Kolmogorov-Arnold Network for Solving 2-D Magnetostatic Problems

Yachao Zhu[1], *Student Member*, Kai Xu[1], Bingkuan Wan[1], Gang Lei[1], *Senior Member, IEEE* and Jianguo Zhu[2], *Senior Member, IEEE*

[1]School of Electrical and Data Engineering, University of Technology Sydney, Ultimo, NSW 2007, Australia
[2]School of Electrical and Computer Engineering, The University of Sydney, Camperdown, NSW 2006, Australia

Abstract: **The data-driven machine-learning approach has significantly advanced the development of computational electromagnetics. This study introduces the Kolmogorov-Arnold Network (KAN) as a novel method to overcome the limitations of traditional multilayer perceptron-based physics-informed neural networks (MLP-PINNs), which often struggle with fixed activation functions and high computational costs. In terms of accuracy, KAN outperforms traditional PINNs with a 13.47% improvement in estimated accuracy. For efficient convergence, KAN achieves stable training with only 175 steps, significantly reducing computational overhead compared to PINNs, which require over 15,000 steps. Additionally, KAN demonstrates superior generalizability and flexibility, achieving an average 5.03% accuracy improvement over PINN in transfer learning scenarios. These results highlight KAN's potential in computational electromagnetics.**

***Index Terms*—computational electromagnetics, magnetostatic fields, partial differential equations (PDEs), Kolmogorov-Arnold Network (KAN)**

I. Introduction

In modern computational electromagnetics, data-driven approaches are becoming increasingly crucial. Traditional physical models and numerical methods, such as the Finite Difference Method (FDM) and Finite Element Method (FEM), have been widely applied across multiple fields. However, those traditional methods often entail high computational costs and substantial extra computational time when dealing with nonlinear, non-stationary complex geometry or high-dimensional problems [1]. To overcome those limitations, residual networks like neural operators, and physics-informed neural networks (PINNs) have been introduced [2, 3]. Both integrate the ability to solve partial differential equations (PDEs), ordinary differential equations (ODEs), boundary conditions and initial conditions, directly into deep learning models. Those approaches not only enhance solution accuracy but also reduce data requirements. In particular the PINN, it has been widely used to provide faster approximate solutions to complex physical phenomena such as multi-scale, chaotic or turbulent behaviour [4-7].

Despite these advantages, PINNs still have certain limitations. Most current PINN research relies on multilayer perceptrons (MLPs) using fully connected feed-forward networks [8]. These highly nonlinear neural networks have complex multilayer structures, making it difficult to understand how they make decisions or extract features and weights. The 'black box' features reduce the transparency of the PINN model's internal mechanisms, hindering its interpretability [9]. PINN activation functions are predetermined before training, requiring multiple trials or experience-based selection to identify suitable configurations. Additionally, PINNs are highly sensitive to hyperparameters (learning rate, layers number, regularisation, etc.), which significantly affect performance. Hyperparameter tuning demands expertise to find optimal combinations and is often time-consuming and labour-intensive, with poor transferability across different problems [10, 11].

Considering the limitations of multilayer perception-based physics-informed neural network (MLP-PINN), the Kolmogorov-Amold theorem offers an alternative approach for physical law approximation, enabling the transformation of complex nonlinear systems into linear systems through appropriate coordinate transformations [12, 13]. Unlike traditional MLPs, the Kolmogorov-Arnold Network (KAN) does not require fixed activation functions. The improved interpretability of KAN over PINNs stems from its unique structure, where traditional weight parameters are replaced with one-variable functions. This design allows for a clearer understanding of how each function contributes to the model's output, providing more transparent insights into the decision-making process. In contrast, PINNs rely on fixed activation functions and dense connections, making it difficult to trace how specific features influence the results [12, 13].

This study applies KAN to solve 2-D magnetostatic field problems, exploring its practical applications in computational electromagnetics. A KAN-based model is constructed, and the optimal hyperparameters are evaluated to solve the vector potential of magnetostatic fields within PDEs. The model's performance and adaptability are assessed in different geometric settings, including square and circular domains. Additionally, a transfer learning approach is employed to validate the model's generalization ability and flexibility.

II. Electromagnetic Theory

Magnetostatic fields are generated by constant currents or permanent magnets and are characterized as curl fields. A 2-D analysis is applicable when electromagnetic field quantities, such as electric and magnetic field strength, remain nearly constant in a specific direction. In 2-D magnetostatic field problems, the current density J_S and vector potential $\vec{A}$ are parallel to each other and parallel to the z-axis. At the same time, the magnetic induction intensity $\vec{B}$ and magnetic field strength $\vec{H}$ have zero components along the z-axis, with nonzero

components in the x and $y-$ axis. According to Ampere's Law and Maxwell's equation:

$$\nabla \times H = J_i \tag{1}$$

$$\nabla \times A = \mu_0 \mu_r H = B \tag{2}$$

where μ_0 is permeability of vacuum, μ_r is relative permeability.

$$\nabla \times H = \nabla \times \left(\frac{1}{\mu_0 \mu_r} B\right) = \nabla \times \left(\frac{1}{\mu_0 \mu_r} \nabla \times A\right) = J_i \tag{3}$$

By applying vector identities and Coulomb norm, the function (3) can be transformed into:

$$\nabla^2 A = -\mu_o \mu_r J_i \tag{4}$$

Since vector potential $\vec{A}$ only has the z-axis component:

$$\frac{\partial^2 A}{\partial x^2} + \frac{\partial^2 A}{\partial y^2} = -\mu_o \mu_r J_i \tag{5}$$

By introducing the conductivity v for linear magnetic materials, defined as the inverse of the product of vacuum permeability and relative permeability, the form of function (6) can be transformed. Considering the Dirichlet boundary conditions on Γ, the 2-D magnetostatic fields satisfy the following boundary value problem in residual form:

$$\begin{cases} Res1 = -\dfrac{\partial}{\partial x}\left(v\dfrac{\partial A}{\partial x}\right) - \dfrac{\partial}{\partial y}\left(v\dfrac{\partial A}{\partial y}\right) - J_i \\ Res2 = A|_\Gamma \end{cases} \tag{6}$$

III. Kolmogorov-Arnold Network (KAN) Architecture

A. KAN Structure

The KAN is based on the Kolmogorov-Arnold representation theorem, serving as a complexity reduction technique. It enables the conversion of a multivariate continuous function into a composition of one-variable continuous functions with two-argument summaries [14]. The following formula shows that an n-variate continuous function $f(x)$ can be represented using $(2n+1)$ outer one-variable continuous function Φ_q and $(2n+1) \times n$ inner one-variable continuous function $\Psi_{q,p}$:

$$f(x) = f(x_1, x_2 \ldots . x_n) = \Sigma_{q=1}^{2n+1} \Phi_q \left(\Sigma_{p=1}^{n} \Psi_{q,p}(x_p)\right) \tag{7}$$

To implement the KAN architecture accordingly, the first step is to construct residual activation functions for each node. These include a basic activation function $g(x)$, and a B-spline function $S(x)$, both with learnable spline coefficients:

$$\phi(x) = \omega_b g(x) + \omega_s S(x) \tag{8}$$

where ω_b and ω_s are the learnable weights. The second step involves scaling initialization, where ω_s is initially set to 1, and ω_b is initialized using the Xavier method. Subsequently, KAN updates spline grids based on input activations and combines the final outputs from each node:

$$KAN(\mathbf{x}) = (\mathbf{\Phi}_{L-1} \circ \mathbf{\Phi}_{L-2} \circ \ldots \circ \mathbf{\Phi}_1 \circ \mathbf{\Phi}_0)\mathbf{x} \tag{9}$$

where $\mathbf{x} = (\boldsymbol{x}, \boldsymbol{y})$, and Φ_L represents the function matrix of the

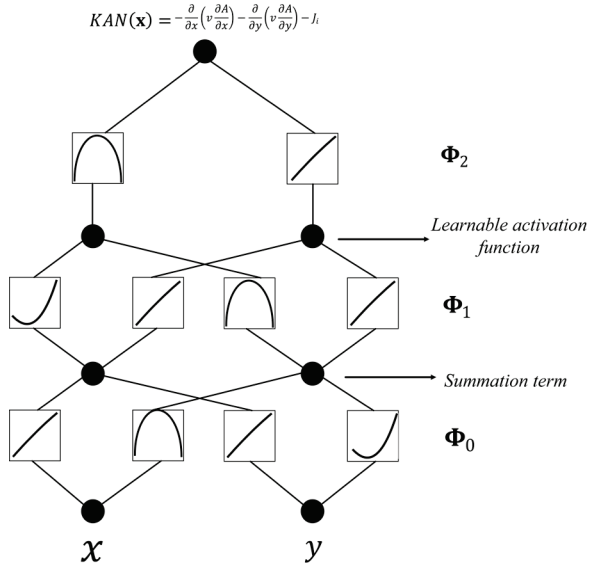

Fig. 1. The KAN architecture for 2-D Magnetostatic Fields Solving.

Lth layer. The overall KAN architecture is illustrated in Fig. 1. The training process minimizes the loss based on the residuals from PDEs and boundary conditions. The total losses are defined as the sum of the l_{PDE} and $l_{boundary}$, as shown below:

$$L_{total} = \alpha \cdot l_{PDE} + \beta \cdot l_{Boundary} \tag{10}$$
$$= \alpha \cdot \left(\frac{1}{N}\sum_{i=1}^{N} (Res(x_i, y_i))^2\right) + \beta \cdot \left(\frac{1}{M}\sum_{i=1}^{M} (Res(x_j, y_j))^2\right)$$

where α and β are weighting parameters that adjust the relative importance of each loss term. N and M represent the number of interior and boundary sample points, respectively. $Res(x_i, y_i)$ and $Res(x_j, y_j)$ denote the residuals at interior and boundary points. KAN employs the Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm to approximate the Hessian matrix, optimize the total loss, and determine optimal step sizes for model convergence through iterative training.

B. KAN hyperparameters

Although KAN hyperparameters are less sensitive than those of MLPs, initial network width adjustments remain necessary for tuning. Unlike MLP-PINN, KAN starts with simpler settings and fewer parameters, including Width (layer structure), Grid (intervals in each spline segment), Spline Order 'k' (the polynomial degree of B-spline basis functions), and Grid Epsilon 'grid_eps' (the control for grid extension). This study systematically evaluates how these hyperparameters affect KAN performance using the L2 error metric and identifies optimal settings for 2-D magnetostatic fields. Test cases are based on the PDE described in Section 2 without boundary conditions. Widths range from [2, 2, 1] to [2, 2, 2, 2, 1], and grid sizes range from 5 to 100.

Fig. 2 shows that larger grid sizes increase training time, especially above 20, while smaller widths such as [2, 2, 2, 1] at a grid size of 50 yields lower and more stable L2 errors and shorter training times.

After confirming these hyperparameters, the model is trained for up to 1000 steps to evaluate L2 errors. Fig. 3 indicates that the L2 error decreases and stabilises after 175 steps. Table I

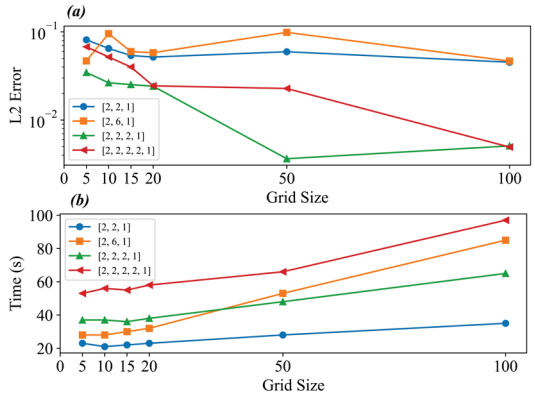

Fig. 2. Result of hyperparameter test. (a) L2 error with grid size. (b) variation training time with grid size.

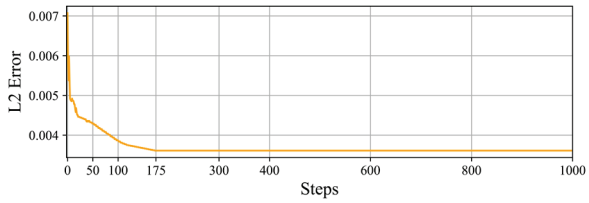

Fig. 3. Result of L2 Error in the KAN Model with [2,2,2,1] Width and 50 grid size over increasing training steps.

TABLE I
HYPERPARAMETERS FOR 2-D MAGNETOSTATIC FIELDS SOLVING KAN MODELS

Hyperparameter	Value	Hyperparameter	Value
Width	[2,2,2,1]	Epsilon	1
Grid size	50	Training steps	175
Spline order	3	Optimiser	Adam
Sampling point for square	42 (Domain) 42 (Boundary)	Sampling point for circular	800 (Domain) 200 (Boundary)

details the optimal hyperparameters for solving 2-D magnetostatic fields with KAN.

IV. NUMERICAL EXAMPLES

Two geometer settings have been designed for solving 2-D magnetostatic problems using the KAN approach trained on the CPU. The PINN method is also implemented under the same problem setup trained on CPU and CUDA-based GPU for the fitting accuracy comparative analysis. Numerical example tests are conducted on the equipped with an AMD Ryzen 5800X3D Desktop CPU and NVIDIA RTX 4070 Laptop GPU. The PINN approach is developed using the DeepXDE library, tailored specifically for scientific machine learning and physics-informed learning with TensorFlow and PyTorch backends [15]. PINN hyperparameters are optimized through iterative searches across multiple combinations within a predefined range.

The PINN architecture consists of a feedforward neural network with 3 hidden layers, each containing 50 neurons and Dirichlet boundary conditions. The activation function is the hyperbolic tangent (tanh), and the number of sampling points matches those used by KAN, as shown in Table I. The Adam optimizer is selected for training. Additionally, the FDM is employed to compare with KAN's approximate solutions and to analyse the error distribution.

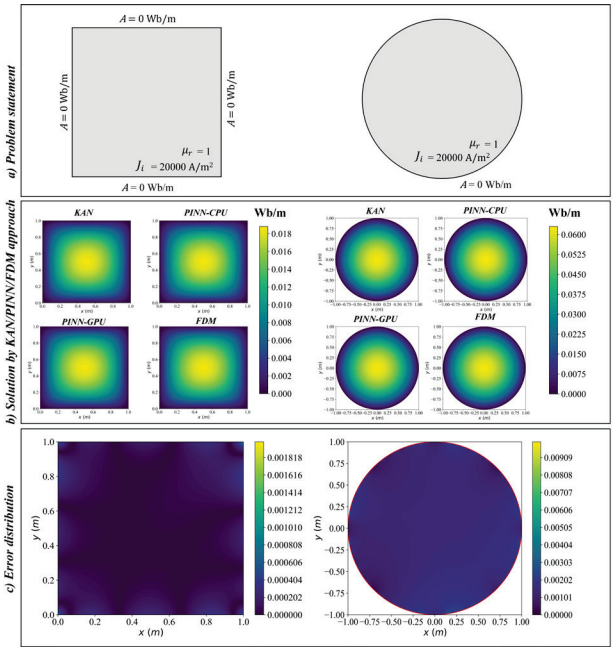

Fig. 4. The comparative study result overview for problems. (a) Problem statement. (b) Solution of vector potentials by KAN/PINN/FDM approach in square geometric (left) and circular geometric (right). (d) KAN approach error (absolute) distribution compares with FDM.

After the initial training of KAN for Case A, transfer learning is applied in Case B, using the pre-trained model from Case A to adapt to the changes in current density and the material's relative permeability.

A. 2-D Magnetostatic in Square and Circular geometric

Case A involves an electromagnetic shielding device within a square area with a side length of 1 meter and an electric cable within a circular area with a 1-meter radius, as shown in Fig. 4(a). The conductors are made of linear, non-magnetic material such as copper, with relative permeability μ_r set to 1 for simplification. The current density through the copper conductor is set at $2 \times 10^5 A / m^2$, approximately one-fourth to one-sixth of the current density in copper composite cables, falling within the safe current-carrying capacity [16, 17].

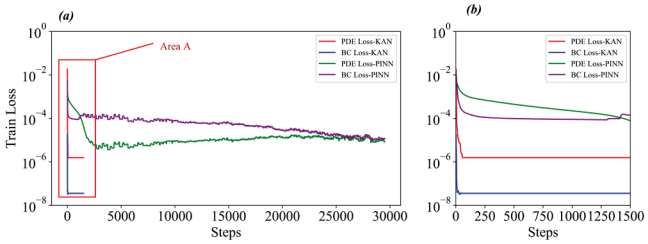

Fig. 5. Training loss for KAN and MLP-PINN (GPU) for square geometric. (a) Full steps history. (b) 1500 steps history.

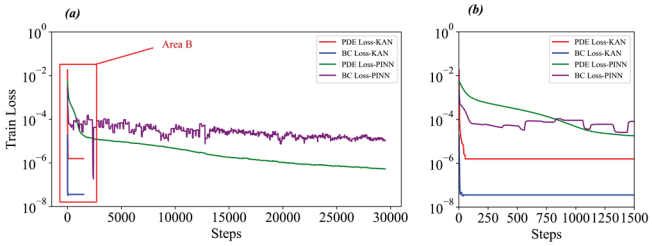

Fig. 6. Training loss for KAN and MLP-PINN (GPU) for circular geometric. (a) Full steps history. (b) 1500 steps history.

B. Transfer learning for 2-D Magnetostatic in Square and Circular Field

Case B extends Case A in two directions for transfer learning model training. First, the conductor material inside the square electromagnetic shielding device is changed to nickel, increasing the relative permeability to 100. Second, the current density in the conductor within the circular electric cable is doubled to $4 \times 10^5 A / m^2$. To demonstrate the generalizability and flexibility of the KAN, a transfer learning technique is applied, using the Case A-trained KAN model as a checkpoint. This enables efficient training with fewer iterations to solve magnetostatic field problems under varying conditions. The optimal number of training steps for the transfer learning phase is determined as 10 from different step sizes experiments conducted from 5 to 20 steps. For comparison, a fine-tuning approach based on PINN has also been implemented. Through hyperparameter search, the optimal retraining configuration for PINN is determined to 3200 steps.

V. Result and Discussion

A. High interpretability of KAN

The results of the two cases described above are depicted in Fig. 4(b) and Fig. 8. In Case A, the KAN model demonstrates the capabilities of solving 2-D magnetostatic fields within both square and circular geometries. KAN decomposes multivariate functions into several one-dimensional continuous functions, with each node representing an explicit mathematical function. The activation functions learned by each node are clearly illustrated in Table II. High R^2 values (close to 1) for each one-variable function indicate strong fitting accuracy with minimal error. The visualization of the KAN model's outputs is presented in Fig. 7. In contrast, PINNs encapsulate learned information within deeply nested weight matrices and activation functions, making it difficult to trace how input features influence outputs. The node structure of KAN enhances accuracy and provides clear insights into the physical meaning of the learned features, reflecting its high interpretability.

B. High accuracy of KAN

The KAN approach visualises vector potentials that match those obtained by the FDM, as shown in Fig. 4(b). To evaluate the accuracy of the KAN model, mean absolute error (MAE), root mean squared error (RMSE), and estimated accuracy (in percentage) are compared between the KAN and PINN (CPU and GPU) models, using FDM results as the reference, as detailed in Table III. The KAN model demonstrates lower MAE and RMSE compared to the PINN model in both square and circular geometries. This is notable when dealing with circular field problems. The estimated accuracy confirms that the KAN efficiently handles geometric constraints while maintaining superior precision, achieving higher accuracy than the PINN model in both square and circular conditions. Especially for square conditions, the KAN model shows a 13.47 % improvement in accuracy over the PINN (GPU) model, highlighting KAN's high accuracy and robustness.

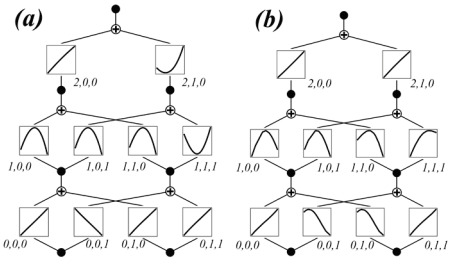

Fig. 7. The KAN learned activation function for each node. (a) Square KAN model. (b) Circular KAN model.

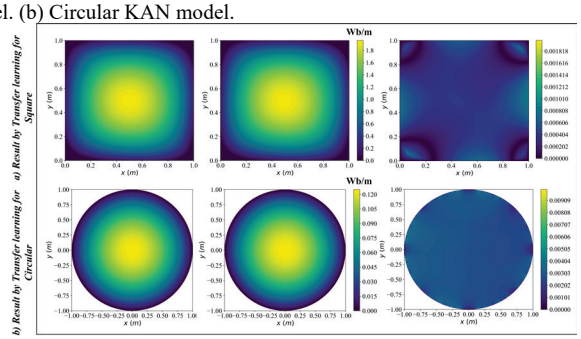

Fig. 8. The comparative study result for Transfer learning; KAN result (Left), PINN(GPU) result (Middle), Error distribution between KAN and FDM (Right) (a) Result for square geometric. (b) Result for circular geometric.

C. Efficient convergence of KAN

The convergence performance of the KAN and MLP-PINN (GPU) models is illustrated in Fig. 5 and Fig. 6 through the loss plot. Both KAN and PINN demonstrate good convergence, with the PDE loss smoothed using a moving average technique to reduce fluctuations. Since KAN requires 175 training steps to converge, the results are extended to 1500 steps for clearer comparison. In contrast, the PINN model requires over 15,000 training steps to achieve a stable loss level. The zoom-in sections (Areas A and B in Fig. 5 and 6) show that while the PDE and boundary condition losses of PINN gradually decrease. However, the decrease rate of PINN is slower than that of the KAN, with large fluctuation in losses. This comparison demonstrates that KAN maintains lower loss fluctuations and higher stability, resulting in better convergence. The results highlight that KAN's unique structure reduces model complexity and improves gradient efficiency during training.

TABLE II
FITTING FUNCTIONS AND R^2 OF NODES IN 2-D MAGNETOSTATIC PROBLEMS USING KAN METHOD

KAN Fitting Function (Square)			KAN Fitting Function (Circular)		
Node	Function	Fitting R^2	Node	Function	Fitting R^2
2, 0, 0	exp	0.999	2, 0, 0	exp	1.00
2, 1, 0	x^2	0.998	2, 1, 0	exp	0.999
1, 0, 0	x^2	0.991	1, 0, 0	x^2	0.996
1, 0, 1	x^2	0.999	1, 0, 1	sin	0.998
1, 1, 0	x^2	0.994	1, 1, 0	x^2	0.988
1, 1, 1	x^2	0.997	1, 1, 1	x^2	0.999
0, 0, 0	x	0.999	0, 0, 0	x	0.999
0, 0, 1	x	0.999	0, 0, 1	cos	0.985
0, 1, 0	x	0.999	0, 1, 0	cos	0.985
0, 1, 1	x	0.999	0, 1, 1	x	0.998

TABLE III
COMPARISON OF ERROR METRICS AND ESTIMATED ACCURACY FOR KAN AND PINN MODELS

Numerical Examples	Training Time (s)	MAE	RMSE	Estimated Accuracy
KAN Square	297	0.66×10^{-4}	0.95×10^{-4}	99.64%
PINN Square (CPU)	1131	1.85×10^{-4}	6.5×10^{-4}	90.12 %
PINN Square (GPU)	102	2.26×10^{-4}	8×10^{-4}	87.81 %
KAN Circular	223	0.55×10^{-4}	2×10^{-4}	99.14%
PINN Circular (CPU)	982	9.2×10^{-4}	3.1×10^{-4}	98.75%
PINN Circular (GPU)	99	1.03×10^{-4}	3.74×10^{-4}	98.38%
KAN Transfer Square	39	81.1×10^{-3}	88.1×10^{-3}	95.62%
PINN Transfer Square (GPU)	49	2.23×10^{-3}	7.16×10^{-3}	87.96%
KAN Transfer Circular	57	2.13×10^{-3}	2.45×10^{-3}	98.32%
PINN Transfer Circular (GPU)	43	7.56×10^{-3}	8.7×10^{-3}	95.92%

D. Generalizability and flexibility of KAN

Case B indicates the generalizability of the KAN model trained in Case A, allowing it to adapt to parameter changes, including the relative permeability (from 1 to 100) and current density (from $2\times 10^5 A/m^2$ to $4\times 10^5 A/m^2$). The governing equation and problem setup remain the same as in Case A. Transfer learning is applied using the pre-trained KAN model from Case A to initialize the new model. The solution plot demonstrates that transfer training refits functions and weight parameters, which are visualised in Fig. 8. Despite requiring only 10 training steps, the model achieves good convergence and maintains a low absolute error compared to the FDM results, as shown in Fig. 8. The transfer learning significantly reduces the training time compared to full retraining. Specifically, for the square geometry, transfer learning requires only 13.13% of the original training time, while for the circular geometry, it requires 25.56%. In the comparative experiment with PINN-based transfer learning, KAN consistently maintains a lower mean MAE and root mean square error RMSE and an average 5.03% improvement in accuracy over PINN, as stated in Table III. These results demonstrate that KAN's transfer learning approach is highly adaptable, precise, robust generalization, and reduces the need for extensive retraining. This proves that transfer learning for KAN is rapidly adaptable and time efficient and reduces retraining requirements.

VI. CONCLUSION

This study explored the practical application of KAN in 2-D computational magnetostatic problems. By comparing it with traditional FDM and MLP-PINN methods, the results show that KAN achieves accurate solutions for both square and circular geometries while maintaining excellent interpretability. KAN requires significantly fewer training steps to reach convergence with high accuracy and efficiency and relies on far fewer hyperparameters than MLP-PINN. Additionally, the application of transfer learning significantly accelerates model fitting while preserving accuracy when parameters change. These findings demonstrate that the KAN model possesses strong generalization ability and flexibility across different settings.

REFERENCES

[1] T. Würth, N. Freymuth, C. Zimmerling, G. Neumann, and L. Kärger, "Physics-informed MeshGraphNets (PI-MGNs): Neural finite element solvers for non-stationary and nonlinear simulations on arbitrary meshes," *Computer Methods in Applied Mechanics and Engineering,* vol. 429, p. 117102, 2024.

[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, 2019/02/01/ 2019, doi: https://doi.org/10.1016/j.jcp.2018.10.045.

[3] K. Azizzadenesheli, N. Kovachki, Z. Li, M. Liu-Schiaffini, J. Kossaifi, and A. Anandkumar, "Neural operators for accelerating scientific simulations and design," *Nature Reviews Physics,* pp. 1-9, 2024.

[4] Z. Gong, Y. Chu, and S. Yang, "Physics-Informed Neural Networks for Solving 2-D Magnetostatic Fields," *IEEE Transactions on Magnetics,* vol. 59, no. 11, pp. 1-5, 2023.

[5] M. Baldan, P. D. Barba, and D. A. Lowther, "Physics-Informed Neural Networks for Inverse Electromagnetic Problems," *IEEE Transactions on Magnetics,* vol. 59, no. 5, pp. 1-5, 2023, doi: 10.1109/TMAG.2023.3247023.

[6] A. Khan and D. A. Lowther, "Physics Informed Neural Networks for Electromagnetic Analysis," *IEEE Transactions on Magnetics,* vol. 58, no. 9, pp. 1-4, 2022, doi: 10.1109/TMAG.2022.3161814.

[7] P. Brendel, V. Medvedev, and A. Rosskopf, "Convolutional Physics-Informed Neural Networks for Fast Prediction of Core Losses in Axisymmetric Transformers," *IEEE Transactions on Magnetics,* pp. 1-1, 2024, doi: 10.1109/TMAG.2024.3431703.

[8] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, "Scientific machine learning through physics–informed neural networks: Where we are and what's next," *Journal of Scientific Computing,* vol. 92, no. 3, p. 88, 2022.

[9] K. Antonion, X. Wang, M. Raissi, and L. Joshie, "Machine Learning Through Physics–Informed Neural Networks: Progress and Challenges," *Academic Journal of Science and Technology,* vol. 9, no. 1, pp. 46-49, 2024.

[10] [Z. Zhao, L. Alzubaidi, J. Zhang, Y. Duan, and Y. Gu, "A comparison review of transfer learning and self-supervised learning: Definitions, applications, advantages and limitations," *Expert Systems with Applications,* p. 122807, 2023.

[11] A. Beltrán-Pulido, I. Bilionis, and D. Aliprantis, "Physics-Informed Neural Networks for Solving Parametric Magnetostatic Problems," *IEEE Transactions on Energy Conversion,* vol. 37, no. 4, pp. 2678-2689, 2022, doi: 10.1109/TEC.2022.3180295.

[12] Z. Liu *et al.*, "Kan: Kolmogorov-arnold networks," *arXiv preprint arXiv:2404.19756,* 2024.

[13] Z. Liu, P. Ma, Y. Wang, W. Matusik, and M. Tegmark, "Kan 2.0: Kolmogorov-arnold networks meet science," *arXiv preprint arXiv:2408.10205,* 2024.

[14] J. Schmidt-Hieber, "The Kolmogorov–Arnold representation theorem revisited," *Neural Networks,* vol. 137, pp. 119-126, 2021/05/01/ 2021, doi: https://doi.org/10.1016/j.neunet.2021.01.020.

[15] 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.

[16] S. Melsom and H. Booth, "The current-carrying capacity of solid bare copper and aluminium conductors," *Journal of the Institution of Electrical Engineers,* vol. 62, no. 335, pp. 909-915, 1924.

[17] M. B. Bazbouz, A. Aziz, D. Copic, M. De Volder, and M. E. Welland, "Fabrication of high specific electrical conductivity and high ampacity carbon Nanotube/Copper composite wires," *Advanced Electronic Materials,* vol. 7, no. 4, p. 2001213, 2021.