Article

Kolmogorov–Arnold Finance-Informed Neural Network in Option Pricing

Charles Z. Liu [1,2], Ying Zhang [1,*], Lu Qin [1] and Yongfei Liu [2]

[1] Australian Artificial Intelligence Institute, University of Technology Sydney, Sydney, NSW 2007, Australia; charles@eulerai.au (C.Z.L.); lu.qin@uts.edu.au (L.Q.)
[2] EulerAI, Sydney, NSW 2036, Australia; fayer@eulerai.au
* Correspondence: ying.zhang@uts.edu.au

Abstract: Finance-Informed Neural Networks (FINNs), inspired by Physical Information Neural Networks (PINNs) and computational finance, aim to enhance risk assessment and support regulatory decision-making. Despite their promising potential, existing FINNs face significant challenges related to learning efficiency and overall performance. This paper examines the KAFIN framework, based on the Kolmogorov–Arnold representation, which aims to improve financial modeling and analytical calculations. To evaluate the performance of FINNs, this study uses generated data instead of real market data. This choice enables controlled simulations of various financial scenarios while avoiding the complexities and noise inherent in actual market data. By relying on generated data, we are able to isolate and assess the core capabilities of KAFIN under well-defined theoretical conditions, facilitating a clearer analysis of its performance. This study focuses on European option pricing as a case study, using generated input data that simulate a range of market scenarios, including typical regulatory conditions in financial markets. Testing KAFIN under these controlled conditions allows for a rigorous evaluation of its ability to handle the complexities of pricing options across different market assumptions and regulatory constraints. Empirical results demonstrate that KAFIN significantly outperforms the baseline method, improving pricing accuracy by minimizing residual errors and aligning closely with analytical solutions. The architecture of KAFIN effectively captures the inherent complexities of option pricing, integrating financial principles with advanced computational techniques. Performance indicators reveal that KAFIN achieves lower average total losses and reduces the variability in loss components, highlighting its advantage in modeling and analyzing complex financial dependencies while ensuring compliance with regulatory standards.

Keywords: Finance-Informed Neural Networks (FINNs); Physics-Informed Neural Networks (PINNs); Kolmogorov–Arnold representation (KAR); option pricing; computational financial modeling; machine learning; risk assessment

check for updates

Citation: Liu, C.Z.; Zhang, Y.; Qin, L.; Liu, Y. Kolmogorov–Arnold Finance-Informed Neural Network in Option Pricing. *Appl. Sci.* **2024**, *14*, 11618. https://doi.org/10.3390/app142411618

Academic Editor: Andrea Prati

Received: 24 October 2024
Revised: 8 December 2024
Accepted: 9 December 2024
Published: 12 December 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The application of Physics-Informed Neural Networks (PINNs) in finance represents a burgeoning field at the intersection of computational methods and financial applications [1–7]. When incorporating finance-specific knowledge, constraints, and theories, these types of networks can also be named Finance-Informed Neural Networks (FINNs), which enable more accurate risk assessments, asset pricing, and portfolio management under regulated circumstances.

Implementing FINNs involves structuring neural networks to solve differential equations representing financial models. A composite loss function guides training, incorporating both data-driven terms and financial equations (e.g., Black–Scholes for option pricing [8–10]). This ensures models adhere to regulated financial dynamics for tasks like pricing derivatives, risk management, and optimizing investment strategies [11,12].

By integrating financial theories and constraints, FINNs align outputs with established financial knowledge. This departure, however, complicates their training compared to traditional neural networks. The intricate nature of their loss functions and heightened computational requirements pose challenges to their efficiency and interpretability, necessitating larger structures and training scale to maintain accuracy [13,14].

Studies in Kolmogorov–Arnold representation in machine learning [15–19] show promise in overcoming the challenges of FINNs with complexity deduction, representation simplification, and improvement in interpretability and performance in scientific computation. Thus, in this study, we propose the Kolmogorov–Arnold Finance-Informed Neural Network (KAFIN), a framework that integrates Kolmogorov–Arnold representation (KAR) with an FINN to address the challenges above. Option pricing is employed as a case study to showcase KAFIN's feasibility and efficiency. Key contributions of this work include the following:

- Integration of KAR with FINN: KAFIN integrates KAR within the FINN framework, enhancing its capability to accurately capture complex financial dynamics by decomposing multivariate functions into simpler components.
- Incorporation of Financial Rules and Constraints: KAFIN incorporates specific financial rules and constraints directly into its architecture, ensuring adherence to fundamental financial principles and improving model reliability and interpretability.
- Case Study Validation: We validate KAFIN through a practical case study, demonstrating its effectiveness in finance practices.
- Performance Comparison: Comparative analysis with existing FINN systems highlights KAFIN's superior accuracy, stability, and computational efficiency, underscoring its potential as a valuable tool in financial analytics and decision-making.

This paper firstly provides a brief introduction to the option pricing scenario and the Black–Scholes model. Then, we will summarize the existing FINN framework and discuss previous research on KAR and its associated machine learning, providing essential context for the readers.

2. Background and Related Work

Option pricing is a pivotal concept in financial markets, determining the theoretical value of derivative instruments known as options [20–23]. These instruments grant holders the right, but not the obligation, to buy (call option) or sell (put option) an underlying asset at a predetermined price within a specified timeframe. Rational option pricing [20,24] ensures fair valuation, supporting informed capital allocation decisions while facilitating effective risk management and pre-empting arbitrage opportunities to promote market stability.

The Black–Scholes (BS) model is pivotal in financial mathematics, offering a structured approach to valuing European options through the modeling of underlying asset price dynamics [10,25–28]. Pricing with the BS model computes theoretical option values, reflecting probabilities of various asset price outcomes at maturity. While foundational to modern finance, ongoing research into FINNs aims to enhance option pricing by accommodating complex financial systems, nonlinear relationships, and empirical data integration, thereby improving adaptivity and bolstering the reliability of option pricing models [1,2,4,6,12].

A straightforward approach to developing Finance-Informed Neural Networks (FINNs) involves incorporating multi-layer perceptrons (MLPs), which introduce nonlinearity through activation functions, enabling the network to model complex financial patterns by mapping inputs to outputs [2,29]. This method is straightforward and effective, yet for more intricate financial scenarios, careful design of loss functions and learning strategies is crucial to prevent low efficiency or underfitting.

To enhance the efficacy of acquiring intricate financial insights, many studies have integrated deep neural networks (DNNs), characterized by multiple hierarchical layers, into FINNs (e.g., refs. [6,12]). This approach incorporates regulatory frameworks and constraints within the FINN architecture, thereby augmenting learning outcomes. However, such methodologies typically amplify system scale and complexity, necessitating heightened computational resources for effective training processes.

Another way to improve the performance is to use Feed-Forward Neural Networks (or Feed-Forward Networks, FFNs), which are fundamental neural architectures where data flows from input nodes through hidden layers to output nodes. They are valued in Finance-Informed Neural Networks (FINNs) for their hierarchical structure and capability to process complex constraint patterns (e.g., refs. [1,11]). Yet, deep training an FFN can encounter challenges like gradient vanishing or exploding, requiring careful management through techniques like gradient clipping or better initialization strategies.

In efforts to enhance the learning efficiency of an FINN without enlarging network dimensions, integrating the residual gain (REG) computation strategy into its learning framework shows its feasibility (e.g., refs. [4,30]). This approach involves augmenting the original MLP network with residual connections, where each hidden layer's output serves as residual learning to refine the MLP's output via correctional gains. Empirical findings indicate that this methodology effectively enhances learning efficiency without necessitating additional network depth.

As a basic conclusion of multivariate function theory, the Kolmogorov–Arnold representation theorem states that any multivariate continuous function can be represented as a superposition of continuous functions of several variables [18,19]. The corresponding representation is also called the Kolmogorov superposition representation. It breaks down the complex functions into manageable components as an interpretable representation. Inspired by the KAR mechanism, many works tried to incorporate the KAR into machine learning (e.g., refs. [15,17]) to simplify the complexity of the machine learning structure with improvement in modeling interpretation by mapping the decomposition of the KAR components to the neurons in the network to represent intricate relationships of the complex target function with the progress of the learning process.

3. Methodology

3.1. Modeling

The elements listed in Table 1 provide essential components for the financial modeling framework, each serving a specific function in the overall structure.

Table 1. List of symbols used in the financial modeling context.

Symbol	Description
g_i	Univariate functions in Kolmogorov–Arnold representation
h_i	Mappings that define how each univariate function interacts with each variable
N	Number of terms in the superposition, which can be finite or infinite
$\mathcal{F}$	Financial differential operator
$\mathbf{u}(\mathbf{x}, t)$	Financial variable of interest (e.g., option price, asset value)
$\mathbf{u}_\theta(\mathbf{x}, t)$	Neural network representation of the financial variable of interest
λ	Parameters of the financial model
Ω	Domain of financial variables
$[0, T]$	Time period of interest

The univariate functions, denoted by g_i, are foundational to the Kolmogorov–Arnold representation, which allows for the decomposition of complex multivariate functions into simpler one-dimensional components. This representation is particularly valuable in financial modeling, as it facilitates the capture of intricate relationships between variables while maintaining computational efficiency. The mappings h_i further define the interactions between these univariate functions and the financial variables at play, elucidating how each function contributes to the model's predictive capacity. This interaction is crucial, as it informs how changes in one variable might propagate through the system, affecting the overall outcome.

The symbol N indicates the number of terms in the superposition of functions, which can be finite or infinite. The choice of N directly impacts the model's complexity and accuracy. A larger N allows for a more nuanced representation of financial phenomena but may also lead to overfitting if not managed properly. Conversely, a smaller N might simplify the model at the expense of capturing essential dynamics. Therefore, determining an optimal N is critical for balancing complexity and interpretability.

The financial differential operator, $\mathcal{F}$, is employed to analyze the dynamics of financial variables, enabling the exploration of their behavior over time under various market conditions. This operator plays a vital role in expressing the continuous evolution of financial instruments, allowing the model to incorporate essential features such as volatility and drift.

The variable $\mathbf{u}(\mathbf{x}, t)$ represents the financial variable of interest, such as an option price or asset value, serving as the model's primary output. Its accurate estimation is crucial for effective decision-making in finance. In contrast, $\mathbf{u}_\theta(\mathbf{x}, t)$ denotes the neural network representation of this variable, leveraging deep learning techniques to enhance the model's predictive capabilities. The use of neural networks allows for capturing nonlinear relationships that traditional methods may overlook, thereby improving the robustness of predictions.

Parameters λ govern the dynamics and relationships among the variables within the financial model. Proper calibration of these parameters is essential for aligning the model with real-world data, ensuring that predictions are both reliable and relevant. Additionally, Ω defines the domain of financial variables, setting boundaries for the model's applicability. This ensures that the model operates within a realistic range, enhancing its practical utility.

The time interval $[0, T]$ specifies the period of interest for the financial analysis, establishing a temporal framework for evaluating the dynamics of the modeled variables. This temporal aspect is critical, as financial markets are inherently dynamic, and the impact of events can vary significantly over time.

These elements form the foundation of a comprehensive financial modeling approach, facilitating the integration of mathematical rigor with practical financial analysis. This integration not only enhances the model's interpretability but also its applicability in real-world financial decision-making contexts, thereby bridging the gap between theory and practice.

3.2. Kolmogorov–Arnold Representation

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuous function. According to the Kolmogorov–Arnold representation theorem, there exist continuous functions $\phi_k : \mathbb{R}^{n-1} \to \mathbb{R}$ for $k = 1, 2, \ldots, n$, such that f can be expressed as:

$$f(x_1, x_2, \ldots, x_n) = \sum \int_{\mathbb{R}^{n-1}} \phi_k(y_2, \ldots, y_n)\, dy_2 \ldots dy_n$$

where each ϕ_k depends only on the respective variable x_k and possibly on the other variables x_j for $j \neq k$.

In other words, f can be written as a superposition of functions of fewer variables:

$$f(x_1, x_2, \ldots, x_n) = \sum_{k=1}^{n} \phi_k(x_k, x_{-k})$$

where $x_{-k} = (x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_n)$.

The Kolmogorov–Arnold representation (KAR) theorem states that any continuous multivariate function can be expressed as a superposition of univariate functions. For a continuous function $f(x_1, x_2, \ldots, x_n)$, KAR asserts the following representation as

$$f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{N} g_i(h_i(x_1), h_i(x_2), \ldots, h_i(x_n)) \tag{1}$$

where g_i refers to univariate functions, h_i refers to mappings that define how each univariate function interacts with each variable, and N refers to the number of terms in the superposition, which can be finite or infinite. Expanding on the theorem, KAR further specifies that any continuous function $f : \mathbb{R}^n \to \mathbb{R}$ can be expressed in a more specific form as

$$f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{2n+1} g_i\left(\sum_{j=1}^{n} h_{ij}(x_j)\right)$$

where g_i and h_{ij} are continuous univariate functions. In this form, each g_i is applied to a sum of transformed input variables $h_{ij}(x_j)$, where j indexes over all the input variables $x_1, x_2, \ldots, x_n$. The number of terms $2n + 1$ reflects the minimum number of components needed to represent the multivariate function, based on the structure of the input space and the interaction of the variables.

With the superposition principle, KAR simplifies the representation of complex multivariate functions into a sum of simpler univariate components of KAR. It simplifies the solving of complex differential equations by breaking them down into simpler parts. In computational contexts, KAR reduces the computational complexity of high-dimensional problems by decomposing them into lower-dimensional operations. It facilitates effective nonlinear approximations, where complex relationships between variables can be approximated by simpler functions, aiding in interpretability and efficiency.

This representation captures the idea that a complex multivariate function can be constructed by summing over simpler, univariate transformations of the individual variables. The functions $h_{ij}(x_j)$ capture the individual contributions of each variable x_j to the overall function, while the functions g_i aggregate these contributions into a final output. The decomposition through KAR is crucial for understanding and modeling complex systems where multiple input variables interact in a nonlinear fashion.

3.3. Machine Learning-Based KAR Decomposition

The mathematical elegance of KAR representation lies in its ability to break down a high-dimensional function into a sum of lower-dimensional components, which can then be modeled independently. This is particularly useful in machine learning and numerical approximation, where the goal is to learn and optimize the parameters of these univariate functions to best approximate a complex target function. The decomposition also facilitates the understanding of how different input variables contribute to the system, which is essential for applications such as financial modeling, where understanding variable interactions is key to predicting outcomes.

When applying KAR to time-dependent systems, such as in financial modeling, a function $\mathbf{u}(\mathbf{x}, t)$, depending on input variables $\mathbf{x}$ and time t, can similarly be decomposed as

$$\mathbf{u}(\mathbf{x}, t) = \sum_{i=1}^{L_2} g_i\left(\sum_{j=1}^{L_1} h_{ij}(x_j, t)\right),$$

where L_1 and L_2 represent the number of inner and outer units, respectively.

In a parametric modeling framework, $\mathbf{u}(\mathbf{x}, t)$ can be approximated by a parameterized machine learning system $\mathbf{u}_\theta(\mathbf{x}, t)$ with learnable parameters θ. Using the KAR-inspired structure, it can be formulated as

$$\mathbf{u}_\theta(\mathbf{x}, t) = \sum_{i=1}^{L_2} g_{i,\theta}\left(\sum_{j=1}^{L_1} h_{ij,\theta}(x_j, t)\right),$$

where $g_{i,\theta}$ and $h_{ij,\theta}$ are parameterized univariate functions learned during training.

The goal of the machine learning process is to determine the parameters θ such that the network output $\mathbf{u}_\theta(\mathbf{x}, t)$ closely approximates the true system solution $\mathbf{u}(\mathbf{x}, t)$. This is achieved by minimizing a total loss function $\mathcal{L}(\theta)$, which measures the deviation from governing system rules:

$$\theta^* = \arg\min_{\theta} \mathcal{L}(\theta).$$

The network architecture, inspired by the Kolmogorov–Arnold representation, utilizes distinct components. The inner functions $h_{ij,\theta}$ transform input variables x_j and time t, capturing their localized contributions to system dynamics. These are aggregated by the outer functions $g_{i,\theta}$, which synthesize these contributions into a global representation via summation. Learnable parameters θ control the weighting, scaling, and interactions, enabling the model to effectively adapt to complex relationships while maintaining interpretability.

4. Kolmogorov–Arnold Finance-Informed Neural Network

The decomposition provided by KAR enables the network to align with financial mechanisms by representing system dynamics as combinations of univariate transformations. This approach facilitates learning even in scenarios with sparse or uncertain data. The modular construction of the loss function ensures solutions remain consistent with theoretical models and boundary conditions, enhancing both robustness and interpretability.

4.1. Minimal Viable KAFIN

Based on the KAR theory, Finance-Informed Neural Networks (FINNs) integrate financial theories and constraints directly into neural network architectures to enhance predictive accuracy and model reliability in financial applications. KAFIN serves as a type of Finance-Informed Neural Network (FINN) with Kolmogorov–Arnold representation (KAR), integrating financial domain knowledge and sophisticated mathematical frameworks to enhance the predictive capabilities and interpretability of financial models.

Consider a financial system characterized by a set of differential equations representing financial dynamics, such as option pricing models or portfolio optimization:

$$\mathcal{F}[\mathbf{u}(\mathbf{x}, t); \lambda] = 0, \quad \mathbf{x} \in \Omega, \quad t \in [0, T] \tag{2}$$

where $\mathcal{F}$ denotes the financial differential operator, $\mathbf{u}(\mathbf{x}, t)$ represents the financial variable of interest (e.g., option price, asset value), λ includes parameters of the financial model, Ω represents the domain of financial variables, and $[0, T]$ denotes the time period of interest.

Kolmogorov–Arnold representation serves as a theoretical backbone for constructing KAFIN that approximates solutions to complex systems. By minimizing the total loss function, the parameters θ are optimized to ensure that network predictions satisfy initial, boundary, and governing conditions. This approach combines mathematical rigor with computational efficiency, offering a robust methodology for addressing financial modeling challenges.

In KAFIN, the neural network $\mathbf{u}_\theta(\mathbf{x}, t)$ is constructed as a Kolmogorov–Arnold Network (KAN) with parameters θ to approximate the solution $\mathbf{u}(\mathbf{x}, t)$. Utilizing KAR, Equation (1) can be represented as a network to reflect the decomposition as

$$\mathbf{u}_\theta(\mathbf{x}, t) = \sum_{i=1}^{L_2} g_{i,\theta}\left(\sum_{j=1}^{L_1} h_{ij,\theta}(x_j, t)\right) \tag{3}$$

where $g_{i,\theta}(\cdot)$ and $h_{ij,\theta}(\cdot)$ are learned by the network, and L_1, L_2 are the numbers of the function units used in layer 1 and 2. By learning the parameters, KAFIN can decompose any continuous multivariate function into the sum of univariate functions of linear combinations of the input variables.

The parameters $\theta, g_{i,\theta}(\cdot)$ and $h_{ij,\theta}(\cdot)$ are optimized by minimizing the total loss function, which can be formulated as

$$\theta^* = \arg\min_{\theta} \mathcal{L}(\theta) \tag{4}$$

The loss function in the context of financial constraints and system comprises several key components, including initial value loss, boundary loss, and financial governing loss.

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{initial}}(\theta) + \mathcal{L}_{\text{boundary}}(\theta) + \mathcal{L}_{\text{financial}}(\theta) \tag{5}$$

in which the initial value loss $\mathcal{L}_{\text{initial}}(\theta)$ ensures that the neural network's prediction at the initial time $t = 0$ matches the given initial condition; the boundary loss $\mathcal{L}_{\text{boundary}}(\theta)$ ensures that the neural network solution satisfies the boundary conditions of the financial domain; the financial governing loss $\mathcal{L}_{\text{financial}}$ enforces adherence to the financial models and constraints.

The formulation of the loss function is crucial in developing a robust minimal viable KAFIN. Integration of convex learning mechanisms [31] ensures that all loss functions satisfy convex constraints related to initial, boundary, and financial considerations. The mean squared error (MSE) metric is particularly effective in this regard, as it precisely measures the average disparities between estimated outputs and their true references, offering a straightforward and robust framework for defining the loss function.

The corresponding initial value loss $\mathcal{L}_{\text{initial}}(\theta)$, boundary loss $\mathcal{L}_{\text{boundary}}(\theta)$, and financial governing loss $\mathcal{L}_{\text{financial}}$ can be formulated as follows

$$\mathcal{L}_{\text{initial}}(\theta) = \frac{1}{N_{init}} \sum_{i=1}^{N_{init}} |\mathbf{u}_\theta(\mathbf{x}_i, 0) - \mathbf{u}_0(\mathbf{x}_i)|^2 \tag{6}$$

where $\mathbf{u}_\theta(\mathbf{x}_i, 0)$ is the neural network's prediction at the initial time $t = 0$, $\mathbf{u}_0(\mathbf{x}_i)$ is the known initial condition at point $\mathbf{x}_i$, and N_{init} is the number of initial condition points.

$$\mathcal{L}_{\text{boundary}}(\theta) = \frac{1}{N_b} \sum_{j=1}^{N_b} \left|\mathbf{u}_\theta(\mathbf{x}_j, t_j) - \mathbf{b}(\mathbf{x}_j, t_j)\right|^2 \tag{7}$$

where $\mathbf{u}_\theta(\mathbf{x}_j, t_j)$ is the neural network's prediction at boundary point $(\mathbf{x}_j, t_j)$, $\mathbf{b}(\mathbf{x}_j, t_j)$ is the known boundary condition at point $(\mathbf{x}_j, t_j)$, and N_b is the number of boundary points.

$$\mathcal{L}_{\text{financial}}(\theta) = \frac{1}{N_f} \sum_{k=1}^{N_f} |\mathcal{F}[\mathbf{u}_\theta(\mathbf{x}_k, t_k); \lambda]|^2 \tag{8}$$

where $\mathcal{F}$ is the financial differential operator, $\mathbf{u}_\theta(\mathbf{x}_k, t_k)$ represents the neural network's prediction at point $(\mathbf{x}_k, t_k)$, λ includes parameters of the financial model, and N_f is the number of points used to enforce the financial constraints.

4.2. Generalization of Deep KAFIN

We can further generalize the KAFIN neural network representation (3) to deeper cases. Let the first layer be $\Phi_1(\cdot) = \sum g(\cdot)$ and the second layer $\Phi_2(\cdot) = \sum h(\cdot)$. Equation (3) can be represented as

$$\mathbf{u}_\theta(\mathbf{x}, t) = \sum_{i=1}^{L_2} g_{i,\theta}\left(\sum_{j=1}^{L_1} h_{ij,\theta}(x_j, t)\right) = \Phi_2 \circ \Phi_1 z \tag{9}$$

where $\Phi_2 = [g_1(\cdot), g_2(\cdot), \cdots, g_{L_2}(\cdot)]$, and

$$\Phi_1 = \begin{bmatrix} h_{1,1}(\cdot) & h_{1,2}(\cdot) & \cdots & h_{1,L_1}(\cdot) \\ h_{2,1}(\cdot) & h_{2,2}(\cdot) & \cdots & h_{2,L_1}(\cdot) \\ \vdots & \vdots & \ddots & \vdots \\ h_{L_2,1}(\cdot) & h_{L_2,2}(\cdot) & \cdots & h_{L_2,L_1}(\cdot) \end{bmatrix} \tag{10}$$

$$z = \left[(x_1, t)(x_2, t) \ldots (x_{L_1}, t)\right]^T \tag{11}$$

Similarly, Φ_k can be defined to perform the deep KAFIN that can be formulated as

$$\mathbf{u}_\theta(\mathbf{x}, t) = \Phi_L \circ \Phi_{L-1} \cdots \circ \Phi_k \circ \cdots \circ \Phi_1 z \tag{12}$$

in which

$$\Phi_k = \begin{bmatrix} \Phi_{1,1}(\cdot) & \Phi_{1,2}(\cdot) & \cdots & \Phi_{1,L_k}(\cdot) \\ \Phi_{2,1}(\cdot) & \Phi_{2,2}(\cdot) & \cdots & \Phi_{2,L_k}(\cdot) \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_{L_{k+1},1}(\cdot) & \Phi_{L_2,2}(\cdot) & \cdots & \Phi_{L_{k+1},L_k}(\cdot) \end{bmatrix} \tag{13}$$

The general loss function can be formulated as a weighted measure of a series of sub-loss functions $\mathcal{L}_l(\theta)$, which can be formulated as

$$\mathcal{L}(\theta) = \sum \lambda_l \mathcal{L}_l(\theta) \tag{14}$$

When considering the initial value loss, boundary loss, and financial governing loss, the corresponding generalized loss can be formulated as

$$\mathcal{L}(\theta) = \lambda_{\text{initial}} \mathcal{L}_{\text{initial}}(\theta) + \lambda_{\text{boundary}} \mathcal{L}_{\text{boundary}}(\theta) + \lambda_{\text{financial}} \mathcal{L}_{\text{financial}}(\theta) \tag{15}$$

For $\mathcal{L}_{\text{initial}}$, $\mathcal{L}_{\text{boundary}}$, and $\mathcal{L}_{\text{financial}}$, the corresponding parameters are λ_{initial}, $\lambda_{\text{boundary}}$, and $\lambda_{\text{financial}}$ as regularization coefficients that control the relative importance of each loss component.

4.3. Analytic Remarks

4.3.1. Modeling

The KAFIN framework employs the Kolmogorov–Arnold representation (KAR) to approximate the solution $\mathbf{u}(\mathbf{x}, t)$ in the context of financial modeling. By leveraging an informed neural network designed with KAR principles, the framework decomposes the multivariate function $\mathbf{u}(\mathbf{x}, t)$ into a sum of simpler univariate functions, each dependent on linear combinations of input variables. This approach enables the network to efficiently capture and represent complex multivariate relationships inherent in financial systems.

In the context of financial modeling, the input variables $\mathbf{x}$ may include crucial factors such as asset prices, time to maturity, volatility, and interest rates, which collectively determine the value of financial derivatives such as options. The use of KAR ensures that these relationships are systematically captured by the network, providing a robust mechanism for approximating solutions in high-dimensional spaces.

This structured decomposition not only augments the interpretability of the model by integrating computational frameworks with established financial principles, but also empowers the machine learning system to execute intelligent computations grounded in mechanistic rigor. By aligning the learning process with the intrinsic dynamics of financial systems, this approach enables the model to capture and replicate the regular, mechanistic evolution of complex financial phenomena, even in scenarios characterized by data uncertainty or scarcity. Such a design ensures that the system remains robust and consistent with theoretical underpinnings, thereby offering reliable predictions and insights despite limited or noisy data.

4.3.2. Optimization

The machine learning process can be conceptualized as the optimization of system parameters with respect to a defined objective function. In this context, the objective is not to achieve alignment with predefined data-driven target outcomes but rather to adhere to the governing principles of financial mechanisms. Consequently, all objective functions and the corresponding loss quantifications are designed to evaluate whether the model's outputs align with the theoretical laws of financial systems.

The goal of machine learning in this framework is to minimize the discrepancy between the behavior of the system and the expected behavior dictated by financial mechanism laws. Unlike conventional scenarios that compare system outputs directly with empirical target results, this study operates in a setting devoid of explicit target data. Instead, the approach involves substituting the system's outputs into the governing financial mechanisms, evaluating these outputs against the theoretical ideal values, and quantifying the resulting errors. The learning process iteratively reduces these errors through optimization, progressively aligning the system's behavior with the target financial mechanisms.

The optimization model proposed in this study employs a composite loss function that integrates three critical components: the initial value loss, boundary value loss, and financial governance loss. Each component corresponds to specific constraints and conditions inherent in real-world financial problems. Collectively, these components serve complementary roles, guiding the model to generate accurate solutions that conform to the precise requirements of financial systems. This integration ensures that the system captures the nuanced interplay between theoretical laws and practical constraints, leading to robust and reliable modeling outcomes.

4.3.3. Initial Value Loss

The initial value loss ensures that the system predictions at the initial time, $t = 0$, match the known initial conditions of the financial system. For example, in the context of option pricing, the initial value condition could represent the option's price at inception, i.e., at $t = 0$. This initial value loss is crucial because it provides a necessary baseline for the network's evolution over time. In financial modeling, especially in derivative pricing, ensuring that the model begins with correct values is essential for the accuracy of future predictions. The initial price of an option must reflect the market's conditions at the start of the contract, and this loss term helps the network align with this starting condition.

4.3.4. Boundary Value Loss

The boundary value loss ensures that the system solution satisfies the boundary conditions, which are critical in financial modeling. For example, in option pricing, boundary conditions often correspond to the payoff of an option at maturity. For a European call option, this could be defined as the maximum of the asset price at maturity minus the strike price, or zero. Boundary conditions also include other constraints that may apply at the edges of the domain, such as constraints on the asset price or time. This component of the loss ensures that the network is consistent with the known outcomes at the boundaries of the model, thus providing a realistic and accurate model of the financial system. The importance of this loss in option pricing cannot be overstated, as it guarantees that the predicted option price reflects the expected payoff at maturity, which is the final step in determining the option's value.

4.3.5. Financial Governing Loss

The financial governing loss enforces that the system solution adheres to the differential equations governing the financial system, such as the Black–Scholes equation in the context of options pricing. The Black–Scholes equation describes the relationship between the price of an option and the underlying asset, time to maturity, volatility, and other market factors. This governing loss ensures that the network's output respects the theoretical structure of the financial model. The underlying differential equation is a cornerstone of modern financial modeling, particularly in the pricing of options and derivatives. By enforcing this loss, we ensure that the network does not merely fit the data at the initial and boundary conditions but also respects the governing financial principles that drive the evolution of the financial system over time.

4.3.6. Total Loss Function and Optimization Process

The total loss function is constructed as a weighted sum of these individual loss components. The inclusion of each loss term ensures that the network's solution meets the specific requirements of the financial problem, ensuring that the predictions are consistent with the initial conditions, boundary conditions, and the underlying financial model. The weights of these components are adjusted during training to balance the influence of each term, which ensures that the network can learn an optimal solution that satisfies all constraints. The optimization process seeks to minimize the total loss function by adjusting the parameters of the Kolmogorov–Arnold network.

In addition to the individual loss components, convex learning mechanisms are integrated into the training process. These mechanisms ensure that the optimization procedure converges reliably to a global minimum, providing stability and efficiency in the training of the network. The mean squared error (MSE) is commonly used to quantify the discrepancies between the network's predictions and the true values for each loss component. The MSE metric is simple but effective, as it measures the average squared differences between the predicted and actual values, providing a clear criterion for optimization.

4.3.7. Model and Financial Meaning

The optimization process connects the mathematical model (i.e., the neural network representation) to the financial meaning through the loss components. By minimizing the total loss function, the network learns to approximate the solution to the financial problem in a way that respects both the financial constraints (such as initial and boundary conditions) and the financial principles (such as the Black–Scholes equation). This is analogous to solving a differential equation with boundary conditions, where the objective is to find a function that satisfies both the equation and the boundary conditions.

In option pricing, the Kolmogorov–Arnold network captures the multivariate relationships that affect the option price over time, such as changes in the underlying asset price and time to maturity. By incorporating the financial governing loss, the network is constrained to adhere to the theoretical financial model, ensuring that its output is not only consistent with observed market data but also with the well-established financial theories.

The KAFIN framework allows for the approximation of complex financial models, such as option pricing, through the use of neural networks. The loss function's integration of initial, boundary, and financial losses ensures that the network's output is both accurate and consistent with financial principles. The optimization process drives the learning of the network, ensuring that the final solution adheres to the necessary constraints while approximating the true value of financial derivatives.

5. KAFIN for Option Pricing

5.1. European Option Pricing

In financial markets, options offer investors the right to buy or sell an underlying asset at a predetermined price within a specified period. European options, known for their simplicity and fixed expiration date, are favored for cost-effective hedging strategies and stability amid low market volatility. Institutional investors often utilize European options for precise risk management and predictable cash flows, although their fixed nature limits flexibility in responding to market changes before expiration.

The Black–Scholes (BS) model, established by Fischer Black and Myron Scholes in 1973, is pivotal in financial mathematics for pricing European options. This model provides a closed-form solution based on key variables such as the underlying asset's current price, the option's strike price, time to expiration, risk-free interest rate, and asset volatility. By employing partial differential equations, the BS model accurately predicts option prices, enabling traders to assess fair values, hedge risks, and execute effective derivative trading strategies in dynamic markets.

5.2. KAFIN for Option Pricing

For European option pricing, KAFIN leverages the fundamental principles of the Black–Scholes (BS) model while incorporating data-driven neural network techniques to enhance prediction accuracy and model robustness. This case study illustrates how KAFIN can be employed to price European options effectively as a case in point for finance applications.

In the context of European option pricing, the KAFIN model begins by defining the financial differential operator based on the BS partial differential equation (PDE). The BS equation for a European call option is given by

$$f_{BS} = \frac{\partial C}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} + rS\frac{\partial C}{\partial S} - rC = 0 \tag{16}$$

where C represents the option price, S is the underlying asset price, σ is the volatility, r is the risk-free interest rate, and t denotes time. Equation (16) serves as financial governing constraints in PDE form.

By applying (12)–(15), we build the KAFIN to learn the KAR to decompose the option price $C(S,t)$ using KAR into a series of simpler functions:

$$C(S,t) = \sum_{i=1}^{N} g_i(h_i(S),t), \tag{17}$$

where g_i are univariate functions, and $h_i(x)$ denote the mappings of these functions with respect to S and t. This decomposition allows for a more granular and interpretable representation of the option price dynamics. The neural network component of KAFIN is designed to learn the parameters of these univariate functions and their interactions. The network is trained using a composite loss function that includes initial value loss, boundary loss and financial governing constraint loss.

The financial governing constraint loss $\mathcal{L}_{\text{financial}}(\theta)$ is built based on the (16) as a constraints to enforces the BS PDE constraints within the domain.

5.2.1. Financial Governing Loss for Call Option Pricing

To mathematically model the financial governing loss for call option pricing using the Black–Scholes (BS) model within a neural network framework, we need to ensure that the network's output adheres to the BS partial differential equation (PDE). The loss function should penalize deviations from the BS model, thereby enforcing the financial constraints.

Let the neural network output for the call option price be $u_\theta(S,t)$, where θ denotes the network parameters. The financial governing loss can be constructed by computing the residual of the BS PDE when applied to the network's output. This residual should ideally be zero across the domain of interest. The residual $\mathcal{R}(S,t)$ is given by

$$\mathcal{R}(S,t) = \frac{\partial u_\theta}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 u_\theta}{\partial S^2} + rS\frac{\partial u_\theta}{\partial S} - ru_\theta. \tag{18}$$

The financial governing loss $\mathcal{L}_{\text{financial}}$ can then be defined as the mean squared error of the residual over the sampled points (S_i, t_i) in the domain:

$$\mathcal{L}_{\text{financial}} = \frac{1}{N}\sum_{i=1}^{N} \mathcal{R}^2(S_i,t_i), \tag{19}$$

where N is the number of sample points.

By integrating this financial governing loss into the overall loss function of the neural network, we ensure that the network's output respects the financial dynamics dictated by the Black–Scholes model.

5.2.2. Initial Condition for European Call Option

The initial value loss for call option pricing, using a neural network, ensures that the network's output satisfies the initial conditions of the option pricing model. For a European call option, the initial condition is determined by the payoff function at the time of option issuance (time $t = 0$). The initial condition for a European call option at $t = 0$ is given by the payoff function:

$$C(S, 0) = \max(S - K, 0), \tag{20}$$

where $C(S, t)$ is the option price, S is the underlying asset price, and K is the strike price.

The initial value loss $\mathcal{L}_{\text{initial}}$ ensures that the neural network output at $t = 0$ matches the initial condition $\max(S - K, 0)$. Mathematically, this loss can be defined as the mean squared error between the neural network's predicted option prices and the actual initial condition over a set of sampled asset prices S_i. Let $u_\theta(S, t)$ be the output of the neural network with parameters θ, representing the option price, and the initial loss can be formulated as

$$\mathcal{L}_{\text{initial}} = \frac{1}{M} \sum_{i=1}^{M} (u_\theta(S_i, 0) - \max(S_i - K, 0))^2, \tag{21}$$

where M is the number of sampled points S_i.

5.2.3. Boundary Loss for Call Option Pricing

To mathematically model the boundary loss for call option pricing using a neural network, we ensure that the network's output satisfies the boundary conditions of the option pricing model. For a European call option, the boundary conditions are determined by the behavior of the option price as the underlying asset price approaches zero and infinity, and at the expiration time.

For a European call option, the boundary conditions include zero boundary and infinity boundary. For zero boundary, as the underlying asset price S approaches zero, the option price should also approach zero:

$$C(0, t) = 0 \quad \forall t \tag{22}$$

for infinity boundary, as the underlying asset price S approaches infinity, the option price should asymptotically approach $S - K$:

$$\lim_{S \to \infty} C(S, t) = S - K \quad \forall t \tag{23}$$

The boundary loss L_{boundary} ensures that the neural network output satisfies these boundary conditions. Mathematically, this loss can be defined as the sum of the mean squared errors at the boundary points.

The zero boundary loss can be formulated as

$$\mathcal{L}_{\text{boundary},0} = \frac{1}{N_0} \sum_{i=1}^{N_0} (u_\theta(0, t_i))^2, \tag{24}$$

where N_0 is the number of sampled points t_i. The infinity boundary can be formulated as

$$\mathcal{L}_{\text{boundary},\infty} = \frac{1}{N_\infty} \sum_{i=1}^{N_\infty} (u_\theta(S_i, t) - (S_i - K))^2 \tag{25}$$

where N_∞ is the number of sampled points S_i.

The total boundary loss $\mathcal{L}_{\text{boundary}}$ is the sum of these individual losses:

$$\mathcal{L}_{\text{boundary}} = \mathcal{L}_{\text{boundary},0} + \mathcal{L}_{\text{boundary},\infty}. \tag{26}$$

5.2.4. Implementation in the Loss Function

We collect the loss function results as a tensor $\mathcal{L}$ that

$$\mathcal{L} = [\mathcal{L}_{\text{initial}} \quad \mathcal{L}_{\text{boundary},0} \quad \mathcal{L}_{\text{boundary},\infty} \quad \mathcal{L}_{\text{financial}}] \tag{27}$$

The corresponding weight for each loss is enclosed in a tensor λ that

$$\lambda = [\lambda_{\text{initial}} \quad \lambda_{\text{boundary},0} \quad \lambda_{\text{boundary},\infty} \quad \lambda_{\text{financial}}] \tag{28}$$

The total loss is obtained by

$$\mathcal{L}_{\text{total}} = \lambda \mathcal{L}^T \tag{29}$$

It comprises multiple components to ensure the network adheres to problem-specific conditions. In financial modeling, the initial value loss aligns the network's output $\mathbf{u}_\theta(\mathbf{x}, t)$ with known initial conditions at $t = 0$, providing a foundational baseline. The boundary value loss enforces compliance with constraints at domain boundaries, such as maturity conditions, ensuring consistency across the problem space. Lastly, the financial governing loss evaluates adherence to the governing differential equations, such as the Black–Scholes model, embedding theoretical accuracy into the solution.

6. Experiment and Results

As an example to show the feasibility of KAFIN, this experiment evaluates the performance of the Kolmogorov–Arnold Finance-Informed Neural Network (KAFIN) in European call option pricing, a crucial financial modeling problem. The Black–Scholes (BS) model, which provides a benchmark analytical solution, is used for comparison. KAFIN enhances computational efficiency and interpretability by leveraging the Kolmogorov–Arnold representation theorem to decompose multivariate functions into simpler univariate components. The experiment is conducted under realistic market conditions: a strike price of 40, a risk-free rate of 0.05, volatility of 0.2, and a maturity period of one year. Asset prices range from 0 to 160 with time intervals from present to maturity, generating 10,000 sample points. The experimental setup includes data generation, model training, and evaluation. By comparing KAFIN's predictions with BS model solutions, we aim to demonstrate KAFIN's accuracy and robustness, highlighting its potential in financial analysis and risk management.

6.1. Experiment Setting and Implementation

In this section, we detail the experimental setup for evaluating the performance of our proposed KAFIN framework in the context of European call option pricing and American put option pricing. The key parameters and settings used in the experiments are outlined below in Table 2.

The strike price for the option is set to $K = 40$. This is the fixed price at which the option holder can buy the underlying asset at maturity. The risk-free interest rate is set to $r = 0.05$. This rate represents the theoretical return on an investment with no risk of financial loss. The volatility of the underlying asset is $\sigma = 0.2$. Volatility measures the price fluctuations of the underlying asset over time. The time to maturity of the option is $T = 1$ year. This is the period over which the option can be exercised. The range of asset prices considered in the experiments spans from $S = 0$ to $S = 160$. This range is chosen to cover a broad spectrum of possible asset prices. The time range considered in the experiments extends from the current time ($t = 0$) to the maturity ($t = T$). This is expressed as $t \in [0, T]$. A total of 10,000 sample points ($N_{\text{sample}} = 10,000$) are used for training and testing the neural network model. These samples are uniformly distributed across the asset price and time ranges to ensure comprehensive coverage of the domain.

Table 2. Parameter settings for option pricing test.

Parameter	**Value**	**Parameter Description**
Strike Price (K)	60	Fixed price at which the option can be exercised.
Risk-Free Rate (r)	0.05	Theoretical return on a risk-free investment.
Volatility (σ)	0.2	Price fluctuation of the underlying asset.
Maturity (T)	1	Timeframe for option exercise.
Asset Price Range (S)	$S \in [0, 160]$	Range of asset prices considered.
Time Range (t)	$t \in [0, T]$	From current time to maturity.
Number of Samples (N_{sample})	10,000	Sample points for training and testing.
Finance Info Weight ($\lambda_{financial}$)	0.1	Finance-informed learning rate for training.
Initial Value Weight ($\lambda_{initial}$)	1	Initial constraint penalty for training.
Zero Boundary Weight ($\lambda_{initial}$)	1	Zero boundary penalty for training.
Infinity Boundary Weight ($\lambda_{initial}$)	1	Infinity boundary penalty for training.

To further validate the effectiveness of the KAFIN model, we compare its performance with FINN REG [4,30], FINN FNN [1,11], FINN DNN [6,12], and FINN MLP [2,29], as baselines under the same experiment setting. The following results show the performance after training with 2300 epochs.

These models were selected not only for their demonstrated effectiveness in financial contexts but also because they represent state-of-the-art approaches capable of addressing the key challenges of nonlinearity, constraint handling, and computational efficiency in financial modeling. We selected multi-layer perceptrons (MLPs) and deep neural networks (DNNs) for their proven ability to model complex, nonlinear relationships in financial data, which is a key requirement for capturing the intricate patterns and interactions inherent in financial systems. Feed-Forward Networks (FFNs) were included due to their effectiveness in processing multi-dimensional financial data and handling complex constraints within Finance-Informed Neural Networks (FINNs). Despite potential challenges like gradient vanishing, these can be mitigated with techniques such as gradient clipping. Additionally, the residual gain (REG) strategy was chosen for its ability to enhance learning efficiency without increasing computational complexity, making it particularly valuable in financial applications where computational feasibility is crucial.

All networks in this study share a common input layer width of 2 and an output layer width of 1. However, the architectures vary in the number of hidden layers and units, which are optimized to balance model complexity and computational efficiency.

The KAFIN architecture consists of a single hidden layer with 3 units. This streamlined design ensures computational efficiency while capturing the key dynamics of carbon emission and sequestration processes. The FINN DNN model employs 4 hidden layers, each with 20 units, to capture complex relationships in high-dimensional data. While this deep architecture enhances the model's ability to learn intricate patterns, it also increases the risk of overfitting due to its complexity. The FINN GRU model consists of 4 hidden layers with 4 units, utilizing recurrent layers to capture temporal dependencies in carbon dynamics. This compact design maintains computational efficiency while leveraging the advantages of recurrent networks for modeling sequential data. For comparison, other models use 4 hidden layers with 20 units each, providing a baseline for evaluating the performance of the KAFIN model against more traditional feed-forward architectures.

The architectural choices were made to strike a balance between computational efficiency, model complexity, and generalizability. The KAFIN model opts for a compact design with fewer hidden units to minimize computational cost while retaining predictive accuracy. In contrast, the FINN DNN, with its deep architecture, is designed to model more intricate data relationships, albeit at the cost of increased training time and potential overfitting. The FINN GRU, with fewer units and recurrent layers, captures temporal dependencies effectively while optimizing for computational efficiency.

The Adam optimizer was selected for its efficiency and adaptability in training deep learning models. Combining the benefits of both Adagrad and RMSprop, Adam adjusts learning rates for each parameter based on first and second moment estimates of the gradient, which accelerates convergence and helps prevent oscillations during training.

For the experiments, the learning rate was set to 1×10^{-5}. This relatively small value, chosen after extensive empirical testing, strikes a balance between convergence speed and model accuracy. A lower learning rate ensures stable convergence and minimizes the risk of overshooting the optimal point, although it may require additional training epochs to reach the desired level of accuracy. This choice was made to accommodate the complexity of carbon emission and sequestration models, ensuring gradual convergence while minimizing overfitting.

Results

In this section, we present the results of our KAFIN-based approach for European option pricing. Figures 1 and 2 show the learning results of KAFIN for both European call and American put pricing. The figure employs a color scheme with a gradient scale, where lighter shades correspond to higher values, and darker shades indicate lower values. After 2300 epochs of training, the KAFIN predictions are close to the analytical solutions of both as a benchmark. We further compare the performance of KAFIN with baselines, and Tables 3 and 4 show the values of the residuals between the analytical solutions and predicted solutions among all the baselines. KAFIN shows the lowest residuals after 2300 training epochs in all residual measures including mean, standard deviation (STD), minimum, absolute minimum, maximum, absolute maximum, and MSE. Figures 3 and 4 show a visualization of the residuals between methods and the analytical benchmark, in which, the residual of KAFIN is bounded within $[-3, +3]$ in European call option pricing, and $[-2, +2]$ in American put option pricing, while the rest of the methods still need more training.

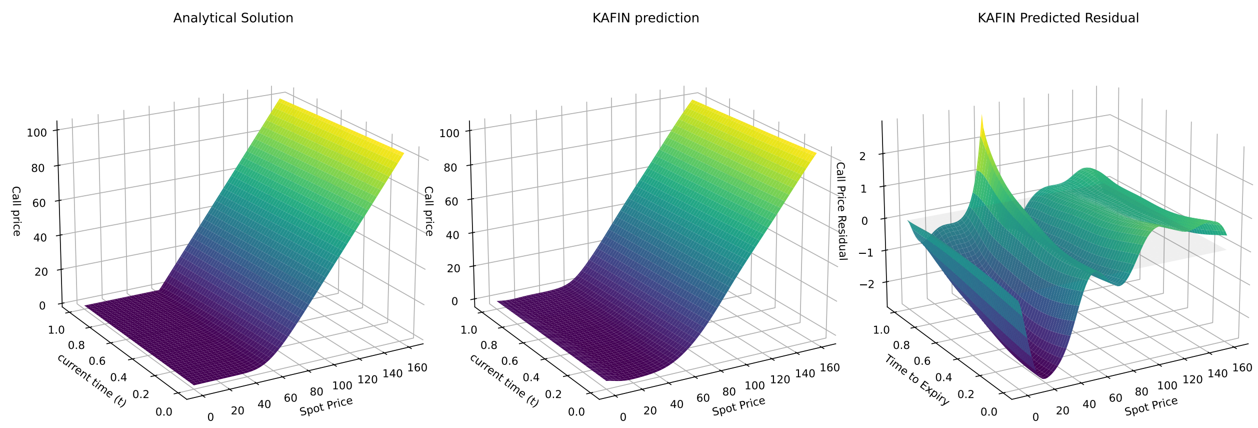


Figure 1. Comparison of predicted European call option pricing by KAFIN model and analytical solutions from Black–Scholes model.

Table 3. Residual Values on European call option pricing with 2300 epochs.

Residual	MEAN	STD	MIN	ABS MIN	MAX	ABS MAX	MSE
KAFIN	−0.2532	0.9578	−2.6867	0.0001	2.8779	2.8779	0.9814
FINN REG	−16.2222	23.7148	−73.0200	0.0002	15.1721	73.0200	825.4969
FINN FFN	−28.6290	32.1351	−97.0683	0.0003	4.1276	97.0683	1852.1833
FINN DNN	−22.9026	33.8007	−93.1510	0.0006	9.7752	93.1510	1666.8705
FINN MLP	−30.4478	33.8083	−100.7151	0.0040	2.1933	100.7151	2069.9526

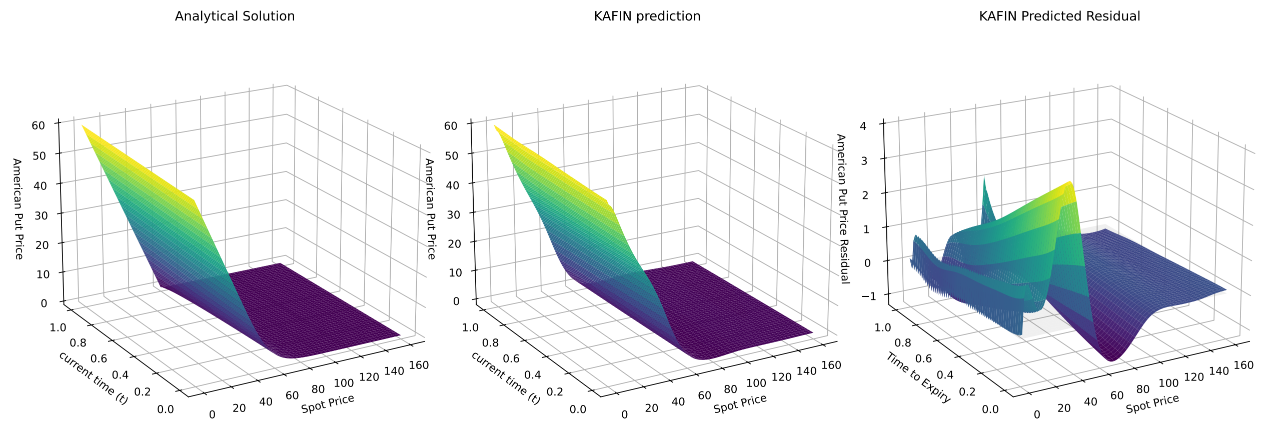
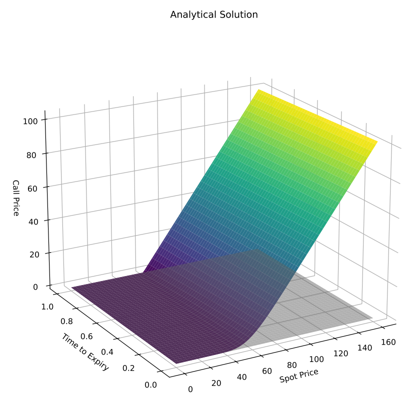
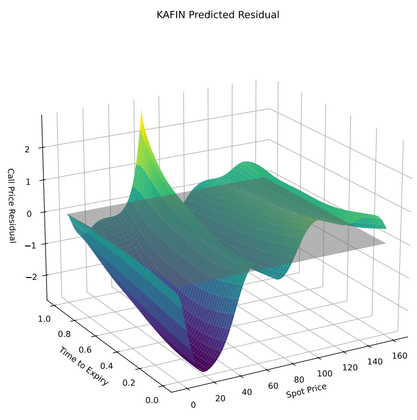


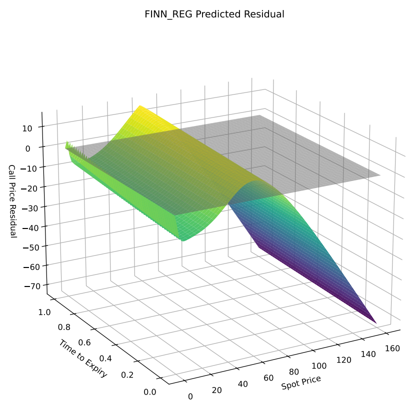
Figure 2. Comparison of predicted American put option pricing by KAFIN model and analytical solutions from Black–Scholes model.



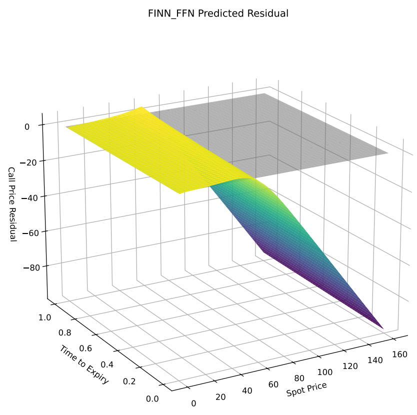
(**a**) Analytical solution



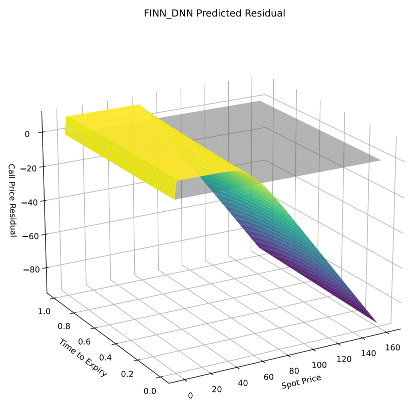
(**b**) Residual of KAFIN



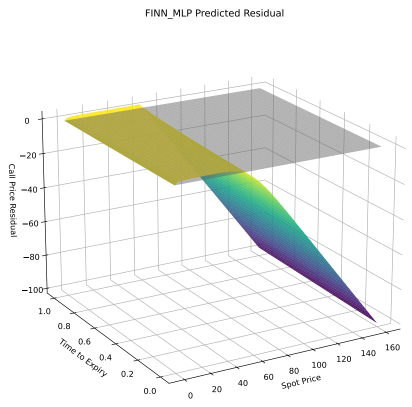
(**c**) Residual of FINN REG



(**d**) Residual of FINN FNN



(**e**) Residual of FINN DNN

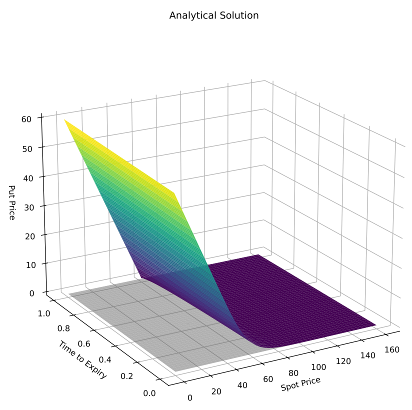


(**f**) Residual of FINN MLP

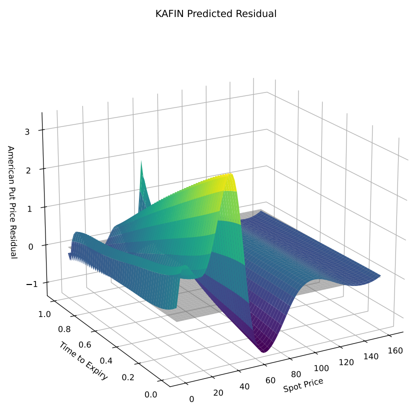
Figure 3. Overall residual comparison among methods.

Table 4. Residual Values on American put option pricing with 2300 epochs.

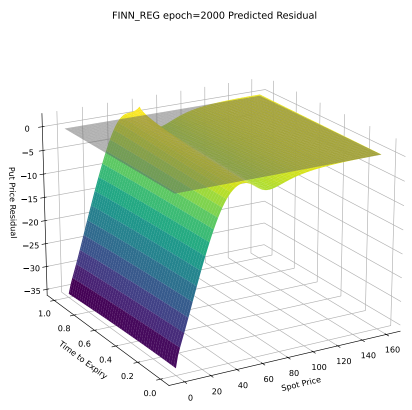
Residual	MEAN	STD	MIN	ABS MIN	MAX	ABS MAX	MSE
KAFIN	0.3027	0.3166	−1.2991	0.0000	3.6489	3.6489	0.8953
FINN REG	−11.4139	17.4739	−59.84995	0.3508	−0.3508	59.84995	435.6164
FINN FFN	−11.7586	17.5028	−59.88425	0.6513	−0.6513	59.88425	444.6143
FINN DNN	−10.8948	17.5751	−59.79089	0.00097	0.2091	59.79089	427.5819
FINN MLP	−10.6701	17.5816	−59.72084	0.00081	0.4364	59.72084	422.9636



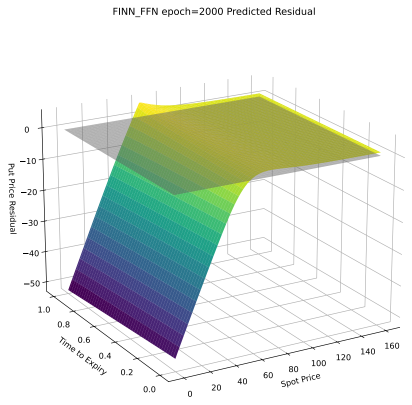
(**a**) Analytical solution



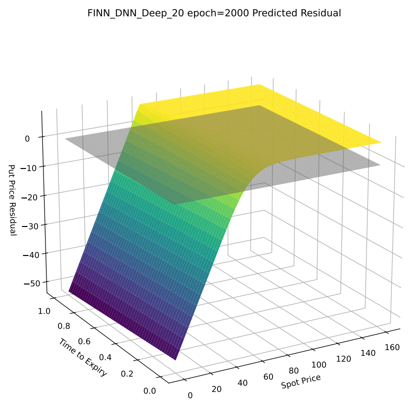
(**b**) Residual of KAFIN



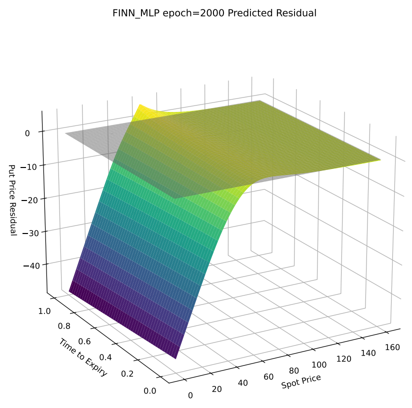
(**c**) Residual of FINN REG



(**d**) Residual of FINN FFN



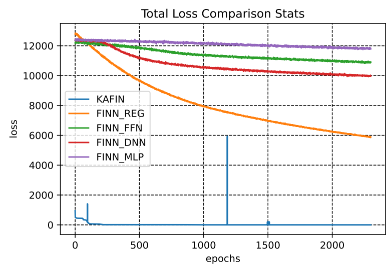
(**e**) Residual of FINN DNN



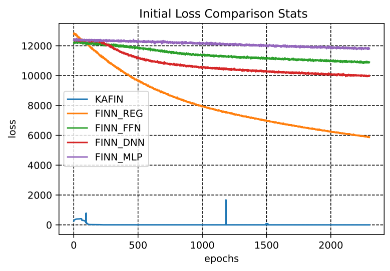
(**f**) Residual of FINN MLP

Figure 4. Overall residual on American put comparison among methods.

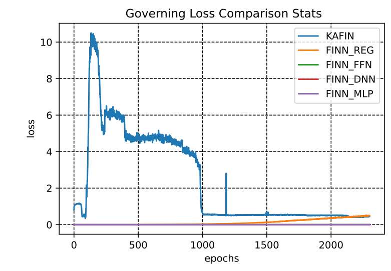
We also analyze the convergence of FINN models during the training process. All the loss function values are monitored over successive training iterations. Figures 5 and 6 show the trajectories loss functions during training. KAFIN shows rapid convergence, with the stabilizing cross all loss functions, reflecting the efficiency of the KAFIN approach in training neural networks for financial applications. Table 5 shows the quantitative stats of all the loss functions under the measures including mean, standard deviation (STD), minimum (MIN), and maximum (MAX). KAFIN obtains the least total loss among all, showing a harmonic training process with the balance among all loss principles.



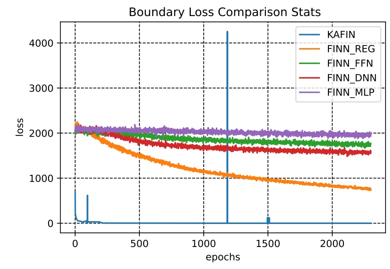
(**a**) $\mathcal{L}_{total}$ Stats



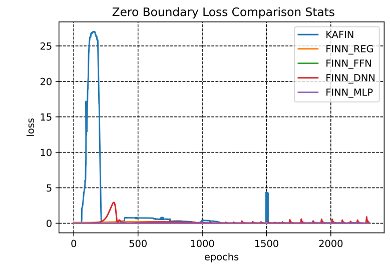
(**b**) $\mathcal{L}_{initial}$ Stats



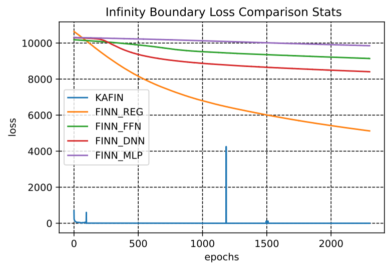
(**c**) $\mathcal{L}_{financial}$ Stats



(**d**) $\mathcal{L}_{boundary}$ Stats

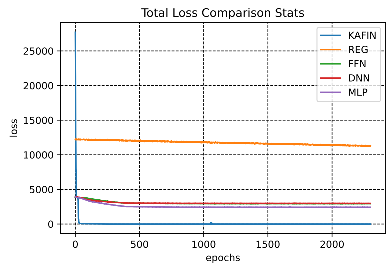


(**e**) $\mathcal{L}_{boundary,0}$ Stats

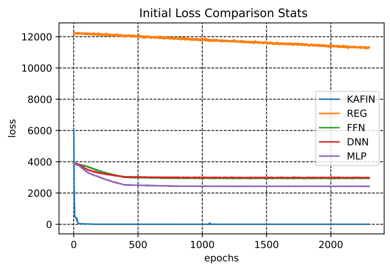


(**f**) $\mathcal{L}_{boundary,\infty}$ Stats

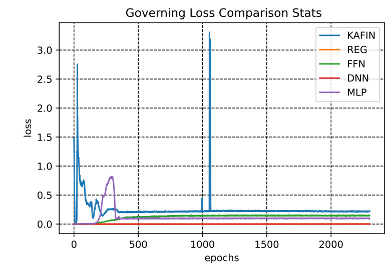
Figure 5. Overall comparison of loss among FINNs on European option pricing.



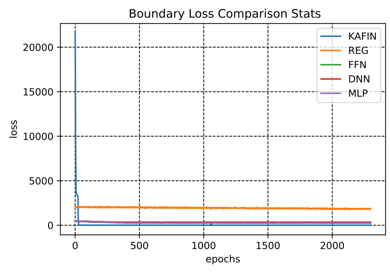
(**a**) $\mathcal{L}_{total}$ Stats



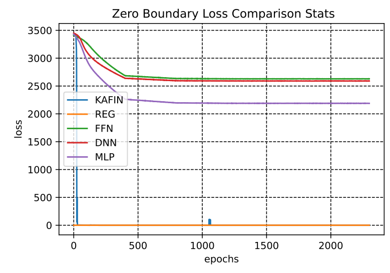
(**b**) $\mathcal{L}_{initial}$ Stats



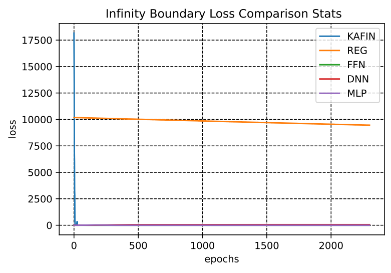
(**c**) $\mathcal{L}_{financial}$ Stats



(**d**) $\mathcal{L}_{boundary}$ Stats



(**e**) $\mathcal{L}_{boundary,0}$ Stats



(**f**) $\mathcal{L}_{boundary,\infty}$ Stats

Figure 6. Overall comparison of loss among FINNs on American put option pricing.

It can be seen that the KAFIN model outperforms conventional approaches in terms of accuracy and convergence speed. As a comparison, KAFIN needs fewer epochs for training with less complexity.

Table 5. Loss values in European call option pricing with 2300 epoch training.

MEAN	$\mathcal{L}_{total}$	$\mathcal{L}_{initial}$	$\mathcal{L}_{financial}$	$\mathcal{L}_{boundary}$	$\mathcal{L}_{boundary,0}$	$\mathcal{L}_{boundary,\infty}$
KAFIN	28.8352	18.6403	2.3630	7.8319	1.3675	6.4644
FINN REG	8156.2915	8156.1657	0.1257	1203.5108	0.0915	6952.5634
FINN FFN	11,421.6493	11,421.6481	0.0012	1861.5178	0.0004	9560.1299
FINN DNN	10,732.5727	10,732.5727	0.0000	1722.6049	0.1503	9009.8176
FINN MLP	12,110.2974	12,110.2974	0.0000	2020.4457	0.0017	10,089.8499
STD	$\mathcal{L}_{total}$	$\mathcal{L}_{initial}$	$\mathcal{L}_{financial}$	$\mathcal{L}_{boundary}$	$\mathcal{L}_{boundary,0}$	$\mathcal{L}_{boundary,\infty}$
KAFIN	151.5633	79.9292	2.4999	91.8692	5.0428	91.6668
FINN REG	1868.4424	1868.5568	0.1521	378.4857	0.0621	1490.7575
FINN FFN	410.3859	410.3870	0.0015	96.8601	0.0010	317.0152
FINN DNN	725.1539	725.1539	0.0001	157.5038	0.3906	569.4847
FINN MLP	180.6481	180.6481	0.0000	51.5401	0.0011	136.8783
MIN	$\mathcal{L}_{total}$	$\mathcal{L}_{initial}$	$\mathcal{L}_{financial}$	$\mathcal{L}_{boundary}$	$\mathcal{L}_{boundary,0}$	$\mathcal{L}_{boundary,\infty}$
KAFIN	0.9411	0.4308	0.3437	0.0558	0.0066	0.0485
FINN REG	5849.0020	5848.5151	0.0003	727.2389	0.0287	5121.2471
FINN FFN	10,825.3271	10,825.3223	0.0000	1670.5862	0.0000	9140.9287
FINN DNN	9933.3311	9933.3311	0.0000	1481.9844	0.0001	8407.6611
FINN MLP	11,744.3564	11,744.3564	0.0000	1879.0813	0.0002	9850.9521
MAX	$\mathcal{L}_{total}$	$\mathcal{L}_{initial}$	$\mathcal{L}_{financial}$	$\mathcal{L}_{boundary}$	$\mathcal{L}_{boundary,0}$	$\mathcal{L}_{boundary,\infty}$
KAFIN	5934.0322	1675.9518	10.5056	4255.2658	27.0373	4255.2524
FINN REG	12,850.6650	12,850.6641	0.5156	2249.6992	0.2395	10,644.5205
FINN FFN	12,270.0977	12,270.0977	0.0055	2103.2632	0.0096	10,183.2969
FINN DNN	12,458.5586	12,458.5586	0.0007	2175.2749	2.9416	10,298.2354
FINN MLP	12,493.1035	12,493.1035	0.0000	2188.5798	0.0034	10,312.2305

7. Discussion

7.1. Discussion on European Option Pricing

7.1.1. Performance

In this section, we present the results of our KAFIN-based approach for European option pricing, emphasizing its effectiveness in comparison to traditional baselines. As depicted in Figure 1, KAFIN's predictions demonstrate a close alignment with the analytical solutions, which serve as the benchmark for evaluating model accuracy. This strong performance suggests that KAFIN is proficient in capturing the underlying dynamics of option pricing, which is a task known for its inherent complexity.

Further comparative analysis is provided in Table 3, which summarizes the residual statistics between the analytical solution and the predicted solutions across various models. Notably, KAFIN exhibits the least residual values across all metrics, including mean, standard deviation (STD), minimum, absolute minimum, maximum, absolute maximum, and mean squared error (MSE). Specifically, the mean residual of KAFIN is -0.2532 with an STD of 0.9578, indicating that its predictions are not only close to the benchmark but also demonstrate minimal variability. This stability is particularly crucial in financial applications, where precision can significantly influence decision-making.

In contrast, the baseline models, including FINN REG, FINN FFN, FINN DNN, and FINN MLP, show substantially larger residuals. For instance, FINN REG has a mean residual of -16.2222, with an STD of 23.7148, highlighting a considerable deviation from the analytical solution. Such discrepancies suggest that these models either struggle to generalize from the training data or fail to adequately capture the nuances of option pricing dynamics.

The visualization in Figure 3 reinforces these findings, illustrating the residuals of KAFIN bounded within the range of $[-3, +3]$. This range indicates a high degree of accuracy and consistency in KAFIN's predictions. Conversely, the other methods exhibit broader and more erratic residual distributions, suggesting that they require further training to approach the performance level of KAFIN.

The performance of KAFIN can be attributed to several factors. Its architecture is likely optimized for learning the complex relationships inherent in financial data, and its training methodology effectively minimizes loss across diverse metrics. The robust integration of features, combined with advanced regularization techniques, likely contributes to KAFIN's ability to stabilize residuals and improve generalization.

The results indicate that KAFIN significantly outperforms its baseline counterparts in the task of European option pricing. Its ability to minimize residuals and maintain a close approximation to the analytical solutions highlights its potential as a reliable tool in financial modeling. Future work may focus on further refining KAFIN's architecture and exploring its applicability across different financial derivatives to enhance its utility in real-world scenarios.

7.1.2. Convergence

The convergence behavior of Finance-Informed Neural Networks (FINNs) during the training process is critical for understanding their efficacy in financial applications. As illustrated in Figure 5, the trajectories of the loss functions reveal distinct patterns across different models. Notably, the KAFIN model exhibits rapid convergence, with a stabilizing trend across all monitored loss functions. This rapid stabilization reflects the KAFIN approach's efficiency in integrating multi-modal data and advanced training techniques, contributing to its superior performance in financial contexts.

The quantitative statistics presented in Table 5 provide further insights into the performance of the models. KAFIN achieves the lowest mean total loss of 28.8352, significantly outperforming other models such as FINN REG, FINN FFN, FINN DNN, and FINN MLP, which show mean losses exceeding 8000. The low mean loss indicates that KAFIN effectively balances various loss principles, resulting in a more harmonious training process. This balance is essential for addressing the complexities inherent in financial data, where capturing intricate relationships and dependencies is paramount.

The standard deviation (STD) values further elucidate the stability of KAFIN during training. With an STD of 151.5633 for total loss, KAFIN demonstrates a consistent training trajectory, whereas other models display significantly higher standard deviations, suggesting greater volatility in their training processes. Such volatility can lead to overfitting or underfitting, making KAFIN's ability to maintain stability a noteworthy advantage.

Additionally, the minimum and maximum loss values highlight KAFIN's robustness. The model's minimum total loss of 0.9411 and maximum of 5934.0322 indicate that it effectively minimizes errors, even under challenging conditions. In contrast, models such as FINN REG and FINN FFN exhibit much higher minimum losses, suggesting they struggle more significantly to capture the underlying dynamic patterns.

The performance metrics indicate that KAFIN not only converges rapidly but also maintains a low total loss across different loss components, including financial and boundary losses. This performance can be attributed to KAFIN's architecture, which likely incorporates mechanisms that enhance the learning of critical features from financial patterns while preventing excessive variance in predictions.

The analysis of loss function trajectories and statistical metrics reveals that KAFIN outperforms other FINN models in terms of convergence speed, stability, and overall loss minimization. These characteristics are crucial for practical applications in finance, where accuracy and reliability are paramount. The insights gained from this training analysis underscore the importance of employing advanced techniques, such as KAFIN, to address the challenges posed by financial patterns, ultimately leading to better-informed decision-making.

7.2. Discussion on American Put Pricing

7.2.1. Performance

In this section, we present the performance analysis of various models in the context of American put option pricing, highlighting the superior performance of the KAFIN-based approach. As demonstrated in Table 4, KAFIN consistently outperforms other models in terms of residual statistics, showing the least error across all metrics, including mean, standard deviation (STD), minimum, absolute minimum, maximum, absolute maximum, and mean squared error (MSE). Specifically, the mean residual for KAFIN is 0.3027, with a standard deviation (STD) of 0.3166, indicating that KAFIN's predictions closely align with the expected outcomes and exhibit minimal variability.

The residuals of KAFIN show the smallest values among all the models, with an absolute minimum of 0.0000, indicating a high level of accuracy. In contrast, the other models, such as FINN REG, FINN FFN, FINN DNN, and FINN MLP, demonstrate significantly larger residuals. For instance, FINN REG has a mean residual of -11.4139 and an STD of 17.4739, indicating a much larger deviation from the ideal predictions. Similarly, the residuals for the other baseline models also show considerable discrepancies, with FINN FFN, FINN DNN, and FINN MLP exhibiting mean residuals of -11.7586, -10.8948, and -10.6701, respectively. These values suggest that these models either fail to generalize effectively from the training data or struggle to capture the complexities of American put option pricing dynamics.

Furthermore, the maximum residual for KAFIN is 3.6489, which is much smaller than the maximum residuals observed for the baseline models. For example, FINN REG has a maximum residual of 59.84995, while FINN FFN has a similar maximum value of 59.88425. These large residuals imply that these models may be overly sensitive to noise or fail to capture the nuanced relationships in the financial data.

Despite KAFIN's relatively larger standard deviation compared to some other models, its overall performance is far superior in terms of reducing the magnitude of residuals. The robust performance of KAFIN can be attributed to its architecture, which is specifically designed to capture complex relationships inherent in financial data. Additionally, KAFIN's training methodology likely optimizes the model to minimize errors across various metrics, resulting in a more stable and accurate performance in American put option pricing.

It can be seen from the results of the residuals that KAFIN stands out as the most reliable model for American put option pricing. Its ability to minimize residual errors and maintain close alignment with the expected results makes it a powerful tool in financial modeling. The superior performance of KAFIN over traditional models underscores its potential for real-world financial applications. Future work may focus on further optimizing KAFIN's architecture and exploring its applicability across a broader range of financial derivatives to enhance its utility.

7.2.2. Convergence

As illustrated in Figure 6, the KAFIN model demonstrates a comparable performance to that observed in the European call option pricing experiment, exhibiting rapid convergence in the context of American put option pricing. All monitored loss functions exhibit a stable trend, indicating consistent performance throughout the training process. This rapid stabilization underscores the effectiveness of the KAFIN approach in integrating multi-modal data and leveraging advanced training techniques. Such capabilities are crucial in enabling the model to consistently perform well in the inherently complex and dynamic financial environments. Table 6 summarizes the quantitative statistics of the loss trend, in which KAFIN outperforms all other models, particularly in terms of the minimum (MIN) loss, demonstrating its superior ability to achieve optimal solutions.

Table 6. Loss values in American put option pricing with 2300 epoch training.

MEAN	$\mathcal{L}_{total}$	$\mathcal{L}_{initial}$	$\mathcal{L}_{financial}$	$\mathcal{L}_{boundary}$	$\mathcal{L}_{boundary,0}$	$\mathcal{L}_{boundary,\infty}$
KAFIN	80.5322	15.9445	0.2464	64.3412	34.6914	29.6498
FINN REG	1615.1479	1613.6442	1.5037	146.5673	1466.3641	0.7128
FINN FFN	2969.4533	2969.4227	0.0306	316.0591	2652.2803	1.0833
FINN DNN	3092.3232	3092.3232	0.0000	347.3228	2698.7922	46.2082
FINN MLP	2519.5703	2519.1787	0.3916	252.6796	2264.6677	1.8314
STD	$\mathcal{L}_{total}$	$\mathcal{L}_{initial}$	$\mathcal{L}_{financial}$	$\mathcal{L}_{boundary}$	$\mathcal{L}_{boundary,0}$	$\mathcal{L}_{boundary,\infty}$
KAFIN	1031.4969	212.0296	0.1620	822.9333	336.1925	605.8546
FINN REG	637.2571	637.6745	0.5911	101.2585	537.3467	2.3560
FINN FFN	246.4827	246.4926	0.0104	39.5853	207.5623	0.3016
FINN DNN	180.8077	180.8077	0.0000	22.8652	173.2816	14.6223
FINN MLP	320.4845	320.6253	0.1511	48.2389	269.3868	6.4497
MAX	$\mathcal{L}_{total}$	$\mathcal{L}_{initial}$	$\mathcal{L}_{financial}$	$\mathcal{L}_{boundary}$	$\mathcal{L}_{boundary,0}$	$\mathcal{L}_{boundary,\infty}$
KAFIN	0.4742	0.2592	0.0007	0.0090	0.0042	0.0001
FINN REG	1321.4523	1319.9526	0.0083	92.8221	1226.2039	0.0000
FINN FFN	2836.8435	2836.8066	0.0001	279.0123	2555.5266	0.0001
FINN DNN	2991.8159	2991.8159	0.0000	315.1060	2618.3367	0.0000
FINN MLP	2358.1545	2357.6838	0.0000	212.8911	2142.5896	0.0000
MIN	$\mathcal{L}_{total}$	$\mathcal{L}_{initial}$	$\mathcal{L}_{financial}$	$\mathcal{L}_{boundary}$	$\mathcal{L}_{boundary,0}$	$\mathcal{L}_{boundary,\infty}$
KAFIN	27,701.9805	6116.9302	3.3060	21,583.5708	3421.9819	18,161.8652
FINN REG	3975.9348	3975.9265	4.7604	492.7719	3481.0391	12.4286
FINN FFN	3893.2429	3893.2427	0.0384	476.4237	3428.1106	2.2759
FINN DNN	3894.0923	3894.0923	0.0000	462.3209	3440.4221	53.7341
FINN MLP	3868.3335	3868.3335	0.5005	462.6045	3414.2256	34.1587

Despite having the highest values in terms of MAX and STD among the models, KAFIN exhibits the lowest MIN total loss of 0.4742, which is significantly better than the next best model, FINN REG, with a MIN total loss of 1321.4523. This stark contrast highlights KAFIN's ability to minimize loss effectively, particularly under favorable conditions, where the model achieves its best performance. The MIN value is critical as it reflects the smallest loss observed during training, serving as an indicator of how well the model performs under ideal scenarios. KAFIN's low MIN suggests that it consistently finds optimal solutions, even in challenging environments.

While MAX and STD are larger for KAFIN compared to other models, these metrics do not detract from its overall superior performance. In fact, the large MAX (27,701.9805) and STD (15.9445) values can be interpreted as a reflection of KAFIN's exploratory nature, where the model is capable of covering a wider range of the loss surface. This characteristic is beneficial in complex tasks where the model needs to escape local minima and discover more generalized patterns that other models might overlook. These larger fluctuations are indicative of KAFIN's capacity to explore different regions of the solution space, potentially identifying better overall solutions.

Furthermore, the ability of KAFIN to maintain a low MIN value indicates that it has achieved a stable and robust solution during the optimization process, even if it experiences occasional fluctuations in the loss values. The larger MAX and STD values should not overshadow the model's effectiveness in consistently achieving low-loss predictions under normal conditions, which is critical in many applications requiring high precision and accuracy.

The performance of KAFIN can be attributed to several key factors. Its architecture is well-suited to capture complex patterns in the data, and its training strategy focuses on minimizing loss across various metrics. Moreover, the flexibility of the model allows it to adapt to diverse data distributions and find optimal solutions that align closely with the underlying data structure.

Despite the occasional larger loss values as indicated by MAX and STD, KAFIN stands out as the most effective model in terms of achieving the lowest MIN total loss. This demonstrates that KAFIN excels in loss minimization, making it an ideal choice for tasks where achieving high accuracy in the best-case scenario is crucial.

7.3. Rule-Based Data

A notable aspect of this paper is that the evaluation primarily relies on generated input data, governed by financial rules, rather than real market data. This choice offers a more controlled, noise-free, and flexible environment for model testing. By using generated data, this study ensures rigorous testing under theoretical conditions, allowing the results to reflect the model's intrinsic capabilities rather than being influenced by the complexities and noise of actual market conditions. This approach improves the reliability and clarity of the evaluation, enabling comprehensive performance testing across a wide range of scenarios.

Moreover, the use of generated data based on financial rules and regulations presents a significant advantage in terms of model applicability and reliability, particularly in situations where real market data may be inaccessible. In practice, market data are often incomplete, confidential, or unavailable due to factors such as regulatory restrictions, data privacy concerns, or market disruptions. By utilizing generated data that adheres to predefined financial rules and regulatory frameworks, the model remains robust and applicable even in the absence of actual market data. This ensures that the model can be consistently evaluated and deployed across various conditions, extending its applicability in environments where data availability is limited or restricted. Consequently, the proposed method offers a more adaptable and reliable solution, enhancing its versatility and expanding its potential for addressing a wide range of financial scenarios.

7.4. Computational Resources

The computational cost analysis between KAFIN and the baseline models, as outlined in Table 7, provides significant insights into both the runtime efficiency and resource utilization associated with each method. This analysis is crucial for understanding the trade-offs between model complexity and computational feasibility, especially in real-world applications where time and memory resources are often constrained.

Table 7. Runtime and resource consumption for 2300 epochs.

Method	KAFIN	FINN REG	FINN FNN	FINN DNN	FINN MLP
Runtime (s)	312.34	1185.42	160.31	1555.49	46.23
Sec/Epoch	0.1358	0.5154	0.0697	0.6763	0.0201
Avg GPU MB	25.7246	4.4531	4.8423	10.3193	1.7456
Model Size (KB)	7	11	13	1375	19

The runtime for completing 2300 epochs across the models demonstrates the inherent efficiency differences in terms of computational demands. Specifically, KAFIN, which is based on a decomposition of the Kolmogorov–Arnold representation (KAR) approach, completes 2300 epochs in 312.34 s, resulting in an average epoch time of 0.1358 s. This is substantially faster than the other models, particularly the FINN REG, which requires 1185.42 s for the same task, or an average of 0.5154 s per epoch.

This stark contrast in epoch time can be attributed to the architectural optimizations in KAFIN, particularly the advanced training techniques employed, which allow it to process data more efficiently. The decomposition of KAR into separate attention-based components is designed to capture complex financial data relationships with fewer training iterations, ultimately leading to faster convergence. In comparison, the other baseline models, such as FINN DNN and FINN FFN, take 1555.49 and 160.31 s, respectively, to complete 2300 epochs, with corresponding average times per epoch of 0.6763 and 0.0697 s. These differences suggest that KAFIN's efficient convergence process, aided by its advanced regularization and optimization techniques, enables it to achieve high performance with significantly less computational overhead.

The computational efficiency of KAFIN can also be linked to its ability to achieve faster stabilization during training. The system's architecture allows for a streamlined training process, which requires fewer epochs to reach an optimal solution, as evidenced by its performance in minimizing residual errors during financial modeling tasks.

In terms of memory consumption, KAFIN demonstrates a moderate average GPU memory usage of 25.7246 MB, which is higher than FINN REG (4.4531 MB) and FINN FNN (4.8423 MB), but still relatively low compared to the FINN DNN (10.3193 MB) and FINN MLP (1.7456 MB). While KAFIN requires slightly more memory due to the increased architectural complexity introduced by the decomposition of KAR, its memory usage is justified by its enhanced ability to capture intricate patterns in financial data.

The decomposition of KAR in KAFIN introduces additional layers and model components that are likely to require more memory. These components involve attention mechanisms and advanced regularization terms, which contribute to a higher memory footprint. However, this increased memory usage is a trade-off for the model's ability to learn complex relationships more effectively, which in turn leads to faster convergence and better predictive accuracy, particularly for tasks such as American put option pricing.

Comparatively, the baseline models like FINN REG and FINN FNN consume less memory, with FINN REG having the lowest average GPU memory usage (4.4531 MB). However, their lower memory consumption correlates with their slower convergence and higher residual errors, as they may not be as well-equipped to handle the complexity of financial data without additional model parameters and sophisticated regularization strategies. These models may require more epochs to converge to an optimal solution, thus compensating for their lower memory demands with a higher runtime.

The model size is another important consideration in computational cost, especially for deployment in resource-constrained environments. KAFIN, with a model size of 7 KB, is relatively compact compared to FINN DNN, which has an extensive model size of 1375 KB. Despite its smaller model size, KAFIN's architecture is optimized to deliver strong performance with minimal memory usage, indicating its efficiency in terms of resource allocation without compromising on model expressiveness.

In contrast, the large model size of FINN DNN suggests that while this model may have the capacity to capture more detailed features, it incurs a higher computational cost due to its complexity. The large model size also suggests that this model may take longer to load and use more memory during inference, further contributing to its higher overall runtime.

The performance of KAFIN on an RTX 4060 GPU is an important factor to consider when discussing the computational cost, as this hardware provides a good balance between high computational power and efficient memory management. The RTX 4060's processing capabilities enable KAFIN to take full advantage of parallelized computation during training, reducing epoch time significantly compared to the other models. KAFIN's ability to rapidly converge and stabilize within fewer epochs is supported by the high processing power and memory bandwidth available on the RTX 4060, which ensures efficient handling of the model's components during training.

Although FINN MLP has the lowest runtime (46.23 s for 2300 epochs), its high model size and lower convergence performance make it less optimal compared to KAFIN, which strikes a better balance between runtime efficiency and predictive accuracy.

It can be seen that KAFIN provides a highly optimized balance of runtime efficiency, memory consumption, and model size, especially in the context of financial option pricing. Despite its slightly higher GPU memory usage compared to simpler models, KAFIN's architectural complexity, enabled by the decomposition of KAR, allows it to converge rapidly, reducing both the number of epochs and the computational cost required to achieve accurate predictions. Its relatively smaller model size further contributes to its computational efficiency, making it a preferable choice in scenarios where both performance and resource constraints are critical.

KAFIN's superior convergence speed, lower residual errors, and its efficient use of memory and GPU resources underscore its strong potential for real-world financial applications, offering an attractive alternative to more traditional, resource-heavy models.

7.5. Convexity for Loss Functions

To analyze the convexity of the loss functions and the total loss function presented (29), we examine the properties of each individual component and their interactions. Convexity is important in ensuring that optimization procedures converge to a global minimum, which is essential for training neural networks.

The financial governing loss $\mathcal{L}_{\text{financial}}$ is defined as the mean squared error of the residual $\mathcal{R}^2(S_i, t_i)$ over a set of sampled points (S_i, t_i), given by (21). If the residual function $\mathcal{R}$ is differentiable and the model is well-specified, then the squared residual is convex with respect to the parameters θ because the square of any differentiable function is convex. Therefore, $\mathcal{L}_{\text{financial}}$ is convex, provided that $\mathcal{R}(S_i, t_i)$ itself is a convex function of the model's predictions $u_\theta(S, t)$.

The initial value loss $\mathcal{L}_{\text{initial}}$ measures the difference between the neural network's output at $t = 0$ and the payoff function $\max(S_i - K, 0)$, given by (21). The function $\max(S_i - K, 0)$ is convex in S, and the square of a convex function is convex. Therefore, $\mathcal{L}_{\text{initial}}$ is convex in $u_\theta(S_i, 0)$. The overall loss will remain convex if the neural network model $u_\theta(S, t)$ is sufficiently smooth and the architecture is designed to preserve convexity, such as in the case of linear activations or simpler architectures.

The boundary loss consists of two components: one for the zero boundary condition and one for the infinity boundary condition, including zero boundary loss (24) and infinity boundary loss (25). Both of these losses are squared terms, which are convex. Therefore, the boundary loss function as a whole is convex in terms of the model's output, provided that the underlying neural network architecture does not introduce non-convexities (e.g., non-convex activation functions like ReLU or sigmoid).

The total loss function is the weighted sum of the individual losses (29). Since each of the individual loss components is convex, the weighted sum of convex functions remains convex as long as the weights of λ_{initial}, $\lambda_{\text{boundary}}$, and $\lambda_{\text{financial}}$ are non-negative.

7.6. Convexity for Varying Network Depths

The convexity of a loss function with respect to the depth of a neural network is a nuanced topic, which is influenced by the architecture and activation functions employed. For shallow networks with simple structures, the loss function is more likely to exhibit convexity, particularly if individual loss terms are convex with respect to the network outputs. In such cases, the total loss remains convex relative to the outputs, simplifying the optimization process. However, as network depth increases, additional nonlinearities introduced by activation functions such as ReLU or sigmoid significantly complicate the optimization landscape. These nonlinear interactions between layers can lead to a highly non-convex total loss function with respect to the network parameters θ. This increased complexity often results in challenges for convergence and optimization.

In deeper networks, the interplay between the network parameters across layers becomes critical in determining the overall convexity. While convexity may still hold under restrictive conditions, such as specific activation functions or constrained parameter interactions, it is generally difficult to guarantee. Regularization coefficients and architectural design, including constraints on parameter magnitudes, can play a crucial role in mitigating non-convex behavior and promoting stable convergence. Therefore, the convexity of the total loss function as a function of network depth is largely context-dependent and requires careful architectural and theoretical considerations.

7.7. Residual Outliers

The observed outliers in the residual distribution for KAFIN warrant a deeper examination, particularly in relation to the underlying dynamics of the model's mechanism-driven training process. The residuals, calculated after training the model for a finite number of steps, reflect the differences between the model's output and the theoretical values. This comparison is important, but it is crucial to recognize that our training approach is mechanism-driven rather than data-driven. In mechanism-driven training, the model is designed to align with the financial mechanisms governing the system, rather than optimizing for predefined target values. As a result, KAFIN's outputs are inherently influenced by the complex financial processes embedded within the system, which can lead to deviations from the theoretical target values in certain situations.

These residuals, therefore, reflect the system's effort to model the financial mechanism as accurately as possible, even if this means diverging from theoretical targets in specific cases. This distinction is key to understanding why the outliers appear. Unlike data-driven approaches, which prioritize minimizing the error between the model's output and the training data, mechanism-driven training prioritizes the alignment with the intrinsic financial dynamics, which may not always match the theoretical values exactly. This can naturally lead to some residuals that are larger than expected, especially when the model encounters scenarios that are particularly complex or when certain assumptions in the theoretical model may not hold perfectly in real-world data.

A more granular analysis of these outliers, particularly by segmenting them according to specific ranges of asset prices or time periods, could shed light on the underlying causes. For instance, certain asset price ranges or time periods, such as those with high volatility or rapid market changes, may lead to greater discrepancies between the model's predictions and the theoretical values. In these cases, the financial mechanism driving KAFIN could be reflecting more nuanced market behaviors that are not fully captured by the theoretical model.

The outliers in KAFIN's residual distribution are a natural consequence of the model's mechanism-driven training, which seeks to align with real-world financial processes rather than strictly adhering to theoretical values. A more detailed analysis of the residuals, considering factors such as asset price ranges and time periods, will provide further insights into the model's behavior and offer opportunities to refine its alignment with financial dynamics.

8. Conclusions and Future Works

8.1. Summary

In this study, we present KAFIN, a novel framework that integrates the Kolmogorov–Arnold representation (KAR) into Finance-Informed Neural Networks (FINNs) to enhance their application in financial modeling, specifically European option pricing. In this paper, to evaluate the performance of KAFIN, we focus on testing the model adaptability to objective laws by utilizing generated data rather than real market data. This approach facilitates controlled simulations of various financial scenarios, effectively mitigating the complexity and noise often inherent in real market data. By relying on generated data, we can isolate and assess the core functionality of KAFIN within well-defined theoretical conditions, ensuring a more precise and focused analysis of its performance.

The results demonstrate that KAFIN significantly outperforms traditional baseline methods in accuracy, as evidenced by its minimized residuals and strong alignment with analytical solutions. The capacity of KAFIN to effectively capture the complexities of option pricing can be attributed to its robust architecture, which optimally integrates financial principles with advanced computational techniques.

KAFIN's performance metrics indicate not only a lower mean total loss but also reduced variability across all monitored loss components, reflecting a stable training process. This stability is particularly crucial in financial contexts where precise and reliable predictions are paramount. The model's ability to converge rapidly and consistently across diverse training scenarios highlights its potential as a powerful tool for financial practitioners seeking accurate pricing mechanisms.

The rigorous analysis of loss function trajectories further underscores KAFIN's superiority in capturing intricate dependencies within financial patterns. This advantage stems from the framework's ability to decompose complex multivariate functions into simpler components, leveraging the Kolmogorov–Arnold representation to enhance interpretability and computational efficiency.

8.2. Reflection

The primary objective of this paper is to examine whether superior performance can be attained by utilizing the most basic version of the model, without introducing unnecessary complexity. To achieve this, we intentionally adopted a simple approach to assess the intrinsic performance characteristics of the KAFIN framework, configured with its original settings. By refraining from adding modifications or additional features, our intention was to evaluate the model's core functionality under minimal conditions, allowing us to focus on its fundamental capabilities.

It is important to acknowledge that certain methods, such as adaptive activation functions (e.g., refs. [32–34]), have the potential to enhance the model's expressiveness. These functions allow the network to adjust its activation behavior across various levels and tasks, thereby enabling the model to learn complex data patterns more effectively. Such adaptive mechanisms address the limitations associated with fixed activation functions. However, the introduction of adaptive activation functions introduces a trade-off, as they increase computational complexity. The process of learning and calculating additional parameters results in higher computational overhead, which may complicate the training process. Furthermore, the increase in the number of parameters may necessitate the adoption of more advanced optimization strategies, ultimately raising the overall design complexity of the model.

This paper primarily explores the feasibility and performance of machine learning approaches for modeling financial mechanisms within a mechanism-driven context. The central aim of this study is to assess the general performance characteristics of the KAFIN framework under standard conditions, with global residuals serving as an appropriate measure of the model's accuracy across the entire input space. The Black–Scholes framework, which underpins this model, assumes that asset prices evolve smoothly and continuously under certain conditions. This assumption suggests that global error analysis is a suitable starting point for evaluating the model's performance. Such an approach is aligned with the objectives of this paper, which seeks to establish a foundational understanding of the model's general capabilities.

In addition, we combined established financial frameworks with machine learning techniques to approximate the underlying financial mechanisms by learning the laws, properties, and parameters of the system. While the KAFIN framework is grounded in established financial principles, it operates within a machine learning context where deriving precise theoretical bounds for prediction errors is inherently challenging. This difficulty is particularly pronounced for complex models that integrate mechanism-driven components, such as neural networks. Nonetheless, the empirical performance demonstrated by KAFIN provides strong evidence of its effectiveness, with its residuals consistently lower than those observed in baseline models. These results highlight the model's capacity to approximate the financial mechanism effectively, despite the inherent challenges in deriving theoretical guarantees.

The integration of error optimization with the Black–Scholes equations in the KAFIN framework introduces a form of regularization and constraint that enhances the model's robustness. This hybrid approach, which blends financial principles with data-driven methodologies, ensures that the model remains closely aligned with the underlying mathematical structures while adapting to the observed data. This synergy between financial laws and data-driven learning contributes to the model's capacity to generate reliable forecasts.

8.3. Future Works

Looking forward, we plan to investigate the theoretical bounds of the KAFIN model in greater depth. This will involve the development of more specific error analysis techniques, as well as the refinement of methods for evaluating the model's performance under boundary conditions or extreme market scenarios. Additionally, we will continue empirical testing to assess the robustness of KAFIN in more challenging financial environments and datasets. This extended evaluation will be crucial for ensuring that the model remains effective across a broader range of real-world financial conditions.

The current framework provides a foundation for future research. By combining the mathematical rigor inherent in financial mechanisms with empirical validation via residual analysis, we can offer a robust assessment of the model's capabilities. Moreover, we fully recognize the value of local residual analysis, particularly in situations involving boundary conditions or extreme asset prices. This form of analysis will be considered in future work, as it can offer valuable insights, especially in complex scenarios where boundary behavior plays a critical role. Additionally, the incorporation of theoretical error bounds will help to provide stronger guarantees regarding the model's robustness.

Ultimately, future work will seek to improve upon the current analysis by providing further theoretical insights and extending the evaluation to more diverse and complex scenarios. This will ensure that the robustness of the KAFIN framework is thoroughly understood and validated in a wide range of financial contexts.

In addition to the core model evaluation, our framework has potential applications in areas such as stochastic volatility and portfolio strategies. For instance, in the context of a straddle portfolio, the model can predict the future volatility of the underlying asset and subsequently adjust the option strike prices and quantities in accordance with the predicted volatility range. In this scenario, the maximum loss of the straddle portfolio is confined to the total premium paid for the two options. Our system analyzes historical market volatility and adapts the model based on actual market conditions to ensure risk control. This enables the mitigation of potential risks and the optimization of options portfolios in alignment with trading strategy. We intend to present further developments and results related to these applications in future work.

Looking ahead, future research will focus on extending KAFIN's capabilities to encompass non-convex learning scenarios and American option pricing. These directions will not only enhance the framework's applicability but also contribute to a deeper understanding of its potential in addressing a broader array of financial derivatives. KAFIN represents an advancement in the application of providing a robust, efficient, and interpretable framework that can adapt to the evolving complexities of the financial landscape.

Author Contributions: Conceptualization, C.Z.L., Y.Z., L.Q. and Y.L.; Methodology, C.Z.L. and Y.Z.; Software, C.Z.L.; Validation, C.Z.L.; Formal analysis, C.Z.L.; Investigation, C.Z.L., Y.Z. and L.Q.; Resources, C.Z.L., Y.Z. and L.Q.; Writing—original draft, C.Z.L. and Y.L.; Writing—review & editing, C.Z.L. and Y.L.; Visualization, C.Z.L.; Supervision, Y.Z. and L.Q.; Project administration, C.Z.L., Y.Z. and L.Q.; Funding acquisition, Y.Z. and L.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: Authors Charles Z. Liu and Yongfei Liu were employed by the company EulerAI. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Hainaut, D.; Casas, A. Option pricing in the Heston model with Physics inspired neural networks. *Ann. Financ.* **2024**, *20*, 353–376. [CrossRef]
2. Santos, D.d.S.; Ferreira, T.A.E. Neural Network Learning of Black-Scholes Equation for Option Pricing. *arXiv* **2024**, arXiv:2405.05780.
3. Wang, X.; Li, J.; Li, J. A deep learning based numerical PDE method for option pricing. *Comput. Econ.* **2023**, *62*, 149–164. [CrossRef]
4. Dhiman, A.; Hu, Y. Physics Informed Neural Network for Option Pricing. *arXiv* **2023**, arXiv:2312.06711.
5. Gatta, F.; Di Cola, V.S.; Giampaolo, F.; Piccialli, F.; Cuomo, S. Meshless methods for American option pricing through physics-informed neural networks. *Eng. Anal. Bound. Elem.* **2023**, *151*, 68–82. [CrossRef]
6. Ibrahim, A.Q.; Götschel, S.; Ruprecht, D. Parareal with a physics-informed neural network as coarse propagator. In Proceedings of the European Conference on Parallel Processing, Limassol, Cyprus, 28 August 2023; Springer: Berlin/Heidelberg, Germany, 2023; pp. 649–663.
7. Bai, Y.; Chaolu, T.; Bilige, S. The application of improved physics-informed neural network (IPINN) method in finance. *Nonlinear Dyn.* **2022**, *107*, 3655–3667. [CrossRef]
8. Black, F.; Scholes, M. The pricing of options and corporate liabilities. *J. Political Econ.* **1973**, *81*, 637–654. [CrossRef]
9. Butler, J.S.; Schachter, B. Unbiased estimation of the Black/Scholes formula. *J. Financ. Econ.* **1986**, *15*, 341–357. [CrossRef]
10. MacBeth, J.D.; Merville, L.J. An empirical examination of the Black-Scholes call option pricing model. *J. Financ.* **1979**, *34*, 1173–1186. [CrossRef]
11. Hainaut, D. Valuation of guaranteed minimum accumulation benefits (GMABs) with physics-inspired neural networks. *Ann. Actuar. Sci.* **2024**, *18*, 442–473. [CrossRef]
12. Villarino, J.P.; Leitao, Á.; Rodríguez, J.G. Boundary-safe PINNs extension: Application to non-linear parabolic PDEs in counterparty credit risk. *J. Comput. Appl. Math.* **2023**, *425*, 115041. [CrossRef]
13. Zhang, Q.s.; Zhu, S.C. Visual interpretability for deep learning: A survey. *Front. Inf. Technol. Electron. Eng.* **2018**, *19*, 27–39. [CrossRef]
14. Chakraborty, S.; Tomsett, R.; Raghavendra, R.; Harborne, D.; Alzantot, M.; Cerutti, F.; Srivastava, M.; Preece, A.; Julier, S.; Rao, R.M.; et al. Interpretability of deep learning models: A survey of results. In Proceedings of the 2017 IEEE Smartworld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (Smartworld/SCALCOM/UIC/ATC/CBDcom/IOP/SCI), San Francisco, CA, USA, 4–8 August 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–6.
15. Polar, A.; Poluektov, M. A deep machine learning algorithm for construction of the Kolmogorov–Arnold representation. *Eng. Appl. Artif. Intell.* **2021**, *99*, 104137. [CrossRef]
16. Kodama, S. A version of kolmogorov-arnold representation theorem for differentiable functions of several variables. *J. Nonlinear Anal. Optim. Theory Appl. (JNAO)* **2011**, *2*, 253–257.
17. Liu, Z.; Wang, Y.; Vaidya, S.; Ruehle, F.; Halverson, J.; Soljačić, M.; Hou, T.Y.; Tegmark, M. Kan: Kolmogorov-arnold networks. *arXiv* **2024**, arXiv:2404.19756.
18. Schmidt-Hieber, J. The Kolmogorov—Arnold representation theorem revisited. *Neural Netw.* **2021**, *137*, 119–126. [CrossRef]
19. Vitushkin, A.G. A proof of the existence of analytic functions of several variables not representable by linear superpositions of continuously differentiable functions of fewer variables. In *Doklady Akademii Nauk*; Russian Academy of Sciences: Moscow, Russia, 1964; Volume 156, pp. 1258–1261.
20. Merton, R.C. Theory of rational option pricing. *Bell J. Econ. Manag. Sci.* **1973**, *4*, 141–183. [CrossRef]
21. Galai, D.; Masulis, R.W. The option pricing model and the risk factor of stock. *J. Financ. Econ.* **1976**, *3*, 53–81. [CrossRef]
22. Smith, C.W., Jr. Option pricing: A review. *J. Financ. Econ.* **1976**, *3*, 3–51. [CrossRef]
23. Bates, D.S. 20 testing option pricing models. *Handb. Stat.* **1996**, *14*, 567–611.
24. Bergman, Y.Z.; Grundy, B.D.; Wiener, Z. *Theory of Rational Option Pricing: II*; Rodney L. White Center for Financial Research: Philadelphia, PA, USA, 1995.
25. MacBeth, J.D.; Merville, L.J. Tests of the Black-Scholes and Cox call option valuation models. *J. Financ.* **1980**, *35*, 285–301. [CrossRef]
26. Achdou, Y.; Pironneau, O. *Computational Methods for Option Pricing*; SIAM: Philadelphia, PA, USA, 2005.
27. Lauterbach, B.; Schultz, P. Pricing warrants: An empirical study of the Black-Scholes model and its alternatives. *J. Financ.* **1990**, *45*, 1181–1209. [CrossRef]
28. Bakshi, G.; Cao, C.; Chen, Z. Empirical performance of alternative option pricing models. *J. Financ.* **1997**, *52*, 2003–2049. [CrossRef]
29. Kumar, H.; Yadav, N. Deep learning algorithms for solving differential equations: A survey. *J. Exp. Theor. Artif. Intell.* **2023**, 1–40. [CrossRef]
30. Hou, M.; Fu, H.; Hu, Z.; Wang, J.; Chen, Y.; Yang, Y. Numerical solving of generalized Black-Scholes differential equation using deep learning based on blocked residual connection. *Digit. Signal Process.* **2022**, *126*, 103498. [CrossRef]
31. Argyriou, A.; Evgeniou, T.; Pontil, M. Convex multi-task feature learning. *Mach. Learn.* **2008**, *73*, 243–272. [CrossRef]

32. Scardapane, S.; Van Vaerenbergh, S.; Totaro, S.; Uncini, A. Kafnets: Kernel-based non-parametric activation functions for neural networks. *Neural Netw.* **2019**, *110*, 19–32. [CrossRef]
33. Bingham, G.; Miikkulainen, R. Discovering parametric activation functions. *Neural Netw.* **2022**, *148*, 48–65. [CrossRef]
34. Pusztaházi, L.S.; Eigner, G.; Csiszár, O. Parametric Activation Functions for Neural Networks: A Tutorial Survey. *IEEE Access* **2024**, *12*, 168626–168644. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.