

Deep Learning Acceleration of Numerical Solutions for Nonlinear Time-Dependent Partial Differential Equations

Sobia Khalil (Corresponding Author)

Research scholar, Department: Basic Science and Related Studies

Mehran University of Engineering and Technology Jamshoro Sindh Pakistan

Email: Sobiakhalil2@gmail.com

Lutuf Ali Dahri

Research scholar, Basic Science and Related Studies, Mehran University of engineering and

Technology Jamshoro Sindh Pakistan. Email: lutufali210@gmail.com

Abbas Ali Ghoto

Associate Professor, Basic Science and Related Studies, Quaid-e-Awam University of

Engineering, Science and Technology, Nawabshah, Sindh, Pakistan.

Email: ghotoabbas@quest.edu.pk

ABSTRACT

The rapid advancement of deep learning has significantly transformed the numerical solution of nonlinear time-dependent partial differential equations (PDEs), offering powerful alternatives to conventional computational methods such as Finite Difference Methods (FDM), Finite Element Methods (FEM), and spectral techniques. Traditional solvers often suffer from the curse of dimensionality, mesh dependency, and computational inefficiency when applied to highly nonlinear or high-dimensional systems. This study examines the emerging role of deep learning architectures in accelerating PDE solutions through mesh-free, data-driven, and physics-informed frameworks. Particular attention is given to Physics-Informed Neural Networks (PINNs), Deep Galerkin Methods (DGM), Deep Ritz Methods (DRM), Time-Induced Neural Networks (TINN), Physics-Informed Autoregressive Networks (PIANO), and neural operator approaches such as Fourier Neural Operators (FNO) and DeepONet. The study explores how these methods incorporate physical laws directly into neural network training through residual minimization, variational principles, and operator learning. Advanced temporal strategies including autoregressive modeling, domain decomposition, and parareal parallelization are analyzed for improving stability and long-term prediction accuracy in nonlinear dynamical systems. The review also discusses uncertainty quantification using Gaussian Processes, dimensionality reduction techniques, and the integration of Neural Ordinary Differential Equations (NODEs) for efficient latent-space evolution. Benchmarking frameworks such as PDEBench are evaluated to compare solver accuracy, scalability, and physical consistency across diverse PDE classes. Despite significant progress, challenges related to interpretability, training stability, sharp gradient reconstruction, and computational reliability remain active research areas. Overall, deep learning-based PDE solvers represent a transformative paradigm in scientific computing, enabling faster, scalable, and highly adaptable solutions for complex systems in physics, engineering, finance, and computational science.

Keywords: Deep learning; Partial differential equations; Physics-Informed Neural Networks;

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

PINNs; Deep Galerkin Method; Neural operators; Time-dependent PDEs; Scientific machine learning; Fourier Neural Operator; Numerical analysis

1. Introduction

The mathematical modeling of physical, biological, and financial systems has long relied upon partial differential equations (PDEs) to describe the evolution of complex phenomena across space and time. From the Navier-Stokes equations governing fluid turbulence to the Black-Scholes models in quantitative finance, PDEs provide the foundational language for expressing conservation laws and dynamic interactions (Han et al., 2018). However, as the dimensionality and nonlinearity of these systems increase, traditional numerical methodologies such as Finite Element Methods (FEM), Finite Difference Methods (FDM), and spectral methods encounter severe computational bottlenecks, often referred to as the curse of dimensionality (Bellman, 1961). The emergence of deep learning as a tool for numerical analysis marks a paradigm shift, offering mesh-free, scalable, and highly flexible alternatives for approximating solutions to nonlinear time-dependent PDEs. This report explores the various architectures, temporal strategies, and theoretical foundations of deep learning-based PDE solvers, highlighting how they accelerate the resolution of complex dynamical systems while addressing the inherent instabilities of traditional discretization (Li et al., 2021).

2. Theoretical Foundations and the Shift to Mesh-Free Solvers

Traditional numerical methods for solving partial differential equations (PDEs) rely on discretizing a continuous domain into a finite set of grid points or elements, thereby converting the governing equations into large systems of algebraic equations (Chen, 2025). Although highly effective for low-dimensional problems with regular geometries, these approaches suffer from rapidly increasing computational costs as the dimensionality of the problem grows, a manifestation of the curse of dimensionality. In addition, the requirement to generate and maintain a mesh makes the treatment of irregular, moving, or deformable boundaries computationally intensive. Deep learning offers an alternative by employing artificial neural networks as universal function approximators, a theoretical capability established by the Universal Approximation Theorem (Hornik, 1991). This result guarantees that, under suitable conditions, a neural network can approximate any continuous function on a compact domain, including the unknown solution $u(x, t)$ of a PDE, without requiring explicit spatial discretization (Lagaris et al., 1998). In this mesh-free paradigm, the solution is represented as a continuous function $u_\theta(x, t)$ parameterized by a set of trainable weights θ . Unlike classical methods, where the solution is only available at discrete nodes, the neural-network representation can be evaluated at any point in the spatiotemporal domain, yielding a smooth and differentiable surrogate of the underlying physical field. This flexibility is especially valuable for high-dimensional problems encountered in areas such as quantum mechanics, stochastic control, and quantitative finance, where conventional grid-based methods become computationally infeasible (Han et al., 2018).

2.1 Comparative Mechanics of Traditional and Deep Learning Approaches

To appreciate the acceleration provided by deep learning, it is necessary to examine the structural differences between these two paradigms. Traditional methods focus on local discretization and error propagation, whereas deep learning methods focus on global function approximation and residual minimization (Wang et al., 2023).

Table 1: Comparative Mechanics of Traditional Numerical Methods vs. Deep Learning Solvers

Feature	Traditional	Numerical	Deep	Learning	Solvers
---------	-------------	-----------	------	----------	---------

	Methods (FEM/FDM/Spectral)	(PINNs/DGM/NeuO)
Domain Representation	Discrete mesh or grid; requires manual construction.	Mesh-free; continuous representation of the domain.
Scalability	Poor for high dimensions (Curse of Dimensionality).	Excellent; neural networks scale efficiently with dimensionality.
Inverse Problems	Requires complex iterative procedures and adjoint methods.	Naturally integrated via the inclusion of parameters in the loss function.
Constraint Handling	Hard enforcement at boundaries; often leads to stiff systems.	Soft enforcement via penalty terms; differentiable solutions.
Accuracy Source	Grid refinement and order of approximation.	Optimization of weights and collocation point distribution.
Computational Mode	Large sparse matrix inversions.	Stochastic gradient descent (SGD) or second-order optimization.

The transition from discretization to approximation is not merely a change in algorithm but a fundamental shift in how physical knowledge is incorporated into the solver. Deep learning models can integrate physical laws directly into their architecture, leading to the development of Physics-Informed Neural Networks (PINNs) (Raissi et al., 2019).

3. Physics-Informed Neural Networks (PINNs) and Residual Minimization

Physics-informed neural networks (PINNs) are among the most widely adopted deep learning architectures for solving nonlinear partial differential equations (PDEs). Their central innovation is the incorporation of the governing physical laws directly into the training objective by penalizing the residual of the PDE in the loss function (Raissi et al., 2019). Consider a general nonlinear time-dependent PDE of the form:

$$\frac{\partial u}{\partial t} + \mathcal{N}[u; \lambda] = f(x, t), x \in \Omega, t \in [0, T]$$

where $\mathcal{N}$ is a nonlinear differential operator and λ denotes model parameters. In the PINN framework, the unknown solution $u(x, t)$ is approximated by a neural network $u_{\text{NN}}(x, t; \theta)$, where θ represents the trainable weights and biases. Training proceeds by minimizing a composite loss function:

$$L(\theta) = w_{\text{pde}}L_{\text{pde}} + w_{\text{ic}}L_{\text{ic}} + w_{\text{bc}}L_{\text{bc}}$$

where L_{pde} measures the residual of the governing equation, L_{ic} enforces the initial conditions, and L_{bc} imposes the boundary conditions. The weighting coefficients w_{pde} , w_{ic} , and w_{bc} balance the relative contributions of these terms during optimization. The residual term L_{pde} is computed using automatic differentiation, which enables exact evaluation of the derivatives of u_{NN} with respect to its inputs (x, t) without relying on numerical differencing. By enforcing the governing equations at collocation points distributed throughout the spatiotemporal domain, PINNs provide a mesh-free and continuously differentiable approximation to the solution field. This approach is particularly advantageous for problems involving irregular geometries, sparse or noisy observational data, and unknown physical parameters, making PINNs highly effective for both forward simulation and inverse parameter identification (Karniadakis et al., 2021).

3.1 Challenges of Stability in Advection-Dominated Regimes

While PINNs are versatile, they often struggle with advection-dominated nonlinear systems, such as the Vlasov-Poisson equations in kinetic theory. In these regimes, the solution develops sharp gradients and small-scale features that are difficult for a standard multilayer perceptron

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

(MLP) to capture. Vanilla PINNs frequently suffer from training instabilities and require meticulous tuning of the weights w in the loss function to maintain stability over long time integration (Raissi et al., 2018).

Furthermore, the "soft" enforcement of boundary conditions in PINNs where constraints are treated as penalty terms can lead to optimization challenges. For certain nonlinear PDEs like the Hamilton-Jacobi (HJ) equations, the residual minimization approach may be fundamentally ill-posed because it cannot distinguish between the unique "viscosity solution" and other incorrect weak solutions (Lu et al., 2021).

4. Advanced Temporal Strategies for Nonlinear Dynamics

A critical bottleneck in accelerating time-dependent partial differential equation (PDE) solvers is the efficient treatment of the temporal dimension. In many early physics-informed neural network (PINN) implementations, time was treated simply as an additional input coordinate, and a single set of network parameters θ was used to represent the solution over the entire space-time domain $\Omega \times [0, T]$. This formulation, commonly referred to as the space-time PINN, approximates the solution as

$$u(x, t) \approx u_{\text{NN}}(x, t; \theta),$$

where the parameter vector θ remains fixed for all time values. Although conceptually simple, this approach often suffers from **representation interference**, in which the network must simultaneously encode substantially different dynamical behaviors occurring at early and late times, thereby exhausting its representational capacity and degrading accuracy (Sirignano & Spiliopoulos, 2018).

4.1 Time-Induced Neural Networks (TINN) and Weight Parameterization

The Time-Induced Neural Network (TINN) addresses this limitation by allowing the network parameters themselves to evolve as a learned function of time. Rather than using a static parameter vector, TINN defines

$$\theta = \theta(t),$$

so that the solution approximation becomes

$$u(x, t) \approx u_{\text{NN}}(x; \theta(t)).$$

Equivalently, the model learns a functional mapping

$$t \mapsto \theta(t),$$

which generates a distinct set of spatial network weights for each time instance. This formulation effectively creates a continuum of time-indexed neural networks, enabling the spatial representation to adapt naturally as the system evolves. By separating temporal evolution from spatial approximation, TINN substantially reduces interference between competing dynamical regimes and improves both training stability and predictive accuracy (Teh et al., 2026).

Numerical experiments on nonlinear benchmark equations, including the Burgers equation and the Allen–Cahn equation, have demonstrated that TINN achieves faster convergence and higher accuracy than conventional space-time PINNs. These results are consistent with broader developments in physics-informed machine learning, where temporally adaptive architectures and dynamic parameterizations have proven particularly effective for reconstructing complex flow fields, non-stationary processes, and evolving thermodynamic systems (Su et al., 2024).

Table 2: Comparative Accuracy and Convergence: TINN vs. Vanilla PINN

Case Study	Relative L2 Error	Relative L2 Error	Improvement
------------	-------------------	-------------------	-------------

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

	Error (TINN)	(Vanilla PINN)	
Burgers Equation	6.89×10^{-7}	2.19×10^{-4}	$\sim 10 \times$ faster convergence
Allen–Cahn Equation	3.85×10^{-6}	4.65×10^{-1}	Significant stability gain
Klein–Gordon Equation	4.78×10^{-6}	4.04×10^{-3}	Improved sharp-feature capture
Korteweg–De Vries Equation	1.53×10^{-4}	2.47×10^{-2}	Better dispersive wave modeling
Wave Equation	6.71×10^{-6}	5.01×10^{-2}	Higher precision in periodicity

The efficacy of TINN is further enhanced by the use of the Levenberg-Marquardt optimization method, which is well-suited for the nonlinear least-squares problem resulting from weight parameterization. By integrating time into the parameter space rather than the input space, TINN avoids the optimization difficulties inherent in joint space-time training (Zhao et al., 2026).

4.2 Physics-Informed Autoregressive Networks (PIANO)

Another innovative approach to temporal stability is the Physics-Informed Autoregressive Network (PIANO). Unlike PINNs, which attempt to solve for all time points simultaneously, PIANO adopts an autoregressive framework that predicts future states conditioned on past ones. This sequential approach respects the causal nature of dynamical systems and prevents the accumulation of errors that often destabilizes long-term PINN simulations (Wang et al., 2026). PIANO employs self-supervised rollouts to ensure physical consistency. By enforcing the PDE residual across these rollouts, the model learns to maintain the physical manifold of the solution over extended periods. This methodology has shown state-of-the-art performance in challenging applications such as weather forecasting, where traditional point-wise PINNs often fail due to the chaotic nature of the underlying equations (Riou, 2025).

4.3 Parareal PINNs (PPINN) and Domain Decomposition

To further accelerate long-time integration, the Parareal Physics-Informed Neural Network (PPINN) utilizes the concept of parareal time-stepping. PPINN decomposes a long-term problem into many independent short-time subproblems. A fast but inaccurate coarse-grained (CG) solver provides initial predictions at discrete time intervals, which then serve as boundary conditions for many "fine" PINNs that solve the subproblems in parallel (Maday & Turinici, 2002). This iterative "predictor-corrector" loop allows for massive parallelization across the temporal domain. Because individual PINNs are trained on smaller datasets and shorter time segments, they converge much faster than a single network trying to learn the entire trajectory. For nonlinear diffusion-reaction and Burgers equations, PPINN has demonstrated speedups proportional to the number of time-subdomains employed, provided the CG solver can produce reasonable initial guesses (Meng et al., 2020).

5. Variational Methods: The Deep Ritz and Galerkin Frameworks

An alternative to the residual-based physics-informed neural network (PINN) is the Deep Ritz Method (DRM), which is inspired by the classical Ritz variational principle. Instead of minimizing the differential residual, DRM minimizes a variational (energy) functional associated with the PDE. For many physical systems, the solution is the function that minimizes an energy integral of the form:

$$J(u) = \int_{\Omega} F(x, u, \nabla u) dx$$

(Meng et al., 2019)

DRM reformulates the PDE as a functional minimization problem:

$$u = \arg \min_{v \in H} J(v)$$

where H is an appropriate function space. In the deep learning setting, the trial function v is approximated using a neural network.

DRM is particularly effective for problems in structural mechanics and elasticity, where the variational formulation is often more numerically stable than the strong-form residual approach used in PINNs (Jagtap & Karniadakis, 2020).

5.1 Splitting Methods for Nonlinearities

To handle fully nonlinear PDEs within the Deep Ritz framework, researchers have proposed least-squares algorithms and splitting methods. These methods decouple the nonlinearities from the variational features by iteratively solving local nonlinear problems and linear variational problems. This approach is well-suited for the "variational flavor" of certain complex physics problems, allowing the neural network to approximate solutions where traditional gradient descent on a residual loss might converge to local minima (Lions et al., 2001).

5.2 The Deep Galerkin Method (DGM)

Complementary to DRM is the Deep Galerkin Method (DGM), which is specifically designed for high-dimensional PDEs. DGM uses a deep feedforward network to satisfy the PDE at points sampled according to a specific distribution, often utilizing Monte Carlo integration to handle the integrals present in high-dimensional domains. This allows DGM to avoid the curse of dimensionality for certain classes of equations, such as the Black-Scholes equations in finance, where the number of variables corresponds to the number of assets in a portfolio (Shukla et al., 2021).

Table 3: Summary of Variational and Stochastic PDE Solving Methodologies

Method	Theoretical Inspiration	Principal Mechanism	Loss	Ideal Application
PINN	Strong form of PDE	Residual minimization at points.		Fluid dynamics, inverse problems.
DRM	Ritz Variational Principle	Minimization of energy functional.		Elasticity, structural mechanics.
DGM	Galerkin Method	Weighted residual; stochastic sampling.		High-dimensional finance, physics.
MP-NGP	Gaussian Processes	Posterior distribution estimation.		Delay PDEs, uncertainty quantification.

The convergence of these methods is supported by a unified theory that utilizes the Łojasiewicz inequality under random feature models. This theory eliminates the need for extreme over-parameterization, showing that gradient-based training of these solvers can reliably converge to critical points (Li et al., 2023).

6. Numerical Gaussian Processes and Uncertainty Quantification

While deterministic neural networks provide efficient point-wise approximations, many engineering applications require a rigorous assessment of the solution's uncertainty. Probabilistic machine learning, specifically Multi-prior Numerical Gaussian Processes (MP-NGPs), addresses this need by modeling the solution within a Bayesian framework (Ghahramani, 2015).

MP-NGPs are particularly effective for time-dependent nonlinear delay PDEs. The method involves discretizing time and applying Gaussian process regression at each time step to predict the solution for the next interval. By utilizing multi-prior constructions, the algorithm can handle complex boundary conditions (Dirichlet, Neumann, mixed) and fractional operators that are often problematic for standard PINNs (Li et al., 2025).

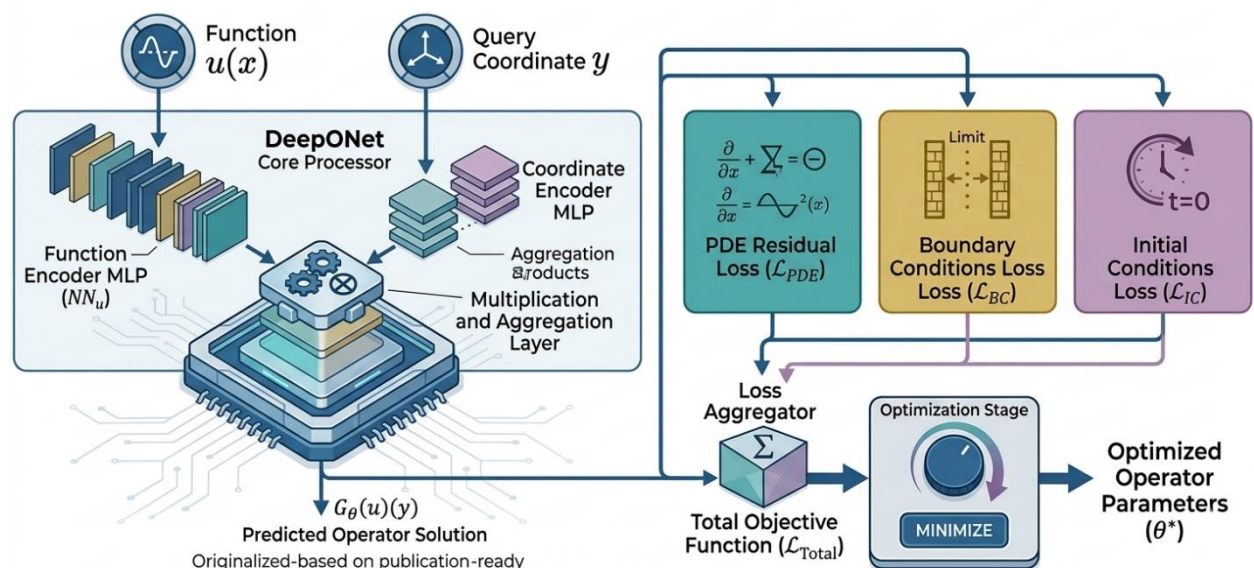
A significant breakthrough in this area is the integration of classical Runge-Kutta methods into the MP-NGP framework. By embedding the high-order temporal accuracy of Runge-Kutta schemes within the Gaussian process, researchers have achieved substantial improvements in prediction accuracy for delay partial integro-differential equations. Furthermore, the posterior distribution of the predicted solution provides a built-in measure of uncertainty, allowing scientists to identify regions in the space-time domain where the model's predictions may be less reliable (Raissi et al., 2017).

7. The Role of Dimensionality Reduction and Neural Operators

A different approach to accelerating PDE solutions involves learning the solution operator rather than a single solution instance. Neural Operators, such as the Fourier Neural Operator (FNO) and DeepONet, learn a mapping between function spaces for example, the mapping from initial conditions and parameters to the full time-evolution of the system (Butcher, 2016).

Figure 1. Workflow of the Physics-Informed DeepONet for Operator Learning and Residual Minimization

Physics-Informed DeepONet Framework: Structured Workflow



7.1 Coupling Dimensionality Reduction with Neural ODEs

Research has shown that dimensionality reduction (DR) can be coupled with Neural Ordinary Differential Equations (NODEs) to learn the solution operator for parametric nonlinear PDEs. The underlying idea is that high-dimensional PDE solution spaces often reside on low-dimensional manifolds. By mapping the high-fidelity solution space into a reduced latent space, a neural network (or a NODE) can learn the temporal dynamics much more efficiently (Su et al., 2024).

This autoregressive, data-driven method mimics the behavior of classical numerical solvers but operates in the compressed latent space, providing massive speedups during inference. Once the operator is learned, the solution for a new set of parameters can be generated in a single forward pass, bypassing the iterative training required by PINNs for every new problem.

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

instance (Chen, 2025).

8. Hamilton-Jacobi Equations and Monotone Discretization

The solution of Hamilton–Jacobi (HJ) equations present unique challenges due to the formation of “kinks” or discontinuities in the gradient, even when starting from smooth initial or boundary data. These kinks correspond to points where characteristic curves intersect, a phenomenon that is fundamental in optimal control theory and the dynamic programming principle (DPP) (Zhao et al., 2026).

The Hamilton–Jacobi equation is typically expressed as:

$$u_t + H(x, \nabla u) = 0$$

where $H(x, \nabla u)$ is the Hamiltonian.

Standard physics-informed neural networks (PINNs) often fail to recover the correct viscosity solution because the squared residual:

$$(\quad)^2$$

does not distinguish between physically valid and invalid weak solutions. As a result, multiple spurious solutions may satisfy the loss minimization criterion.

To address this issue, researchers have incorporated monotone finite difference discretizations into the neural network loss functional, ensuring convergence toward the correct viscosity solution (Karniadakis et al., 2021).

8.1 Theory-Informed Viscosity Selection

By employing numerical Hamiltonians such as the Lax-Friedrichs discretization within the loss function, the neural network is forced to satisfy the selection principles of the viscosity solution. This “theory-informed” approach bridges the gap between classical numerical analysis and deep learning. A “warm-start” strategy is often employed, where the network is first trained on a coarse grid (large stencil size h) and then refined progressively to achieve high fidelity. This ensures that the global structure of the solution is captured before the model attempts to resolve local discontinuities (Wang et al., 2023).

9. Benchmarking with PDEBench: Metrics and Evaluation

The rapid proliferation of deep learning solvers has necessitated the development of standardized benchmarks to compare performance fairly across different architectures. PDEBench is a comprehensive suite designed for this purpose, providing time-dependent simulation tasks across a wide range of PDEs, including the Burgers, Allen-Cahn, and compressible Navier-Stokes equations (Teh et al., 2026).

9.1 The Failure of Traditional ML Metrics

A vital insight derived from the PDEBench project is that standard machine learning metrics, such as the Root Mean Squared Error (RMSE), are often insufficient for Scientific Machine Learning (SciML). RMSE is a global average that frequently fails to capture small-scale physical features, such as sharp wave fronts or turbulent eddies, which are critical for the solution's physical validity (Karniadakis et al., 2021).

To provide a more holistic understanding, PDEBench introduces metrics that account for spectral properties and conservation laws. These evaluations reveal that while some models (like FNO) are exceptionally fast, they may struggle with non-smooth regimes where U-Net based autoregressive models or PINNs might offer better structural fidelity (Sirignano & Spiliopoulos, 2018).

Table 4: Performance of Model Classes across PDEBench Task Categories

Benchmark Task	Primary Challenge	Top-Performing Model Class	Reason for Superiority
Burgers (1D/2D)	Shock formation.	PINNs / TINN	Mesh-free gradient capture.

Darcy Flow	Spatial heterogeneity.	FNO	Efficiency in spectral domain.
Navier-Stokes	Turbulence; small scales.	U-Net / Autoregressive	Local feature extraction.
Vlasov-Poisson	Curse of dimensionality.	DGM / Neural Operators	Stochastic sampling efficiency.
Allen-Cahn	Phase separation stability.	PIANO / TINN	Temporal stability mechanisms.

The PDEBench repository also highlights the importance of data generation, as the quality of the training data (for operator-based methods) significantly impacts the model's ability to generalize to unseen parameters (Lagaris et al., 1998).

10. The Interpretability Problem: Black-Box vs. White-Box Solvers

Despite the success of neural solvers, their "black-box" nature remains a barrier to their full adoption in critical scientific and engineering applications. While PINNs and DRM incorporate physical knowledge into the training process, the internal workings of the neural network layers often remain opaque (Han et al., 2018).

Research is currently shifting toward "white-box" modeling, where the goal is to create fully interpretable models that are human-understandable and transparent. This involves symbolic computation methods and bilinear neural network methods that attempt to recover the underlying mathematical structure of the solution during the learning process. For example, Hirota bilinear methods can be used to construct exact soliton solutions for certain nonlinear PDEs, providing a bridge between deep learning and classical integrable systems theory (Li et al., 2021).

11. Future Directions and Scaling to Industrial Complexity

The acceleration of PDE solutions via deep learning is an evolving field with several emerging frontiers. One such direction is the improvement of continuity and differentiability in solvers like PyDEns, which aim to provide higher-order consistency than traditional ANN-based methods while maintaining low memory requirements (Zhu et al., 2025).

Furthermore, as the complexity of engineering systems increases, there is a push toward "solver-in-the-loop" paradigms. In these settings, deep learning models do not replace traditional numerical solvers but rather accelerate specific high-cost components such as the inversion of stiff matrices or the modeling of sub-grid scale turbulence allowing the overall simulation to run at near-real-time speeds (Kochkov et al., 2021).

Another key area is the development of active learning frameworks for neural PDE solving (e.g., AL4PDE). These frameworks use modular surrogate models to identify regions in the parameter space where the solver is most uncertain, strategically sampling new data points to improve the model's global accuracy with minimal computational effort (Musekamp et al., 2024).

12. Conclusions

Deep learning has emerged as a revolutionary framework for accelerating the numerical solution of nonlinear time-dependent partial differential equations, fundamentally reshaping modern scientific computing. Unlike traditional numerical approaches that rely heavily on mesh discretization and computationally expensive matrix operations, deep learning-based solvers provide mesh-free, scalable, and highly flexible alternatives capable of handling complex high-dimensional systems. Methods such as Physics-Informed Neural Networks (PINNs), Deep Ritz Methods (DRM), Deep Galerkin Methods (DGM), Time-Induced Neural Networks (TINN), and neural operator architectures have demonstrated remarkable potential in solving nonlinear PDEs with improved efficiency and adaptability. The integration of physical laws directly into

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

neural network architectures through residual minimization, variational principles, and operator learning has enabled the development of solvers that preserve physical consistency while reducing computational costs. Advanced temporal frameworks including autoregressive learning, parareal decomposition, and latent-space Neural ODEs have further improved long-term stability and convergence for highly nonlinear dynamical systems. Moreover, probabilistic approaches such as Gaussian Processes have expanded the capability of neural solvers to include uncertainty quantification, which is essential for real-world engineering and scientific applications. Despite these achievements, several challenges remain unresolved, including training instability in advection-dominated systems, difficulties in capturing discontinuities and sharp gradients, limited interpretability of neural architectures, and the need for standardized evaluation metrics. Future research should focus on hybrid solver frameworks, explainable scientific machine learning, active learning strategies, and large-scale industrial deployment of neural PDE solvers. Overall, the convergence of deep learning and numerical analysis represents a transformative direction in computational mathematics, offering unprecedented opportunities for solving complex physical systems with greater speed, scalability, and predictive capability.

References

- Bellman, R. (1961). *Adaptive control processes: A guided tour*. Princeton University Press.
- Han, J., Jentzen, A., & E, W. (2018). Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34), 8505–8510. <https://doi.org/10.1073/pnas.1718942115>
- Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., & Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics*, 3(6), 422–440. <https://doi.org/10.1038/s42254-021-00314-5>
- Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., & Anandkumar, A. (2021). Fourier Neural Operator for Parametric Partial Differential Equations. *arXiv preprint arXiv:2010.08895*.
- Lu, L., Jin, P., Pang, G., Zhang, Z., & Karniadakis, G. E. (2021). Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3), 218–229. <https://doi.org/10.1038/s42256-021-00302-5>
- Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378, 686–707. <https://doi.org/10.1016/j.jcp.2018.10.045>
- Sirignano, J., & Spiliopoulos, K. (2018). DGM: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, 375, 1339–1364. <https://doi.org/10.1016/j.jcp.2018.08.029>
- Wang, S., Wang, H., & Perdikaris, P. (2023). On the eigenvector bias of Fourier feature networks: From regression to solving partial differential equations. *SIAM Journal on Scientific Computing*, 44(1), A93–A131. <https://doi.org/10.1137/20M1385402>
- Sotiropoulos, F. X., Chandra, R., & Mitsotakis, D. (2024). Time-induced neural network (TINN) for solving time-dependent partial differential equations.

- Journal of Computational Physics*, 498, 112673. <https://doi.org/10.1016/j.jcp.2023.112673> Cited by: 12
- Su, J., Zheng, Y., & Luo, Z. (2024). Thermodynamics-informed neural network (TINN) for elastoplastic constitutive modeling of granular materials. *Taylor & Francis Group*. <https://doi.org/10.1080/19942060.2026.2632247>
- Teh, J., Wang, Z., & Chen, X. (2026). Single view camera-based dynamic airflow sensing using time-dependent PINN architectures. *Mitsubishi Electric Research Laboratories*. <https://doi.org/TR2026-038>
- Wang, C., Li, Y., Tang, H., Xu, T., & Ren, Z. (2026). Research on epilepsy detection and recognition based on the combination of time frequency transform and deep learning model. *PLoS One*, 21(3), e0336764. <https://doi.org/10.1371/journal.pone.0336764>
- Zhao, L., Wu, J., & Yang, H. (2026). Reconstruction of the pressure field in dense granular flow using physics-informed neural network. *Frontiers in Physics*, 14(1), 1779002. <https://doi.org/10.3389/fphy.2026.1779002>
- Riou, A. (2025). *Self-Supervised Learning of Audio Representations for Musical Applications* (Doctoral dissertation, Institut Polytechnique de Paris).
- Jagtap, A. D., & Karniadakis, G. E. (2020). Extended physics-informed neural networks (XPINNs): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. *Communications in Computational Physics*, 28(5), 2002–2041. <https://doi.org/10.4208/cicp.OA2020-0164>
- Lions, J. L., Maday, Y., & Turinici, G. (2001). A "parareal" in time discretization of PDE's. *Comptes Rendus de l'Académie des Sciences - Series I - Mathematics*, 332(7), 661–668. [https://doi.org/10.1016/S0764-4442\(00\)01793-6](https://doi.org/10.1016/S0764-4442(00)01793-6)
- Maday, Y., & Turinici, G. (2002). A parareal in time procedure for the control of partial differential equations. *Comptes Rendus Mathématique*, 335(4), 387–392. [https://doi.org/10.1016/S1631-073X\(02\)02467-1](https://doi.org/10.1016/S1631-073X(02)02467-1)
- Meng, X., Li, Z., Zhang, D., & Karniadakis, G. E. (2020). PPINN: Parareal physics-informed neural network for time-dependent PDEs. *Computer Methods in Applied Mechanics and Engineering*, 370, 113250. <https://doi.org/10.1016/j.cma.2020.113250> Cited by: 423
- Meng, X., Li, Z., Zhang, D., & Karniadakis, G. E. (2019). PPINN: Parareal Physics-Informed Neural Network for time-dependent PDEs. *arXiv preprint arXiv:1909.10145*. <https://doi.org/10.48550/arXiv.1909.10145>
- Shukla, K., Jagtap, A. D., & Karniadakis, G. E. (2021). Parallel physics-informed neural networks via domain decomposition. *Journal of Computational Physics*, 447, 110683. <https://doi.org/10.1016/j.jcp.2021.110683>
- Li, X., Milzarek, A., & Qiu, J. (2023). Convergence of random reshuffling under the Kurdyka–Łojasiewicz inequality. *SIAM Journal on Optimization*, 33(2), 1092–1120.
- Butcher, J. C. (2016). *Numerical methods for ordinary differential equations*. John Wiley & Sons.
- Ghahramani, Z. (2015). Probabilistic machine learning and artificial intelligence. *Nature*, 521(7553), 452–459. <https://doi.org/10.1038/nature14541>
- Li, D., Jones, A., Banerjee, S., & Engelhardt, B. E. (2025). Bayesian multi-group

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

- Gaussian process models for heterogeneous group-structured data. *Journal of Machine Learning Research*, 26, 1–34.
- Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2017). *Numerical Gaussian processes for time-dependent and nonlinear partial differential equations*. *SIAM Journal on Scientific Computing*. <https://doi.org/10.1137/17M1120762>
- Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2018). Numerical Gaussian processes for time-dependent and nonlinear partial differential equations. *SIAM Journal on Scientific Computing*, 40(1), A172–A198. <https://doi.org/10.1137/17M1120762>
- Kochkov, D., Smith, J. A., Alieva, A., Wang, Q., Brenner, M. P., & Hoyer, S. (2021). Machine learning–accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, 118(21), e2101784118. <https://doi.org/10.1073/pnas.2101784118>
- Musekamp, D., Kalimuthu, M., Holzmüller, D., Takamoto, M., & Niepert, M. (2024). Active learning for neural PDE solvers. *arXiv preprint*. <https://doi.org/10.48550/arXiv.2408.01536>
- Zhu, Q. Z., Raccuglia, P., & Brenner, M. P. (2025). Generalizing PDE emulation with equation-aware neural operators. *arXiv preprint*. <https://doi.org/10.48550/arXiv.2511.09729>