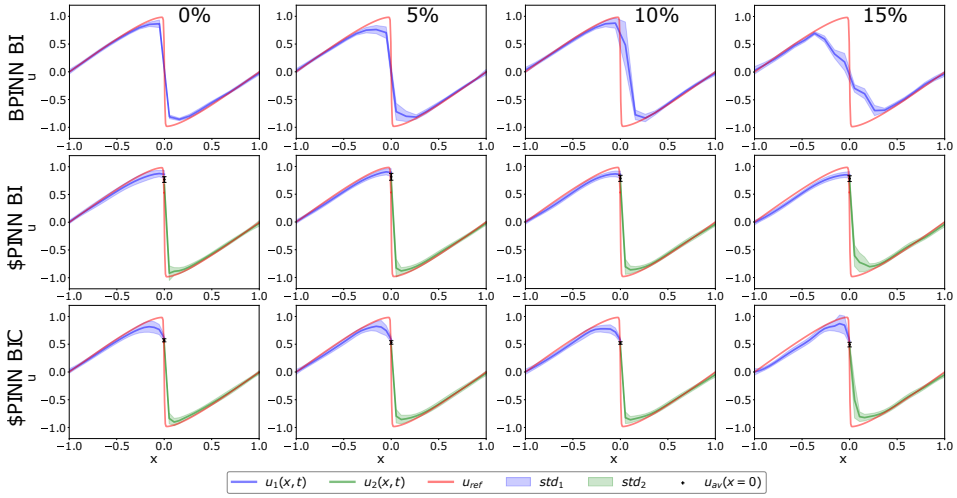

Figure 5: Burgers with BPINN and $PINN using IC, BC and CDC.

Fisher-KPP Forward Problem. Figure 6 presents a notable reduction in epistemic uncertainty after additional data points are added at the interface for $PINN BIC (bottom row). When using only random data points (not shown) without specifying the initial condition, boundaries, or interface, an increased epistemic uncertainty appears at the left boundary of the first subdomain and the right boundary of the second subdomain.

Fokker-Planck Forward Problem. As shown in Figure 7, including data points at the interface in $PINN again shows higher improvement. It significantly reduces epistemic uncertainty, which propagates across both subdomains. In this way, $PINN closely matches the reference solution, achieving results comparable to BPINN. As noise increases, $PINN BI progressively worsens its performance, showing a more significant effect of random uncertainty when 15% noise is reached. $PINN BIC also slightly suffers from the highest noise level. However, its predictions are comparable to those of the BPINN case. When only randomly distributed data points are used (not shown), both models become more prone to overfitting, predicting less accurate results with increasing noise levels.

Fokker-Planck Forward 2D Problem. Extending the implementation of $PINN to 2D spatial domains with time provides a more complex and realistic test bed. Figure 8 shows the validation of the method in a 2D space with time, (x, y, t), for the Fokker-Planck equation. With 300 residual points randomly distributed in space and time and 60 initial points randomly distributed in space, the forward problem is solved correctly without noise. The standard deviation is low in all solutions, but slightly higher when the gap between the subdomain at the interface and the domain boundaries

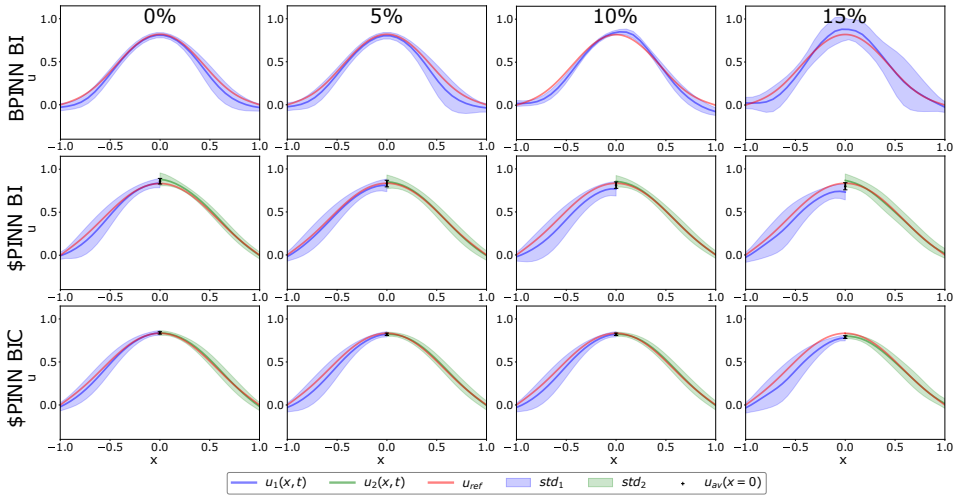

Figure 6: Fisher-KPP with BPINN and $PINN using IC, BC and CDC.

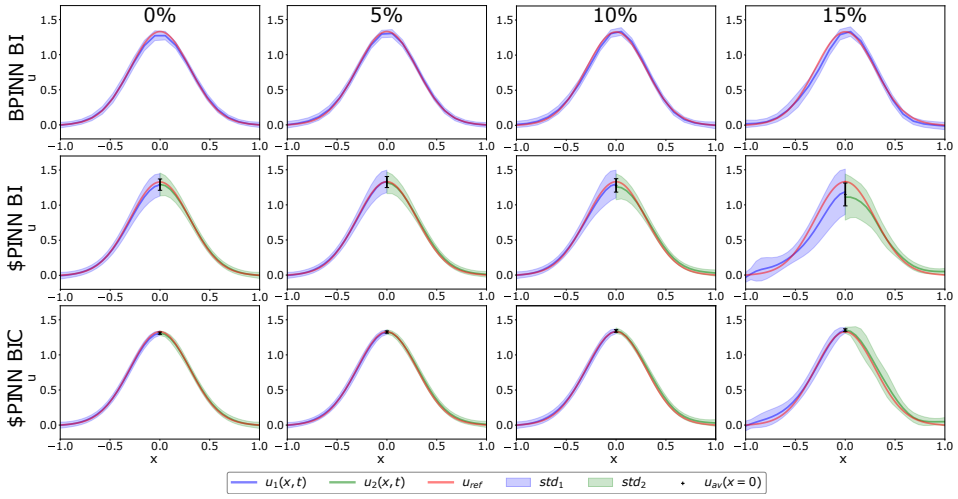

Figure 7: Fokker-Planck with BPINN and $PINN using IC, BC and CDC.

is larger. However, this test confirms that the use of $PINN in 2D is feasible. The use of a GPU and additional data points would enable even higher accuracy.

A.2 Future work

Future work could extend the evaluation of $PINN beyond comparisons with standard BPINN architectures. In particular, it would be useful to compare this method with other probabilistic frameworks that can solve PDEs and also offer uncertainty quantification, such as approaches based on Gaussian processes. Such comparisons would help clarify the trade-offs between different probabilistic PDE solvers in terms of predictive accuracy, uncertainty calibration, and computational cost.

Another important direction concerns the method's computational scalability. Since $PINN is based on domain decomposition, each subdomain model can, in principle, be trained independently. This structure naturally enables parallel training across multiple CPUs or GPUs, which could significantly reduce computational cost and improve training efficiency. In this way, we could further study the model's performance for more complex and larger domains in 2D and 3D. Investigating efficient parallelization strategies is therefore an important aspect of the method's development, and work in this direction is currently ongoing.

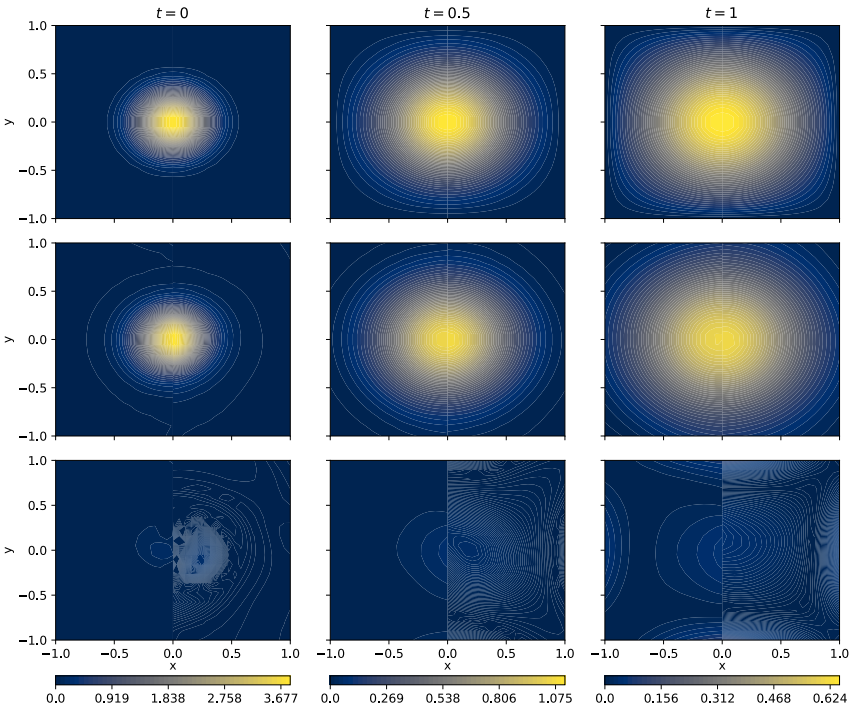

Figure 8: Fokker-Planck for BPINN and $PINN 2D forward problem using data points.