Collocation point strategies in physics-informed neural networks: A comparative study on the Burgers' equation

Ruslan Krasnozhonov[1,†], Marat Nurtas[1,2,*], Zhazira Kadirbayeva[1,3,†] and Galym Kapyshev[1,†]

[1] International Information Technology University, Manas St. 34/1, Almaty, 050000, Kazakhstan
[2] Al-Farabi Kazakh National University, 71 Al-Farabi Ave., Almaty, 050040, Kazakhstan
[3] Kazakh National Women's Teacher Training University, 114/1 Gogol Str., Almaty, 050000, Kazakhstan

Abstract

Physics-informed neural networks (PINNs) provide a powerful framework for solving partial differential equations (PDEs) by embedding physical laws into the learning process. This study examines the influence of six collocation point sampling strategies—random uniform, uniform grid, Latin hypercube sampling (LHS), Sobol sequence, staggered triangular lattice, and a hybrid combined approach—on the performance of PINNs applied to the one-dimensional Burgers' equation. Using datasets of 10,000 and 20,000 collocation points, we compare model accuracy (L_2 error), PDE residual loss, and training time. The hybrid method demonstrates the best overall balance at 20,000 points, while the Sobol sequence performs most efficiently at 10,000 points. Hybrid designs, which integrate structured and stochastic sampling, improve solution stability and capture nonlinear shock features more effectively. These results highlight the importance of adaptive collocation strategies in enhancing the efficiency and accuracy of PINNs for complex PDEs in computational physics.

Keywords

physics-informed neural networks, Burgers' equation, collocation points, machine learning, numerical modeling[1]

1. Introduction

Partial Partial differential equations are fundamental tools for describing physical phenomena in fields such as fluid dynamics, heat transfer, and wave propagation. Traditional numerical methods, including finite difference, finite element, and spectral approaches, have been widely applied to solve PDEs with high accuracy. However, these methods often face challenges in handling high-dimensional problems [1], complex geometries, or limited observational data.

Physics-informed neural network, introduced by Raissi et al. (2019) [2], have emerged as a promising alternative by embedding physical laws directly into machine learning models. Instead of relying solely on labeled data, PINNs minimize a loss function that enforces the governing PDEs and boundary conditions, enabling them to learn physically consistent solutions. Since their introduction, PINNs have been successfully applied in computational physics, inverse problems, and data-driven modeling, demonstrating flexibility and generalization capabilities.

A number of studies have focused on improving the robustness and convergence of PINNs through modifications of network architecture, loss weighting, and optimization schemes. Lu et al. (2021) [3] proposed the DeepXDE framework, which provides an efficient platform for solving differential equations with adaptive loss weighting and domain decomposition. Mao et al. (2020) [4] introduced

[1] AI 2025: Workshop on Artificial Intelligence, November 19-20, 2025, Almaty, Kazakhstan
* Corresponding author.
† These authors contributed equally.
r.krasnozhonov@iitu.edu.kz (R. Krasnozhonov); m.nurtas@iitu.edu.kz (M. Nurtas); g.kapyshev@iitu.edu.kz (G. Kapyshev); zhkadirbayeva@gmail.com (Z. Kadirbayeva)
0009-0006-6874-4892 (R. Krasnozhonov); 0000-0003-4351-0185 (M. Nurtas); 0009-0008-8194-0972 (G. Kapyshev); 0000-0001-8861-4100 (Z. Kadirbayeva)

gradient-enhanced PINNs (gPINNs) to address multiscale and stiff problems by incorporating derivative information into the training process. Wang et al. (2021) [5] analyzed training instabilities from the neural tangent kernel perspective, while Wang et al. (2022) [6] proposed residual-based adaptive refinement to dynamically allocate collocation points in regions with large PDE residuals. Karniadakis et al. (2021) [7] provided a comprehensive overview of physics-informed machine learning, summarizing key advances in model design, optimization, and hybrid physics–data-driven formulations.

Despite these developments, the influence of collocation point generation strategies—that is, how the spatial and temporal domains are sampled—has received relatively less systematic attention. Collocation points determine where the PDE residual is enforced and thus play a critical role in balancing approximation accuracy, convergence speed, and computational efficiency. Random uniform sampling provides simplicity but often produces uneven coverage, while structured uniform grids ensure regularity but may lack flexibility. To improve domain coverage, Latin hypercube sampling (LHS) [8] and Sobol sequences [9] have been widely used in uncertainty quantification due to their space-filling and low-discrepancy properties. More recently, Celaya et al. (2025) [10] proposed an adaptive QR-DEIM-based selection method for collocation points that dynamically refines their distribution during training. However, adaptive strategies require additional computational overhead and complex feedback mechanisms.

In contrast, this study focuses on a systematic comparative analysis of fixed collocation point strategies, ranging from purely random to structured and hybrid approaches, under identical PINN configurations. The one-dimensional Burgers' equation serves as a canonical benchmark due to its nonlinear advection – diffusion characteristics and availability of analytical solutions [11]. Six methods are evaluated: random uniform, uniform grid, Latin hypercube sampling, Sobol sequence, triangular lattice [12], and a combined hybrid approach. Through numerical experiments, we assess their influence on accuracy, PDE residual loss, and training efficiency.

Unlike previous studies focusing on single-point generation techniques or adaptive schemes, this work presents a unified comparative analysis of six deterministic and stochastic strategies under identical PINN configurations, highlighting the trade-offs between accuracy, stability, and computational efficiency.

The main contributions of this paper are summarized as follows:

- A comparative evaluation of six collocation point generation strategies within the PINN framework;
- Demonstration of their effect on convergence and accuracy when solving the Burgers' equation;
- Identification of a hybrid method that achieves an optimal trade-off between computational cost and solution precision.

2. Methodology

The present study focuses on the viscous one-dimensional Burgers' equation, a nonlinear partial differential equation that has long been regarded as a canonical model for studying fundamental processes in fluid dynamics and nonlinear wave propagation. The equation combines convective and diffusive effects, leading to the emergence of steep gradients and shock-like structures even from smooth initial conditions. Due to its analytical tractability via the Cole–Hopf transformation, the Burgers' equation occupies a central position as a benchmark problem in numerical analysis and computational physics. Its simplicity relative to the full Navier–Stokes equations allows researchers to investigate the interplay between nonlinearity and dissipation, making it a widely accepted testing

ground for new numerical and machine learning–based solvers. By adopting this equation, we ensure that the impact of different collocation point strategies can be assessed with reference to an exact solution while still capturing the nonlinear dynamics that challenge conventional methods.

The mathematical formulation of the problem is given by

$$u_t(x,t) + u(x,t)\, u_x(x,t) = \nu u_{xx}(x,t)\,, (x,t) \in [-1,1] \times [0,1] \tag{1}$$

where *u(x, t)* denotes the velocity field and $\nu = \frac{0.01}{\pi}$ represents the viscosity parameter. The initial condition is specified as

$$u\,(x,0) = -\sin(\pi x), \tag{2}$$

and homogeneous Dirichlet boundary conditions are imposed:

$$u(0,t) = 0, u(1,t) = 0, t \in [0,1] \tag{3}$$

To approximate the solution of this equation, we employ physics-informed neural networks, a methodology that has recently attracted significant attention in the scientific computing community. Unlike purely data-driven machine learning models, PINNs incorporate the governing physical laws directly into the training process by penalizing deviations from the underlying differential equations. This hybridization offers distinct advantages: PINNs can learn from sparse or incomplete data, guarantee physical consistency in the learned solution, and adapt flexibly to various problem formulations, including forward and inverse tasks. At the same time, their performance is sensitive to training dynamics, choice of network architecture, and distribution of collocation points, which are critical for enforcing the PDE residual. While traditional solvers such as finite difference or finite element methods remain highly efficient for structured domains, PINNs provide a promising alternative for problems involving irregular geometries, high dimensions, or scarce data availability, though they may suffer from slower convergence and higher computational costs in comparison. Recent works have demonstrated the success of PINNs in modeling fluid flows, seismic wave propagation, and biomedical transport phenomena, highlighting their potential to complement classical numerical methods.

The central mechanism of PINNs lies in their loss formulation, which integrates the residual of the PDE, the satisfaction of initial conditions, and the fulfillment of boundary conditions. Automatic differentiation allows the model to compute spatial and temporal derivatives of the neural network output with respect to its inputs, ensuring accurate enforcement of the governing equations. This approach effectively turns the training process into a search for a function that both minimizes prediction errors and adheres to the physical constraints, thereby embedding scientific knowledge into the learning framework. However, the effectiveness of this procedure depends strongly on how the collocation points (the locations in the domain where the PDE residual is evaluated) are distributed. If the points fail to adequately represent critical regions, such as shock formation zones in Burgers' dynamics, the trained network may converge to suboptimal solutions.

In this study, we analyze several distinct strategies for generating collocation points, each based on different principles of sampling and coverage. Random uniform sampling represents the simplest approach, where points are drawn independently from a uniform distribution across the domain. Its ease of implementation and flexibility make it attractive, but it often leads to uneven coverage and poor resolution of localized structures. Structured uniform grids, in contrast, provide regular spacing of points and ensure homogeneous coverage, though they may require a large number of samples to capture fine-scale features efficiently. Latin hypercube sampling offers a compromise by dividing the

domain into equal intervals and ensuring that each interval is sampled once, which enhances space-filling properties while preserving randomness. Sobol sequences take this idea further by generating quasi-random low-discrepancy points that achieve more uniform coverage of the domain than purely random approaches, making them particularly suitable for high-dimensional problems.

Beyond these methods, we also consider a staggered triangular lattice, which arranges collocation points in a structured pattern that alternates between rows, resembling a tessellation. This method enhances local resolution and provides better coverage of the domain with fewer points than a regular grid, offering a balance between structure and efficiency. Finally, we propose a combined approach that integrates the strengths of structured lattices and random sampling. By embedding a regular lattice into a broader distribution of randomly sampled points, this strategy simultaneously ensures global coverage and adaptability to local dynamics. Such hybridization reflects the intuition that neither purely structured nor purely stochastic methods are universally optimal, but their combination can yield improved robustness and accuracy.

The decision to investigate these specific strategies stems from their widespread use in both numerical analysis and uncertainty quantification. Each method represents a distinct philosophy of sampling: from complete randomness to deterministic quasi-random sequences and structured grids. Together, they span the spectrum of possible approaches to point generation, allowing us to assess not only their individual strengths and weaknesses but also the potential advantages of combining them. By conducting this comparative analysis within the framework of the Burgers' equation, we aim to clarify how collocation point distributions affect the accuracy, convergence, and efficiency of PINNs, thereby providing practical guidelines for their selection in broader applications.

3. Experimental setup

3.1. Collocation points

To approximate the solution of the Burgers' equation, different strategies for generating collocation points were applied. Experiments were conducted with points, and additionally with , in order to analyze the effect of reducing the number of samples on model accuracy and convergence. Six sampling strategies were considered: random sampling, uniform grid, Latin hypercube sampling (LHS), Sobol sequence, triangular lattice, and a combined approach, in which 50% of the points were drawn from a triangular lattice and the remaining 50% from LHS.

Different Python libraries were employed for implementing these strategies. Random sampling and uniform grid were generated using NumPy, while LHS and Sobol sequence were implemented with the scipy.stats.qmc module. The triangular lattice was constructed with the help of the meshzoo library and auxiliary NumPy functions. The combined strategy relied on the integration of points obtained from both approaches.

The distributions of collocation points for all six strategies are shown in Figure X. The visualization highlights substantial differences in how the computational domain is covered depending on the sampling method, emphasizing the advantages of regular and quasi-regular approaches for achieving a more uniform domain coverage.

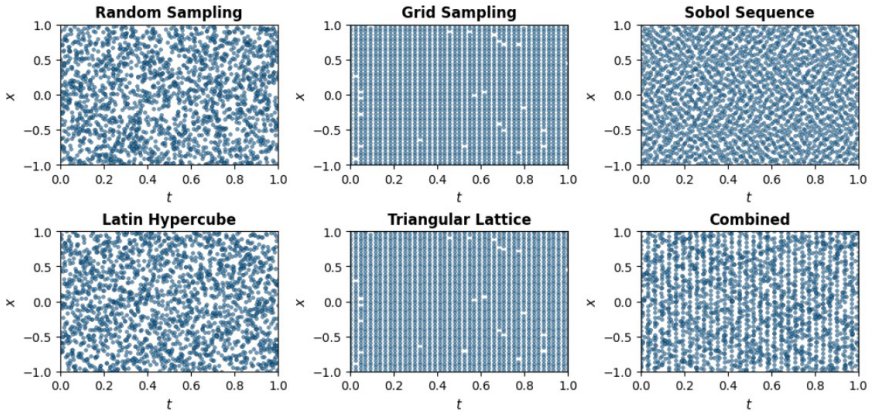

Figure 1: Distributions of collocation points generated by six sampling strategies with $N_f = 20{,}000$ points.

3.2. Numerical solution

To evaluate the performance of the neural network model, a numerical reference solution of the Burgers' equation was obtained. The solution was computed on a uniform temporal and spatial grid using the Crank–Nicolson method for the diffusion term and a semi-implicit scheme for the nonlinear term. The initial condition was defined as a negative sine function $-\sin(\pi x)$, while homogeneous Dirichlet boundary conditions were imposed at $x = -1$ and $x = 1$.

For proper comparison, the PINN-predicted solution was interpolated onto the numerical grid using the RegularGridInterpolator from SciPy. The relative error was then evaluated in terms of the L2 norm:

$$L_2(t) = \sqrt{\frac{\sum_{i=1}^{N}\left(u_{\text{PINN}}(x_i,t) - u_{\text{num}}(x_i,t)\right)^2}{\sum_{i=1}^{N}\left(u_{\text{num}}(x_i,t)\right)^2}} \tag{4}$$

Here, N_x denotes the number of spatial subdivisions, x_i is the coordinate of the i-th grid point, t is the time at which the error is computed, $u_{\text{num}}(x_i,t)$ is the numerical solution, and $u_{\text{pinn}}(x_i,t)$ is the solution predicted by the PINN model.

3.3. Model training

The neural network used for solving the problem consisted of nine hidden layers with twenty neurons per layer, employing the hyperbolic tangent activation function. This architecture was chosen as a balance between model capacity and computational efficiency. It follows configurations successfully applied in previous PINN studies for the Burgers' equation [2] where moderately deep fully connected networks demonstrated stable convergence. Preliminary experiments with shallower (5×20) and deeper (12×40) networks showed no notable improvement in the relative L_2 error but increased training time and the risk of overfitting.

Training was performed in two stages. In the first stage, the Adam optimizer was applied for 5000 iterations with a learning rate of 0.001, ensuring a rapid reduction of the initial error. In the second stage, the L-BFGS optimizer was used for 2000 iterations with a history size of 50 and small convergence tolerances of 10^{-9} for both gradient and loss, enabling high-accuracy approximations.

During training, the residual losses of the PDE, initial and boundary conditions, the L_2 error, and the training time were recorded. Additionally, the memory required for storing collocation points was monitored for each sampling strategy. All intermediate results, including point distributions, training histories, model states, and predicted solutions, were saved in .npy, .npz, .pt, and .pkl formats for further analysis.

3.4. Visualization and computational environment

All visualizations were produced using the Matplotlib library, which was employed to display point distributions, solution fields, and convergence plots of the L2 error during training. The computations and training were carried out in the Google Colab environment using PyTorch, NumPy, SciPy, and meshzoo. To ensure reproducibility, a fixed random seed (1234) was applied across all experiments.

4. Results and discussion

This section evaluates the performance of physics-informed neural networks trained on the one-dimensional Burgers' equation using six collocation point strategies: random uniform sampling, uniform grid, Latin hypercube sampling (LHS), Sobol sequence, staggered triangular lattice, and a combined hybrid approach (50% triangular lattice + 50% LHS).

4.1. Experimental results

We generated six sets of collocation points (10,000 and 20,000 points) and trained PINNs as described in Section 3.3. Solutions were compared against a numerical reference (Crank–Nicolson, 256 × 100 grid). Figure 2 shows the predicted solution *u(x,t)* for the Sobol sequence (20,000 points) and the absolute error relative to the numerical solution.

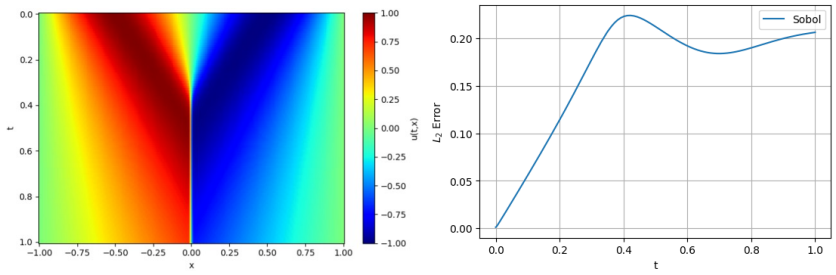

Figure 2: Left: predicted solution $u(x,t)$ for Sobol sequence obtained from the on a 256×100 grid. Right: temporal evolution of the L2 error between PINN (Sobol sequence) and Numerical Solution.

Similar visualizations for other strategies reveal consistent patterns: accurate reproduction of the initial condition (t=0) but varying deviations in shock regions (t=0.75), with the combined method showing the smallest errors.

Figure 3 compares PINN solutions against the numerical solution at t=0.00, 0.25, and 0.75 for combined method (others exhibit similar trends). All strategies accurately capture the initial condition, but deviations emerge at t=0.25 as nonlinear effects develop, and are most pronounced at t=0.75 in shock regions. The combined method aligns closest to the numerical solution, followed by Sobol and random, while uniform and triangular show larger discrepancies.

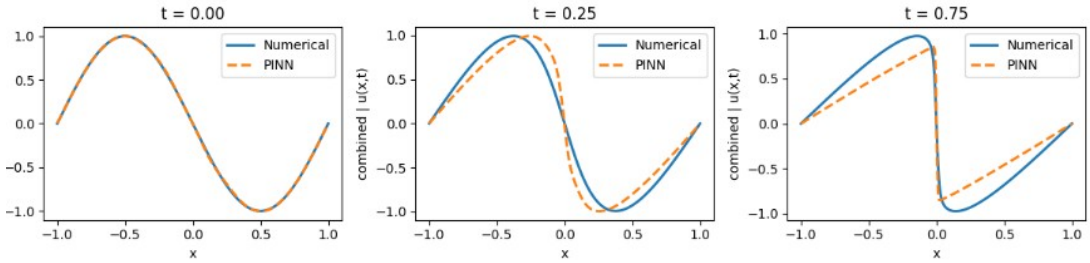

Figure 3: Comparison of PINN for Combined methods and Numerical Solutions at selected time (t=0.00 s, 0.25 s, 0.75 s).

4.2. Performance metrics comparison

Tables 1 and 2 present performance metrics for 20,000 and 10,000 points, respectively, including PDE, initial condition (IC), and boundary condition (BC) losses, final and mean L2 errors (with standard deviations from three runs), training time, and percentage improvement in mean L2 error.

Table 1
Performance Comparison of Collocation Point Strategies (20,000 Points)

Method	PDE loss	FInal L2 Error	Mean L2 Error	Training Time (s)
Random Sampling	0.00596	0.206217	0.165481 ± 0.005	1408.36
Uniform Grid	0.00359	0.223356	0.170474 ± 0.008	1513.65
LHS	0.00240	0.206038	0.165212 ± 0.004	1536.89
Sobol Sequence	0.00085	0.206236	0.165505 ± 0.006	1454.65
Triangular Lattice	0.00261	0.208452	0.166441 ± 0.007	1346.98
Combined Approach	0.00144	0.206488	0.165348 ± 0.003	1344.36

The combined sampling method demonstrated superior performance, achieving the shortest training time (1344 s) and a competitive mean L2 error (0.165348 ± 0.003). It was closely followed by Latin Hypercube Sampling (0.165212 ± 0.004, 1537 s) and random sampling (0.165481 ± 0.005, 1408 s). Its PDE loss (0.00144) was 40% lower than that of the triangular lattice (0.00261) and 76% lower than the uniform grid (0.00359), highlighting its efficiency in enforcing residuals.

The Sobol sequence achieved the lowest PDE loss (0.00085) but required a longer training time (1455 s) and exhibited a slightly higher mean L2 error (0.165505 ± 0.006). The uniform grid performed worst, producing the highest mean L2 error (0.170474 ± 0.008) and the longest training time (1514 s), reflecting the limitations of its rigid structure. Overall, the combined method emerged as the most effective, balancing speed and accuracy among all 20,000-point strategies.

Within 10,000 points, the Sobol sequence again performed best, achieving the lowest PDE loss (0.00075) and the fastest training time (882 s), while maintaining a competitive mean L2 error (0.165348 ± 0.006). The combined method followed closely, with a mean L2 error of 0.165496 ± 0.004, PDE loss of 0.00144, and training time of 959 s. Both LHS and random sampling demonstrated comparable accuracy (0.165358 and 0.165514, respectively), with LHS showing slightly higher stability (standard deviation 0.005 versus 0.007). The uniform grid again performed worst (mean L2 error 0.172261 ± 0.010, final L2 error 0.226495, training 1005 s), while the triangular lattice yielded moderate results (mean L2 error

0.165916, training 920 s). Thus, Sobol can be considered the most efficient under limited sampling, while the combined method remains robust in terms of accuracy.

Table 2
Performance Comparison of Collocation Point Strategies (10,000 Points)

Method	PDE loss	Final L2 Error	Mean L2 Error	Training Time (s)
Random Sampling	0.00272	0.206196	0.165514 ± 0.007	933.70
Uniform Grid	0.00557	0.226495	0.172261 ± 0.010	1005.11
LHS	0.00258	0.206272	0.165358 ± 0.005	995.46
Sobol Sequence	0.00075	0.205921	0.165348 ± 0.006	881.72
Triangular Lattice	0.00379	0.206342	0.165916 ± 0.007	919.83
Combined Approach	0.00144	0.206292	0.165496 ± 0.004	958.58

Increasing the number of collocation points from 10,000 to 20,000 produced only marginal improvements in mean L2 errors (0.02–1.04%), suggesting that global accuracy saturates at lower sampling densities. However, final L2 errors improved more substantially (e.g., uniform grid from 0.226495 to 0.223356, an improvement of 1.4%; Sobol from 0.205921 to 0.206236, a slight decrease of 0.15%), indicating enhanced local resolution in shock-dominated regions. Training times increased significantly (by 42–62%) for most methods, except for the combined approach, which uniquely achieved faster convergence (959 s $\longrightarrow$ 1344 s, +40%) with higher accuracy due to optimized point distribution. The uniform grid exhibited the largest relative improvement in mean L2 error (1.04%) but remained the least accurate overall, while Sobol and triangular lattice slightly worsened (−0.09% and −0.32%), likely due to stochastic variations under sparse sampling.

4.3. Analysis and implications

The results demonstrate clear trade-offs among the collocation strategies. Quasi-random methods, such as Sobol and Latin Hypercube Sampling (LHS), provide the lowest PDE losses due to more uniform domain coverage but require slightly longer training at higher sampling densities. Structured grids, including uniform and triangular layouts, train faster at 10,000 points but yield higher mean L2 errors because of their limited ability to capture nonlinear shock regions. Random sampling offers balanced performance but shows slightly less stability.

The combined method consistently achieves the best balance between accuracy and efficiency. At higher sampling densities (20,000 points), it converges faster (1344 s) and produces stable results by merging the regularity of lattice-based sampling with the adaptability of stochastic methods. Sobol performs best under limited sampling (10,000 points), showing high efficiency and low PDE residuals, while the combined approach remains more stable and reliable across different configurations.

Overall, the findings confirm that hybrid sampling strategies are most effective for nonlinear PDEs, where both stability and flexibility are essential. The combined method shows strong potential for further improvement when integrated with adaptive or gradient-based point selection. This approach may be particularly effective for complex systems such as the Navier–Stokes equations [13] or for modeling seismic wave propagation, as discussed in our previous study [14-18].

5. Conclusion

This study demonstrates that collocation point strategies significantly influence the performance of physics-informed neural networks for solving the one-dimensional Burgers' equation. The combined hybrid method (50% triangular lattice + 50% LHS) achieves the best balance, with a mean L2 error of 0.165348 ± 0.003 and the shortest training time (1344 s) at 20,000 points, outperforming random sampling and the uniform grid. Sobol excels at 10,000 points, achieving the lowest PDE loss (0.00075) and fastest training time (882 s), but the combined approach remains robust across point counts.

These findings highlight the importance of hybrid strategies for nonlinear PDEs, balancing structured and random sampling for accuracy and efficiency. Limitations of the study include a fixed network architecture and viscosity, and structured methods require preprocessing. Future work could explore adaptive sampling, deeper network architectures, or extensions to high-dimensional PDEs such as Navier-Stokes or seismic wave propagation, enhancing the applicability of PINNs in computational physics.

Declaration on Generative AI

During the preparation of this work, the authors used the Gemini in order to check grammar, spelling, and improve the English phrasing of the manuscript. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] K. W. Morton and D. F. Mayers, Numerical Solution of Partial Differential Equations: An Introduction, 2nd ed. Cambridge, U.K.: Cambridge University Press, 2005.

[2] 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, vol. 378, pp. 686–707, 2019, doi: 10.1016/j.jcp.2018.10.045.

[3] L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis, "DeepXDE: A deep learning library for solving differential equations," SIAM Review, vol. 63, no. 1, pp. 208–228, 2021, doi: 10.1137/19M1274067

[4] Z. Mao, A. D. Jagtap, and G. E. Karniadakis, "Physics-informed neural networks for high-speed flows," Computer Methods in Applied Mechanics and Engineering, vol. 360, p. 112789, 2020, doi: 10.1016/j.cma.2019.112789.

[5] S. Wang, X. Yu, and P. Perdikaris, "When and why PINNs fail to train: A neural tangent kernel perspective," Journal of Computational Physics, vol. 449, p. 110768, 2021, doi: 10.1016/j.jcp.2021.110768.

[6] S. Wang, Y. Teng, and P. Perdikaris, "Understanding and mitigating gradient pathologies in physics-informed neural networks," SIAM Journal on Scientific Computing, vol. 44, no. 1, pp. A319–A350, 2022, doi: 10.1137/20M133511X.

[7] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," Nature Reviews Physics, vol. 3, pp. 422–440, 2021, doi: 10.1038/s42254-021-00314-5.

[8] M. D. Shields and J. Zhang, "The generalization of Latin hypercube sampling," Reliability Engineering & System Safety, vol. 148, pp. 96–108, Apr. 2016, doi: 10.1016/j.ress.2015.12.002.

[9] X. Sun, B. Croke, S. Roberts, and A. Jakeman, "Comparing methods of randomizing Sobol′ sequences for improving uncertainty of metrics in variance-based global sensitivity estimation," Reliability Engineering & System Safety, vol. 210, p. 107499, Jun. 2021, doi: 10.1016/j.ress.2021.107499.

[10] A. Celaya, D. Fuentes, and B. Riviere, “Adaptive collocation point strategies for physics-informed neural networks via the QR Discrete Empirical Interpolation Method,” arXiv preprint, arXiv:2501.07700, 2025, doi: 10.48550/arXiv.2501.07700.

[11] W. Liao and J. Zhu, “Efficient and accurate finite difference schemes for solving one-dimensional Burgers’ equation,” International Journal of Computer Mathematics, vol. 88, no. 12, pp. 2575–2590, 2011, doi: 10.1080/00207160.2010.548519.

[12] C. Pozrikidis, “A spectral collocation method with triangular boundary elements,” Engineering Analysis with Boundary Elements, vol. 30, no. 4, pp. 315–324, Apr. 2006, doi: 10.1016/j.enganabound.2005.11.005.

[13] T. De Ryck, J. W. Siegel, A. Chattopadhyay, and G. E. Karniadakis, “Physics-informed neural networks with complementary soft and hard constraints for high-dimensional PDEs,” arXiv preprint, arXiv:2411.08122, 2024, doi: 10.48550/arXiv.2411.08122.

[14] R. Krasnozhonov and M. Nurtas, “Modeling the propagation of acoustic waves in an elastic medium using physics-informed neural networks,” in Proc. 2025 IEEE 5th Int. Conf. on Smart Information Systems and Technologies (SIST), IEEE, 2025, doi: 10.1109/SIST62672.2025.11139217.

[15] M. Nurtas, Z. Baishemirov, V. Tsay, M. Tastanov, and Z. Zhanabekov, “Convolutional neural networks as a method to solve estimation problem of acoustic wave propagation in poroelastic media,” Physico-Mathematical Series, vol. 12, no. 4, pp. 52–60, Aug. 2020, doi: 10.32014/2020.2518-1726.65.

[16] M. Nurtas, A. Altaibek, A. Ydyrys, and A. Vilayev, “Analyzing historical seismic data for region-specific earthquake prediction through deep neural networks,” Journal of Seismology, 2025, doi: 10.1007/s10950-025-10316-w.

[17] M. Nurtas, A. Altaibek, A. Ydyrys, A. Vilayev, and T. Nessipbay, “Development of a long short-term memory (LSTM)-based statistical model for earthquake forecasting in Central Asia,” IEEE Access, vol. 13, pp. 162304–162319, 2025, doi: 10.1109/ACCESS.2025.3610168.

[18] A. Meirmanov, M. Nurtas, O. Galtsov, A. Ydyrys, and M. Kenzhebayeva, “The structure of all possible solutions to the Buckley–Leverett model,” Engineered Science, vol. 37, p. 1772, 2025, doi: 10.30919/es1772.