International Journal of Multidisciplinary Research and Publications
ISSN (Online): 2581-6187

A Review of Physics-Informed Kolmogorov-Arnold Networks for Solving Partial Differential Equations

Xiuliang Yu
School of Management Science and Engineering, Anhui University of Finance and Economics
Email address: 2261673874@qq.com

Abstract*—Partial differential equations are important mathematical tools for describing physical and engineering systems. Physical information neural networks provide meshless deep learning methods for solving partial differential equations by embedding control equations, boundary conditions, and initial conditions into loss functions. However, traditional PINNs typically use multi-layer perceptrons as function approximators, which can lead to issues such as insufficient expression efficiency, unstable training, and weak interpretability in high-frequency, multi-scale, and locally singular problems. The Kolmogorov Arnold network, based on the Kolmogorov Arnold representation theorem, sets learnable one-dimensional functions on the edges of the network, providing a new idea for improving the function expression ability of PINNs. This article focuses on the physical information deep learning framework PIKAN based on KAN, systematically reviewing its theoretical basis, network structure, physical constraint mechanism, and research progress, and analyzing its advantages, limitations, and future development directions in solving complex partial differential equations. The review indicates that PIKAN has potential in multi-scale modeling, local complex function approximation, and model interpretability, but its computational efficiency, training stability, and adaptability to complex geometry still need further research.*

***Keywords*—** *Kolmogorov Arnold network; PIKAN ; Physical information neural network; Partial differential equations; Scientific Machine Learning*

I. INTRODUCTION

Partial differential equation is an important mathematical tool to describe the evolution law of natural science and engineering system. It is widely used in the fields of fluid mechanics, solid mechanics, heat conduction, electromagnetic field, material damage and multi physical field coupling. For many practical engineering problems, the physical field variables are often affected by space location, time, material parameters, boundary conditions and external loads at the same time, so it is usually necessary to establish a mathematical model through partial differential equations and their initial boundary conditions. Traditional numerical methods such as finite element method, finite difference method, finite volume method and boundary element method have formed a mature system in scientific calculation and engineering simulation, but in high-dimensional problems, multi-scale characteristics, strong nonlinear coupling, complex geometric regions and parameter inversion tasks, they still face problems such as difficult grid construction, high computational cost and limited generalization ability [2].

In recent years, the deep integration of artificial intelligence and scientific computing has promoted the rapid development of AI for science, in which AI for PDEs has become an important research direction in the field of scientific machine learning. Physics informed neural networks (pins) embed partial differential equations, initial conditions, boundary conditions and a small amount of observation data into the loss function of the neural network to make the model not only fit the data, but also meet the physical laws in the training process [5]. Compared with traditional numerical methods, Pinns has the advantages of Meshless, easy to handle inverse problems, fusing multi-source data and suitable for high-dimensional modeling, so it is widely used in fluid dynamics, heat conduction, elasticity, wave equation and material parameter identification.

However, traditional Pinns usually use multi-layer perceptron (MLP) as the solution function approximator. Although MLP has the general approximation ability, there are still some deficiencies in the complex PDE solving task. Firstly, the nonlinearity of MLP mainly comes from the fixed activation function on the node, and its expression ability depends on the combination of a large number of linear weights and nonlinear activation, which may require a large network scale in high-frequency, multi-scale and local singular problems. Secondly, MLP Pinns are easily affected by spectral deviation, gradient propagation difficulties and loss term imbalance in the training process. Thirdly, the interpretability of the internal structure of MLP is weak, and it is difficult to directly analyze the functional relationship between the input variables and the physical field output.

Kolmogorov Arnold networks (Kan) provides a new idea for improving the function approximator of pins. Inspired by Kolmogorov Arnold representation theorem, the core difference between Kan and MLP is that MLP places the fixed activation function on the neuron node, while Kan places the learnable one-dimensional function on the edge of the network and parameterizes it with spline function or other basis functions [1]. This structure makes Kan have potential advantages in function fitting, local feature expression and model interpretability.

On this basis, the researchers further proposed Kolmogorov Arnold informed neural network, also known as physics informed Kolmogorov Arnold network, or pikan. The core idea of pikan is to replace the traditional MLP with Kan as the solution function approximator in the physical information neural network, while retaining the basic framework of PDE residual and physical constraint loss constructed by Pinns through automatic differentiation [2]. Therefore, pikan can be understood as the combination of pins

Xiuliang Yu, "Physically Informed Neural Networks for Solving Hyperbolic Partial Equations," *International Journal of Multidisciplinary Research and Publications (IJMRAP)*, Volume 8, Issue 11, pp. 179-183, 2026.

and Kan: the former provides a physical constraint mechanism, while the latter provides a new functional representation structure.

This paper does not involve specific numerical experiments, but focuses on the systematic analysis of pikan from the perspective of principle and review. This paper first introduces the basic idea of physical information neural network, then expounds the theoretical basis and structural characteristics of Kan, and then analyzes the method framework of pikan and the modeling methods under different PDE expressions, and further summarizes its research progress in PDE solution, operator learning, solid mechanics and fracture mechanics. Finally, this paper discusses the advantages, limitations and future development direction of pikan.

II. Methodology

A. Partial differential equation solving problem

The boundary value problem of general partial differential equations can be expressed as:

$$N[u](x) = f(x),\ X \in \Omega,$$
$$B[u](x) = g(x), x \in \partial\Omega,$$

Where Ω is the solution region, ∂ Ω is the boundary of the region, u (x) is the physical field variable to be solved, n is the differential operator, B is the boundary operator, f (x) and G (x) are the source term and boundary condition respectively.

In traditional numerical methods, the continuous region is usually discretized into grids or elements, and then the approximate solution is obtained by constructing algebraic equations. For example, the finite element method approximates continuous field variables by dividing elements and constructing shape functions, the finite difference method approximates derivatives by difference schemes, and the finite volume method discretizes the control volume based on conservation laws. These methods have high reliability in engineering applications, but in the problems of complex geometry, high-dimensional space and frequent parameter changes, they often need to be re meshed or solved repeatedly, and the calculation cost is high.

The difference between Pinns and traditional numerical methods is that it directly uses neural network $u_\theta(x)$ approximates the true solution of U (x), and calculates the derivative term required in the equation by automatic differential. The goal of network training is no longer just to fit the data, but to make the network output meet the control equation, boundary conditions and initial conditions at the same time.

B. Construction of loss function for pinns

Physics-Informed Neural Networks are PDE-solving methods that combine deep learning with physical laws. The core idea is to use a neural network $u_\theta(\mathbf{x})$ to approximate the unknown physical field and force the network output to satisfy governing equations, boundary conditions, and initial conditions through the loss function.

The total loss function of PINNs is usually written as:

$$\mathrm{L}(\theta) = \lambda_f \mathrm{L}_f + \lambda_b \mathrm{L}_b + \lambda_i \mathrm{L}_i + \lambda_d \mathrm{L}_d.$$

Here, $\mathcal{L}_f$ denotes the PDE residual loss, $\mathcal{L}_b$ denotes the boundary-condition loss, $\mathcal{L}_i$ denotes the initial-condition loss, $\mathcal{L}_d$ denotes the data-supervision loss, and $\lambda_f, \lambda_b, \lambda_i, \lambda_d$ are the weights of different loss terms.

The PDE residual loss can be written as:

$$\mathrm{L}_f = \frac{1}{N_f}\sum_{j=1}^{N_f}\left|\mathrm{N}[u_\theta](\mathbf{x}_f^j) - f(\mathbf{x}_f^j)\right|^2.$$

The boundary-condition loss can be written as:

$$\mathrm{L}_b = \frac{1}{N_b}\sum_{j=1}^{N_b}\left|\mathrm{B}[u_\theta](\mathbf{x}_b^j) - g(\mathbf{x}_b^j)\right|^2.$$

If the problem contains an initial condition, the initial-condition loss can be expressed as:

$$\mathrm{L}_i = \frac{1}{N_i}\sum_{j=1}^{N_i}\left|u_\theta(\mathbf{x}_i^j) - u_0(\mathbf{x}_i^j)\right|^2.$$

If observational data are available, a data-supervision term can be added:

$$\mathrm{L}_d = \frac{1}{N_d}\sum_{j=1}^{N_d}\left|u_\theta(\mathbf{x}_d^j) - u^j\right|^2.$$

By minimizing the above loss function, PINNs can learn approximate solutions that satisfy physical laws without requiring large labeled datasets. Their advantages include being mesh-free, easy to integrate with observational data, and applicable to both forward and inverse problems. However, the performance of traditional PINNs highly depends on the neural network approximator itself. When MLPs are insufficient for representing complex physical fields, the accuracy and stability of PINNs may be affected.

C. Theoretical Basis of Kolmogorov-Arnold Networks

The Kolmogorov-Arnold representation theorem states that a multivariate continuous function defined on a bounded domain can be represented as a finite combination of univariate continuous functions and additions. Its basic form is:

$$f(x_1, x_2, \cdots, x_n) = \sum_{q=1}^{2n+1}\Phi_q\left(\sum_{p=1}^{n}\phi_{q,p}(x_p)\right).$$

This representation suggests that the complexity of a multivariate function can be expressed through several univariate functions and their combinations. Traditional MLPs are mainly inspired by the universal approximation theorem, whereas KANs are inspired by the Kolmogorov-Arnold representation theorem and generalize scalar weights into learnable univariate functions [1].

A standard MLP layer is usually written as:

$$\mathbf{x}_{l+1} = \sigma(W_l \mathbf{x}_l + \mathbf{b}_l).$$

Here, W_l is the linear weight matrix, $\mathbf{b}_l$ is the bias term, and σ is a fixed activation function. The nonlinearity of MLPs is mainly provided by activation functions on nodes.

A KAN layer can instead be written as:

Xiuliang Yu, "Physically Informed Neural Networks for Solving Hyperbolic Partial Equations," *International Journal of Multidisciplinary Research and Publications (IJMRAP)*, Volume 8, Issue 11, pp. 179-183, 2026.

$$x_{l+1,i} = \sum_{j=1}^{n_l} \phi_{l,i,j}(x_{l,j}).$$

Here, $\phi_{l,i,j}$ denotes the learnable univariate function connecting the j-th node in layer l to the i-th node in layer $l+1$. In other words, an edge in an MLP is usually only a scalar weight, while an edge in a KAN is itself a function.

The original KAN usually parameterizes the edge function using B-splines:

$$\phi(x) = \sum_k c_k B_k(x).$$

Here, $B_k(x)$ is the spline basis function and c_k is a trainable coefficient. Since spline functions have local support, KAN can learn different local functional patterns in different input intervals, giving it potential in local complex function approximation and multiscale representation.

Compared with MLPs, KAN has three potential advantages. First, each edge has nonlinear expressive capability, allowing the model to learn local functional structures more flexibly. Second, edge functions can be visualized, improving interpretability. Third, KAN is related to traditional numerical approximation methods such as splines and orthogonal polynomials, making it theoretically attractive for scientific computing. However, KAN also faces challenges such as high computational cost, hyperparameter sensitivity, and insufficient training stability [3].

D. Framework of Physics-Informed Kolmogorov-Arnold Networks

The core idea of PIKAN is to replace the traditional MLP in PINNs with KAN. Conventional PINNs use MLP as the solution approximator:

$$u_\theta(\mathbf{x}) = MLP(\mathbf{x};\theta).$$

PIKAN uses KAN to represent the unknown solution:

$$u_\theta(\mathbf{x}) = KAN(\mathbf{x};\theta).$$

Therefore, PIKAN does not change the physical constraint mechanism of PINNs; instead, it changes the function approximator. It still computes PDE residuals through automatic differentiation and constrains the network to satisfy governing equations, boundary conditions, initial conditions, and observational data through the total loss function [2]. Methodologically, PIKAN can be understood as the combination of the physical constraint mechanism of PINNs and the function representation structure of KAN.

For strong-form PIKAN, the governing-equation residual can be defined as:

$$r_f(\mathbf{x}) = \mathrm{N}[u_\theta^{KAN}](\mathbf{x}) - f(\mathbf{x}).$$

The corresponding PDE residual loss is:

$$\mathrm{L}_f = \frac{1}{N_f}\sum_{j=1}^{N_f} |r_f(\mathbf{x}_f^j)|^2 .$$

Strong-form PIKAN is direct and relatively easy to implement. It is suitable for problems such as the Poisson equation, Burgers equation, Helmholtz equation, and Navier-Stokes equation. However, when the PDE contains high-order derivatives or the solution has singularities, automatic differentiation and residual optimization may become difficult.

Besides the strong form, PIKAN can also be extended to the energy form. For many solid mechanics problems, the true solution can be regarded as the minimizer of an energy functional. Suppose the total potential energy is:

$$\Pi[u] = U[u] - W[u],$$

where $U[u]$ is the internal strain energy and $W[u]$ is the external work potential. Energy-form PIKAN can be written as:

$$u_\theta^{KAN} = \arg\min_\theta \Pi[u_\theta^{KAN}].$$

λ can be optimized together with the network parameter θ:

$$\mathrm{L}(\theta,\lambda) = \mathrm{L}_f(\theta,\lambda) + \mathrm{L}_b(\theta) + \mathrm{L}_d(\theta).$$

This form is suitable for material parameter identification, source inversion, boundary-condition estimation, and physical field reconstruction.

III. Research Progress And Discussion

A. PIKAN for Forward and Inverse PDE Problems

The most direct application of PIKAN is solving forward and inverse PDE problems. In forward problems, governing equations, boundary conditions, and initial conditions are usually known, and the model aims to learn a solution function that satisfies these constraints. In inverse problems, some physical parameters, source terms, or boundary conditions are unknown, and the model needs to identify them while fitting observational data.

Wang et al. proposed the Kolmogorov-Arnold-Informed Neural Network and introduced KAN into different PDE-solving frameworks, including strong-form, energy-form, and inverse-form formulations [2]. Existing studies show that PIKAN has potential in multiscale problems, singularity problems, stress concentration, nonlinear hyperelasticity, and heterogeneous materials. These problems usually contain local complex structures or strong nonlinear features, where conventional MLP-PINNs may require large network scales, while the edge-function structure of KAN provides more flexible local representation.

However, PIKAN does not unconditionally outperform MLP-PINNs in forward and inverse problems. Its performance is affected by the type of KAN basis functions, grid size, spline order, network depth, sampling strategy, and optimization method. Therefore, in practical applications, the model structure should be selected according to the specific PDE type and solution characteristics.

B. PIKAN in Operator Learning

In addition to solving a single PDE problem, KAN has also been introduced into neural operator learning. Traditional PINNs mainly focus on one specific PDE problem, whereas neural operator methods, such as DeepONet and Fourier Neural Operator, aim to learn mappings from input functions to output functions, enabling fast prediction for a family of PDE problems [6][7].

Models such as DeepOKAN attempt to replace the MLP modules in DeepONet with KAN, giving operator learning

Xiuliang Yu, "Physically Informed Neural Networks for Solving Hyperbolic Partial Equations," *International Journal of Multidisciplinary Research and Publications (IJMRAP)*, Volume 8, Issue 11, pp. 179-183, 2026.

models stronger univariate function representation capability. Shukla et al. conducted a fair comparison between MLP and KAN representations in differential equations and operator networks, involving PIKAN, PINNs, DeepOKAN, and DeepONet [3]. The study indicates that the original B-spline KAN does not always have advantages in accuracy and efficiency. Modified low-order orthogonal polynomial versions can achieve performance comparable to PINNs and DeepONet, but they may still suffer from random-seed sensitivity and instability with high-order polynomials.

This indicates that the combination of KAN and operator learning is valuable, but stricter benchmark tests are still needed. In the future, the physical constraint mechanism of PIKAN can be combined with the generalization capability of KAN-based neural operators to build intelligent solvers that satisfy physical laws and rapidly adapt to different parameters, boundary conditions, and source terms.

C. PIKAN in Solid Mechanics and Fracture Mechanics

PIKAN also has strong application potential in solid mechanics, material damage, and fracture mechanics. In solid mechanics, displacement fields, stress fields, and strain fields often contain local high gradients, singularities, or discontinuities, such as stress concentration at crack tips, abrupt changes at heterogeneous material interfaces, and nonlinear responses under complex loads. These characteristics make the training of conventional MLP-PINNs more difficult.

Energy-form PIKAN has a natural connection with variational principles in solid mechanics. By minimizing energy functionals, the model can avoid directly handling some high-order differential operators, which is advantageous in elasticity and structural mechanics. Meanwhile, the local function representation capability of KAN helps capture complex field variations caused by stress concentration and material heterogeneity.

In fracture mechanics, PD-KINN combines KAN with peridynamic-informed neural networks to predict elastic deformation and brittle damage [4]. Peridynamics uses nonlocal integral operators to describe interactions among material points and is suitable for handling cracks, damage, and discontinuous variables. The emergence of PD-KINN shows that KAN can be used not only for general PDE solving, but also for integration with specific physical theories to form specialized physics-informed learning frameworks for complex engineering problems.

D. Advantages and Limitations of PIKAN

The main advantage of PIKAN lies in its function representation capability. Since each edge in KAN is a learnable univariate function, the model can learn different local functional patterns in different input intervals. For multiscale, high-frequency, locally abrupt, or singular solution functions, PIKAN theoretically has stronger local representation capability than conventional MLP-PINNs.

Second, PIKAN has better interpretability. The edge functions of KAN can be visualized, allowing researchers to observe the functional patterns learned on different edges. This is particularly important for scientific machine learning because scientific computing focuses not only on prediction error, but also on whether the model has learned reasonable physical structures.

Third, PIKAN is related to traditional numerical analysis. Splines, orthogonal polynomials, radial basis functions, and wavelets are important approximation tools in numerical computation. KAN introduces these univariate approximation ideas into the edge structure of deep networks, making it theoretically attractive for PDE-solving tasks.

However, PIKAN also has clear limitations. First, the original B-spline KAN has relatively high computational cost, and current deep learning frameworks are less optimized for it than for MLP matrix multiplication. Second, PIKAN is sensitive to grid size, spline order, network depth, initialization, and loss weights. Third, its performance in complex geometries is not always stable and may require coordinate mapping, domain decomposition, or adaptive sampling. Finally, the theoretical analysis of PIKAN is still insufficient, and its error estimates, convergence, and generalization capability require further investigation [3].

IV. Conclusions And Future Perspectives

This paper reviews the theoretical foundation, methodological framework, research progress, advantages, limitations, and future directions of Physics-Informed Kolmogorov-Arnold Networks. The core idea of PIKAN is to replace the traditional MLP solution approximator with KAN while preserving the physical residual constraint mechanism of PINNs. Since KAN places learnable univariate functions on network edges, PIKAN has potential advantages in multiscale modeling, local complex function approximation, and model interpretability.

Existing studies show that PIKAN has promising applications in multiscale problems, singularity problems, stress concentration, nonlinear hyperelasticity, and heterogeneous materials. However, it still has limitations in computational efficiency, training stability, adaptability to complex geometries, and theoretical analysis. Future research may proceed in several directions. First, the basis functions of KAN can be improved by exploring Chebyshev polynomials, Legendre polynomials, RBFs, wavelets, and NURBS. Second, adaptive PIKAN can be developed to dynamically adjust sampling points and grid density according to PDE residuals. Third, PIKAN can be combined with domain decomposition methods to improve adaptability to complex geometries and large-scale PDE problems. Fourth, neural operator methods such as DeepONet, FNO, and PINO can be integrated so that the model can learn solution operators for a family of PDEs rather than solving only a single PDE. Fifth, unified benchmarks should be established to fairly compare PIKAN with MLP-PINNs under the same parameter scale, training strategy, and evaluation metrics.

Overall, PIKAN provides a theoretically inspired and practically promising method for intelligent solving of complex partial differential equations. With further developments in network architecture, basis function design, optimization strategies, and physical constraint modeling,

Xiuliang Yu, “Physically Informed Neural Networks for Solving Hyperbolic Partial Equations,” *International Journal of Multidisciplinary Research and Publications (IJMRAP)*, Volume 8, Issue 11, pp. 179-183, 2026.

PIKAN is expected to become an important research direction in scientific machine learning and engineering simulation.

ACKNOWLEDGMENT

This research is supported by the Undergraduate Innovation and Entrepreneurship Training Program of Anhui University of Finance and Economics, Project No. 202510378014.

REFERENCES

[1] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark, "KAN: Kolmogorov-Arnold Networks," *International Conference on Learning Representations*, 2025.

[2] Y. Wang, J. Sun, J. Bai, C. Anitescu, M. S. Eshaghi, X. Zhuang, T. Rabczuk, and Y. Liu, "Kolmogorov-Arnold-Informed neural network: A physics-informed deep learning framework for solving forward and inverse problems based on Kolmogorov-Arnold Networks," Computer Methods in Applied Mechanics and Engineering, vol. 433, 117518, 2025.

[3] K. Shukla, J. D. Toscano, Z. Wang, Z. Zou, and G. E. Karniadakis, "A comprehensive and FAIR comparison between MLP and KAN representations for differential equations and operator networks," arXiv:2406.02917, 2024.

[4] Y. Nie, Y. Zhang, Y. Zhu, and X. Guo, "PD-KINN: Kolmogorov-Arnold representation theorem enhanced peridynamic-informed neural network for predicting elastic deformation and brittle damage," Engineering Analysis with Boundary Elements, vol. 176, 106214, 2025.

[5] 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.

[6] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, "Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators," Nature Machine Intelligence, vol. 3, pp. 218–229, 2021.

[7] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, "Fourier Neural Operator for Parametric Partial Differential Equations," International Conference on Learning Representations, 2021.