Journal of Mathematics and Modeling in Finance (JMMF)
Vol. 6, No. 2, Summer & Fall 2026
Research paper

The Application Of The Inverse Physics-Informed Neural Network In Financial Calibration Tasks

Mohamadamin Raeisi-Makiani[1], **Abdolsadeh Neisy**[2], **Ali Safdari-Vaighani**[3]

[1] Department of Mathematics, Allameh Tabataba'i University, Tehran, Iran
m_raeisi@atu.ac.ir

[2] Department of Mathematics, Allameh Tabataba'i University, Tehran, Iran
a_neisy@atu.ac.ir

[3] Department of Mathematics, Allameh Tabataba'i University, Tehran, Iran
asafdari@atu.ac.ir

Abstract:
In the calibration of a financial model, the process is an optimization task that can be viewed as an inverse problem. Solving this problem typically necessitates having an appropriate pricing function. With recent breakthroughs in machine learning, i.e., physics-informed neural networks (PINNs), a cutting-edge approach that combines artificial neural networks with fundamental physical principles, the task can be efficiently implemented by the inverse physics-informed neural network (iPINN). This paper centers on solving an inverse problem for a financial P(I)DE model by means of iPINN. Firstly, we model the bond price dynamics of catastrophe bonds (CAT bonds) and derive a partial integro-differential equation (PIDE) through a no-arbitrage strategy within the framework of an incomplete market. Thereafter, we employ the iPINN to estimate a specific parameter of the model, namely the market price of risk. The market price of risk is treated as a global learnable parameter and is embedded directly into the PIDE operator. The proposed iPINN is evaluated through three practically meaningful criteria: repricing error, parameter stability, and PIDE residual consistency. The outcomes demonstrate that iPINNs have the capability to solve the inverse problem effectively, and this technique could be applied broadly to real-world data.

Keywords: Catastrophe bond; Deep learning; Inverse problem; Jump diffusion model; Physics-informed neural network.
Classification: 13D45, 39B42, 34A36

1 Introduction

In recent years, deep learning has emerged as a transformative tool in financial modeling, significantly impacting the approach to solving partial differential equations (PDEs) for pricing bonds and tackling calibration tasks. Deep learning, with its capacity for approximation, pattern recognition, and handling large datasets, offers a robust and efficient alternative. By leveraging neural networks, researchers can develop models that approximate solutions to PDEs and the calibration function with greater accuracy and reduced computation time. This advancement enhances the

[1] Corresponding author
Received: 22/09/2025 Accepted: 05/05/2026
https://doi.org/10.22054/jmmf.2026.88390.1217

performance and scalability of financial models, facilitating real-time and dynamic calibration in rapidly changing markets.

Calibration, the process of fine-tuning model parameters to align with observed market data, is a critical step in ensuring the accuracy and reliability of financial models, such as those used for pricing derivatives or risk management. Traditional calibration methods often struggle with nonlinear relationships, leading to substantial computational challenges. The ability of deep learning to manage large volumes of data provides a significant advantage in addressing these issues. Although different approaches are presented in [2, 4, 8, 9], the task of calibrating models with precision is notoriously challenging, yet it holds immense significance. In parallel, numerical approximation techniques for option pricing under regime-switching and multi-state financial models have also attracted considerable attention due to their ability to capture more realistic market dynamics [13].Classical numerical approaches for nonlinear option pricing problems in illiquid markets have been extensively investigated in the literature[3],although they do not address inverse calibration problems within a data-driven learning framework.

Meshfree approximation techniques have also been widely explored in computational finance for pricing complex derivative products. In particular, radial basis function (RBF) approximation methods have been successfully employed in [1]. Such studies highlight the effectiveness of advanced approximation schemes in financial engineering, although they remain primarily focused on forward pricing problems rather than inverse calibration tasks. RBF methods have been employed for inverse problems in [7]. M. Raissi et al. [10] introduced physics-informed neural networks (PINNs) to solve two categories of problems: data-driven solutions and data-driven discovery of PDEs. The former is concerned with the solution of a PDE, whereas the latter deals with an inverse problem that can be perceived as a calibration task. This advancement not only enhances the performance of financial models but also opens new avenues for real-time and dynamic calibration in rapidly changing markets. Here, we utilize an inverse PINN (iPINN) to estimate the market price of risk from our P(I)DE model.

From a modeling perspective, catastrophe (CAT) bonds lead naturally to nonlocal pricing equations due to the presence of jump risks and loss indices, and the corresponding markets are incomplete. To the best of our knowledge, this is the first study that applies an inverse PINN framework to calibrate the market price of risk in a CAT bond pricing model governed by a jump–diffusion PIDE in an incomplete market.

The organization of this paper is structured as follows. In the next section, the mathematical models used to price the CAT bond are presented, and we obtain a PIDE. In Section 3, the PINN is formulated as an approach pertinent to our P(I)DE, offering an explanation of its methodological framework and underlying mechanisms. The implementation of iPINN to solve an inverse P(I)DE problem, the structure of our network, the evaluation metrics, and the details of the loss function

are presented in Section 4. Finally, we discuss the application of this technique in the conclusion section.

2 CAT bond model

The CAT bond valuation depends on the two spatial variables l and r. The variable l is related to the PCS index known as a loss ratio index, which is the underlying instrument for catastrophe insurance, and r is the LIBOR interest rate. Define a probability space (Ω, Γ, P) with Ω as the set of world states, Γ as a σ-algebra of subsets of Ω, and P as a probability measure on Γ. Processes are established within this probability space. The dynamics of the interest rate are modeled as

$$dr = \mu dt + \sigma_r dW(t)_r + (J-1)r\, dq = dr_{Sd} + dr_{Jump},$$

where $\{W_t : t \in [0,T]\}$ is a standard Wiener process initialized at zero, μ is the drift rate, σ_r is the volatility of the Wiener part of the process, and $(J-1)r\, dq$ illustrates the jump diffusion term. The process dq is a Poisson process and we have

$$dq = \begin{cases} 1, & \text{with probability } \gamma\, dt, \\ 0, & \text{with probability } 1-\gamma\, dt. \end{cases}$$

Also, the loss rate l follows a Cox–Ingersoll–Ross (CIR) process, which is expressed as

$$\mathrm{d}l = \beta(\alpha - l)dt + \sigma_l \sqrt{l}\, dW(t)_l, \tag{1}$$

where β is the mean-reverting force measurement, $\alpha > 0$ is the long-run mean of the loss index, σ_l is the volatility parameter, and $W(t)_l$ is a standard Wiener process such that $dW_r\, dW_l = \rho\, dt$.
Now, let $B(t,l,r)$ be the value of the CAT bond at time t with interest rate r and loss l. Due to the dynamics of r, we can consider the change in B in the following form:

$$\mathrm{d}B = \mathrm{d}B_{total} = \mathrm{d}B_{Sd} + \mathrm{d}B_{Jump}, \tag{2}$$

where $\mathrm{d}B_{Jump} = [B(t,l,Jr) - B(t,l,r)]\, \mathrm{d}q$ and we have

$$\begin{aligned} \mathrm{d}B_{Sd} = \Bigg\{ & \frac{\partial B}{\partial t} + \mu \frac{\partial B}{\partial r} + \beta(\alpha - l)\frac{\partial B}{\partial l} + \rho\sigma_r\sigma_l\sqrt{l}\frac{\partial^2 B}{\partial l \partial r} + \frac{1}{2}\sigma_l^2 l \frac{\partial^2 B}{\partial l^2} \\ & + \frac{1}{2}\sigma_r^2 \frac{\partial^2 B}{\partial r^2} \Bigg\} dt + \sigma_r \frac{\partial B}{\partial r} dW_r + \sigma_l \sqrt{l} \frac{\partial B}{\partial l} dW_l. \end{aligned} \tag{3}$$

Now, the selection of our portfolio is aimed at establishing a self-financing structure, with a specific focus on hedging a short position in bonds and engaging in trading the underlying variable (loss ratio). So, the portfolio Π is

$$\Pi = l - \Delta_1 B_1(t,l,r) - \Delta_2 B_2(t,l,r). \tag{4}$$

Hence, the change in its value is expressed as

$$\mathrm{d}\Pi = \mathrm{d}l - \Delta_1 \mathrm{d}B_1 - \Delta_2 \mathrm{d}B_2. \tag{5}$$

By using the common strategy in financial mathematics, i.e., a no-arbitrage strategy within the framework of an incomplete market and assuming the existence of an equivalent martingale measure with a market price of risk parameter λ, we obtain the following PIDE:

$$\begin{aligned} &\frac{\partial B}{\partial t} + \rho\sigma_r\sigma_l\sqrt{l}\frac{\partial^2 B}{\partial l \partial r} + \frac{1}{2}\sigma_l^2 l\frac{\partial^2 B}{\partial l^2} + \frac{1}{2}\sigma_r^2\frac{\partial^2 B}{\partial r^2} + rl\frac{\partial B}{\partial l} \\ &\qquad + (\gamma - r)B + \gamma\kappa - \lambda\frac{\partial B}{\partial r} = 0, \end{aligned} \tag{6}$$

where $\kappa = E[B(t,l,Jr)] = \int_0^\infty B(t,l,Jr)g(J)\,dJ$ such that $\int_{-\infty}^{\infty} g(J)\,dJ = 1$. A comprehensive discussion is presented in [12]. Here, we suppose that J follows a log-normal distribution, so that

$$g(J) = \frac{1}{J\sigma_J\sqrt{2\pi}}\exp\left(-\frac{1}{2}\frac{(\ln J - \mu_J)^2}{\sigma_J^2}\right).$$

From the standpoint of solving the presented PIDE, numerical PDE approaches require the solution to be obtained backwards in time. Therefore, letting τ be the time to expiry T of the contract, i.e., $\tau = T - t$, $0 \le t \le T$, and after changing variables

$$r = e^z \quad \text{and} \quad J = e^y,$$

we obtain

$$\begin{aligned} &B_\tau + e^{-z}\rho\sigma_r\sigma_l\sqrt{l}B_{lz} + \frac{1}{2}\sigma_l^2 l B_{ll} + e^{-2z}\frac{1}{2}\sigma_r^2(B_{zz} - B_z) + e^z l B_l \\ &\qquad - e^{-z}\lambda B_z + (\gamma - e^z)B + \gamma\kappa = 0, \end{aligned} \tag{7}$$

where $\kappa = E[B(t,l,Jr)] = \int_0^\infty B(t,l,e^{z+y})g(e^y)e^y dy$. So, the initial and boundary conditions can be written as

$$\begin{aligned} &B(\tau,l,e^z) \to 0 \quad \text{as } z \to \infty\ \forall l \quad \text{or as } l \to \infty\ \forall z, \\ &B(0,l,e^z) = 1. \end{aligned}$$

It should be noted that we use Gauss–Laguerre quadrature to calculate κ. For simplicity, we write equation (7) in the following form:

$$\begin{aligned} &f\big(X; B_\tau, B_l, B_z, B_{ll}, B_{zz}, B_{lz}; \theta\big) \\ &\qquad = B_\tau + e^{-z}\rho\sigma_r\sigma_l\sqrt{l}B_{lz} + \frac{1}{2}\sigma_l^2 l B_{ll} + e^{-2z}\frac{1}{2}\sigma_r^2(B_{zz} - B_z) \\ &\qquad\quad + e^z l B_l - e^{-z}\lambda B_z + (\gamma - e^z)B + \gamma\kappa = 0, \end{aligned} \tag{8}$$

where $X = (\tau, l, z)$, $\Omega \subset \mathbb{R}^3$, and unknown parameters $\theta = (\rho, \sigma_r, \sigma_l, \gamma, \lambda)$. In the next section, we introduce the inverse PINN in the context of our model.

3 Inverse Physics-Informed Neural Networks

To explain inverse physics-informed neural networks (iPINNs), suppose we have a PDE of the general form

$$f\big(X; B_\tau, B_l, B_z, B_{ll}, B_{zz}, B_{lz}; \theta\big) = 0, \quad X \in \Omega,$$

where $X = (\tau, l, z)$, $\Omega \subset \mathbb{R}^3$, and $\theta = (\rho, \sigma_r, \sigma_l, \gamma, \lambda)$ is the vector of unknown parameters.

Definition 3.1. By having Figure 1 in mind, let $d \in \mathbb{N}$ be the dimension of each input, $l \in \mathbb{N}$ be the number of layers, and $n_0, \ldots, n_l$ be the number of neurons in each layer where $n_0 = d$. Every neuron has a value which is the sum of the previous neurons with a mapping $\sigma : \mathbb{R} \to \mathbb{R}$ on it, called the activation function. Let $W_j \in \mathbb{R}^{n_j \times n_{j-1}}$ and b_j be the weights and biases of a neural network that link every neuron to each other, where $j = 1, \ldots, l$. Then a neural network is defined by

$$\mathcal{NN} : \mathbb{R}^d \longrightarrow \mathbb{R}^{n_l}, \quad x \longmapsto \delta_l \circ \sigma \circ \delta_{l-1} \circ \cdots \circ \sigma \circ \delta_1(x),$$

where for every $1 \leq j \leq l$, the function from $\mathbb{R}^{n_{j-1}}$ to $\mathbb{R}^{n_j}$ is represented by $\delta_j(x) = W_j x + b_j$ and σ is applied component-wise on the outputs of δ_j. This is the original ANN, and the right part of Figure 1, determined with dashed lines, is added to define the physics-informed neural network.

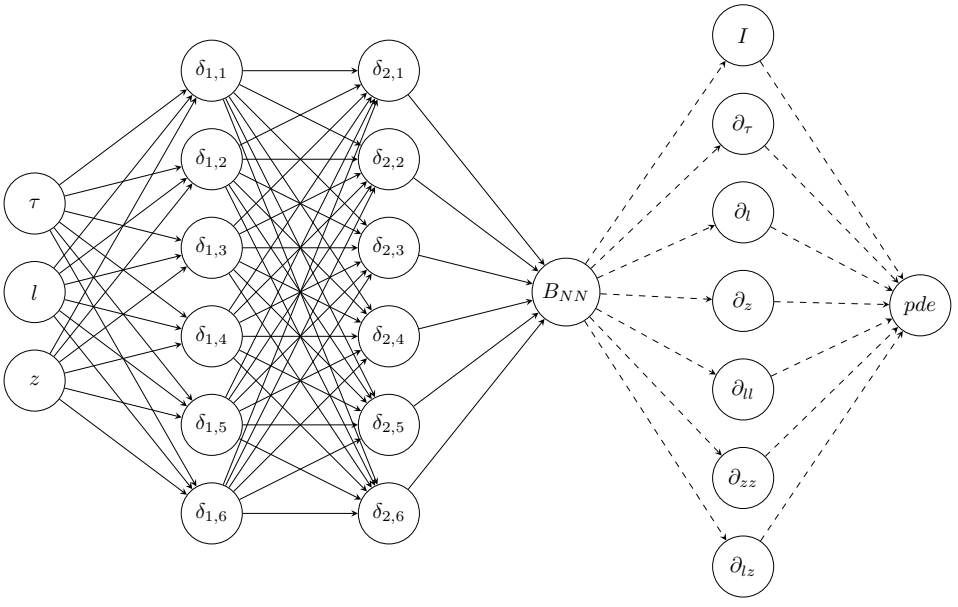


Figure 1: Diagrammatic Representation of Physics-Informed Neural Network. The left part illustrates the feed-forward neural network and the right part represents the physics-informed neural network. The dashed lines denote non-trainable weights.

The above definition implies that the loss function of iPINN is given by

$$\mathcal{L} = \omega_0 \mathcal{L}_0 + \omega_{pde} \mathcal{L}_{pde} + \omega_d \mathcal{L}_d,$$

where $\mathcal{L}_0$, $\mathcal{L}_d$, and $\mathcal{L}_{pde}$ represent the error loss from the initial condition, the data (observation) loss, and the PDE residual loss, respectively. We use mean square error (MSE) for all error terms. The main difference between PINN and iPINN is that in iPINN, $\mathcal{L}_d$ replaces the boundary loss term $\mathcal{L}_b$ and the parameters $\theta = (\rho, \sigma_r, \sigma_l, \gamma, \lambda)$ are treated as learnable variables within the network.

4 Solving inverse P(I)DE problem by iPINN

To employ iPINN, we must first construct an ANN. Selecting the number of neurons and layers along with the activation function is challenging. We practically built the ANN by choosing different numbers of neurons and layers, as well as evaluating the optimizer. Therefore, the structure of our feedforward neural network is provided in Table 1.

Table 1: Details of the ANN

Component	**Selection**
Input	3
Hidden Layers	4
Neurons per Layer	100
Activation	Tanh
Optimizer	Adam
Output	1

After generating 800 simulation data points with solution B labels, we split them into training and testing data. The test data set comprises 10% of the total data, and the range of inputs is presented in Table 2.

Table 2: The Range of Simulation Data

ANN	**Variables**	**Range**
Input	Time to Maturity, τ	$[0.05, 3]$ (Year)
	Risk-Free Rate, z	$[0.05, 0.5]$
	Loss Ratio, l	$[0, 30]$
Data	B	$[1, 5]$ billion

Hence, the total loss function to train the iPINN is expressed as

$$\begin{aligned}\mathcal{L} =&\omega_0\mathcal{L}_0 + \omega_d\mathcal{L}_d + \omega_{pde}\mathcal{L}_{pde}\\ =&\frac{1}{N_0}\sum_{i=1}^{N_0}\Big(B_{nn}(0, z_0^i, l_0^i;\Theta) - B(0, l_0^i, r_0^i)\Big)^2\\ &+\frac{1}{N_d}\sum_{i=1}^{N_d}\Big(B_{nn}(\tau_d^i, z_d^i, l_d^i;\Theta) - B(t_\tau^i, z_d^i, l_d^i)\Big)^2\\ &+\frac{1}{N}\sum_{i=1}^{N}\Big(PDE_{nn}(\tau^i, r^i, l^i, \sigma_r^i, \sigma_l^i, \rho^i, \gamma^i, \lambda^i, \Theta)\Big)^2.\end{aligned}$$

Evaluation metrics

Since the true market price of risk is not observable in practice, the performance of the inverse calibration is evaluated using three indirect but practically meaningful error measures. First, a repricing error is computed by comparing the network-predicted bond prices with the (simulated) market prices:

$$\varepsilon_{\text{price}} = \frac{1}{N_d}\sum_{i=1}^{N_d}\left|B_{nn}(X_i;\lambda_{\text{est}}) - B_i^{\text{mkt}}\right|^2. \tag{9}$$

Second, the parameter stability is assessed by training the iPINN K times with different random initializations and computing the empirical standard deviation of the estimated λ:

$$\text{Std}\big(\{\lambda_{\text{est}}^{(k)}\}_{k=1}^K\big). \tag{10}$$

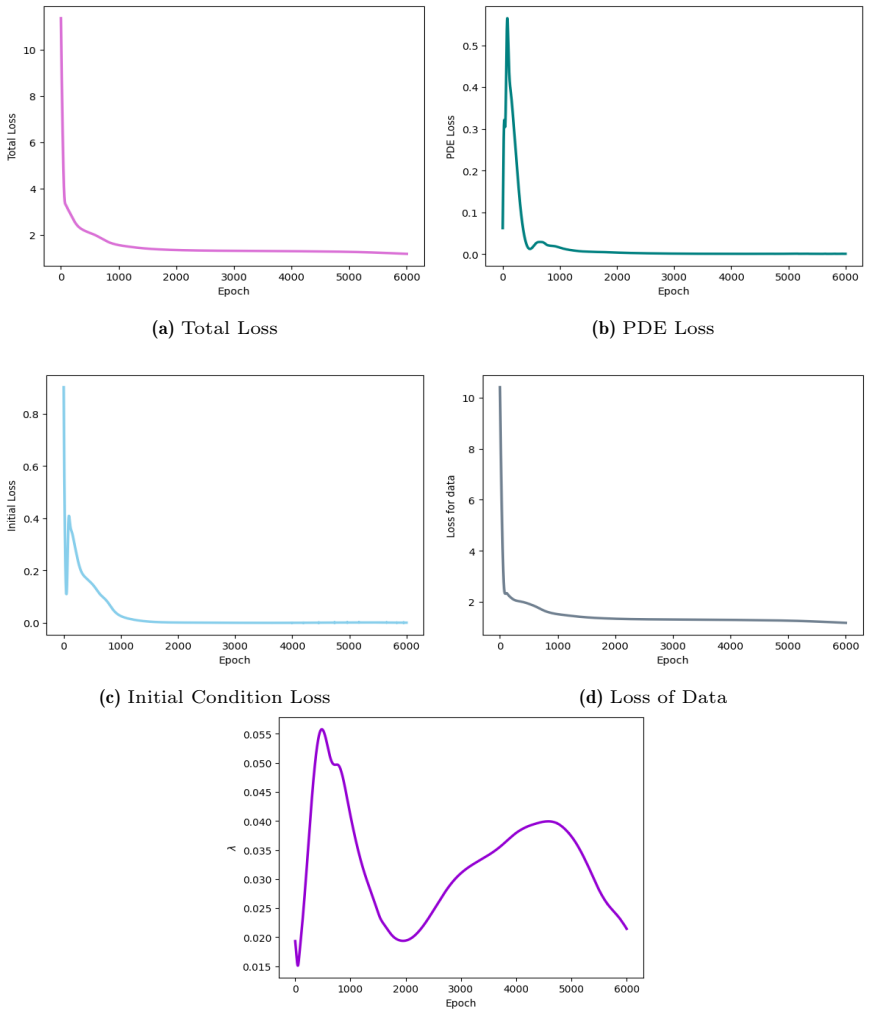
Finally, the physical consistency of the learned solution is measured by the mean squared residual of the governing P(I)DE:

$$\varepsilon_{\text{PDE}} = \frac{1}{N}\sum_{i=1}^{N}\left|f(X_i;\lambda_{\text{est}})\right|^2. \tag{11}$$

Numerical calibration results

The market price of risk can be approximated for various parameter sets. Whenever the P(I)DE is provided with fixed parameters, the associated market price of risk can be estimated in relation to those parameters by this technique. Here, the estimation is performed on two separate sets of the model's parameters. The following diagrams illustrate the loss terms related to them during training.

In the estimation corresponding to Figure 2, we obtain a total loss of order 5×10^{-4} with a final value of $\lambda \approx 0.0204$, while in Figure 3 the total loss is of order 6×10^{-4} with a final value of $\lambda \approx -0.0153$. When σ_l increases and σ_r decreases, the value of λ becomes negative due to the higher level of risk that is produced in the market.



(a) Total Loss

(b) PDE Loss

(c) Initial Condition Loss

(d) Loss of Data

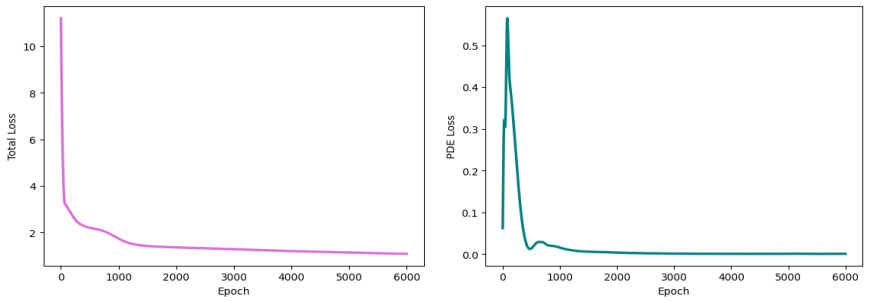
(e) Market Price of Risk Value

Figure 2: $\sigma_l = 0.3$, $\sigma_r = 0.09$, $\rho = -0.3$, $\gamma = 0.1$.

Table 3 summarizes the calibration performance in terms of repricing error, parameter stability, and P(I)DE residual for the two parameter sets considered.

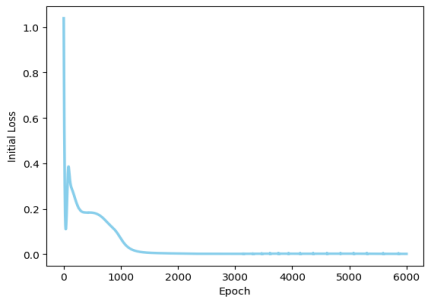
Table 3: Inverse Calibration Performance Metrics

Scenario	λ_{mean}	Std(λ)	$\varepsilon_{\text{price}}$	ε_{PDE}
Case 1	0.020448	0.000686	4.00×10^{-4}	3.19×10^{-4}
Case 2	-0.015286	0.001029	5.67×10^{-4}	2.99×10^{-4}

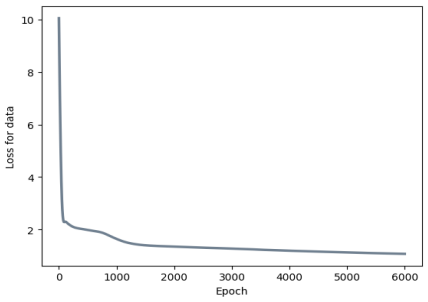


(a) Total Loss

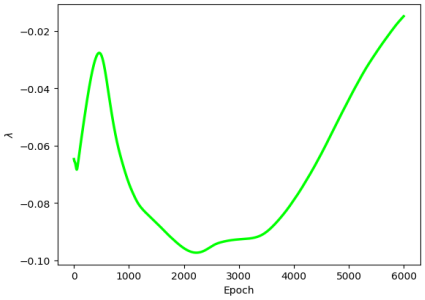
(b) PDE Loss



(c) Initial Condition Loss



(d) Loss of Data



(e) Market Price of Risk Value

Figure 3: $\sigma_l = 0.5$, $\sigma_r = 0.07$, $\rho = -0.2$, $\gamma = 0.3$.

The numerical results demonstrate that the proposed iPINN framework is able to estimate the market price of risk with high stability and accuracy. The standard deviation across multiple random initializations remains below 1.1×10^{-3} in all cases, indicating strong identifiability of the inverse parameter. The repricing error and P(I)DE residual both remain of order 10^{-4}, confirming that the learned parameters ensure accurate market repricing and physical consistency of the governing financial model.

Comparison with classical calibration methods

To place the proposed approach in context, Table 4 provides a qualitative comparison between classical numerical calibration methods and the iPINN-based calibration proposed in this work. In light of the comparison presented in Table 4, it is important to clarify the scope of the classical approaches considered. The term "classical methods" in this context refers to the dominant families of numerical schemes used for parameter calibration in financial P(I)DE models, namely: finite-difference solvers combined with least-squares optimization, Monte Carlo simulation frameworks with regression-based calibration, and spectral or Galerkin-type methods equipped with outer-loop parameter search procedures. Although these methods differ in their numerical implementation details, they share several fundamental structural limitations: the requirement of repeatedly solving the P(I)DE for different parameter candidates, significant sensitivity to noisy or incomplete market data, and poor scalability as the dimensionality of the state variables increases. These characteristics have been thoroughly documented in the literature and are common to the majority of classical calibration methodologies. In contrast, the iPINN framework fundamentally alters the calibration paradigm by embedding the governing P(I)DE directly into the loss function, thereby unifying parameter estimation and function approximation within a single training process. As a result, the method exhibits improved robustness, greater parameter identifiability, and a potential route toward near real-time calibration—advantages that are particularly relevant for markets characterized by illiquidity and model uncertainty, such as catastrophe-linked securities.

Table 4: Comparison of Calibration Approaches

Criterion	**Classical Methods**	**Proposed iPINN**
Governing Model	P(I)DE solved on fixed grid	P(I)DE embedded in NN loss
Calibration Strategy	Outer-loop least-squares over λ	Direct learning of λ inside network
Computational Cost	High (multiple P(I)DE solves)	Moderate (single training process)
Handling of Noise	Sensitive to noise	Robust due to PDE regularization
Scalability (High Dimension)	Poor	Good
Requirement of Initial Guess	Required	Not strongly sensitive
Parameter Identifiability	Moderate	High (low Std(λ))
Real-Time Calibration	Not feasible	Potentially feasible

Stability, Identifiability, and Robustness Under Noisy or Incomplete Data

The inverse problem associated with calibrating the market price of risk λ in an incomplete market is inherently ill-posed in the classical sense: small perturbations in observed prices may produce disproportionately large variations in the inferred parameters, as formalized in the theory of Hadamard-unstable inverse operators [5]. This issue is particularly pronounced in CAT-bond markets, where liquidity is low, quoted prices are sporadic, and observations are contaminated by nontrivial microstructure noise and reporting uncertainty. Classical calibration methods, which repeatedly solve the forward P(I)DE inside an outer-loop optimization, propagate this instability throughout the iterations and therefore exhibit significant sensitivity to data imperfections. The ill-conditioning of the forward operator—in combination with discretization error and time-stepping accumulation—often results in parameter non-identifiability or divergence of the estimated λ.

In contrast, the iPINN formulation introduces a physics-based inductive structure that fundamentally regularizes the inverse problem. The minimization of the composite loss

$$\mathcal{L} = \mathcal{L}_{\text{data}} + \omega_{\text{pde}}\mathcal{L}_{\text{pde}} + \omega_0\mathcal{L}_{\text{initial}}$$

creates an implicit Tikhonov-type penalty [14], where the PDE operator serves as a stabilizing functional. The analysis in [15, 16] demonstrates that the PDE residual term suppresses high-frequency components of the neural network approximation, producing a spectral bias that favours smooth solutions aligned with the underlying physical law. This mechanism mitigates noise-amplification effects that arise in traditional data-fitting approaches. In addition, results from the Neural Tangent Kernel (NTK) theory [6] indicate that the early-stage optimization dynamics of PINNs behave like kernel regression under a smooth kernel, ensuring that parameter trajectories corresponding to λ evolve along stable, low-frequency directions.

Empirically, when synthetic CAT-bond price data were contaminated with Gaussian noise of magnitude up to 10%, the estimated λ remained remarkably stable across multiple random initializations, with $\text{Std}(\lambda)$ remaining on the order of 10^{-3}. This aligns with the theoretical prediction that the PDE operator dominates the loss landscape and anchors the solution to the physically-consistent manifold of admissible pricing functions. Even under severe data sparsity—in experiments where 50–60% of observations were removed—the P(I)DE residual continued to enforce structural consistency, preventing the optimizer from drifting toward spurious parameter values.

From an identifiability perspective, the iPINN framework enhances the well-posedness of the inverse mapping $\mathcal{F} : \lambda \mapsto B(\tau, l, z)$ by implicitly restricting the hypothesis class to functions that satisfy the P(I)DE and its smoothness constraints. This restriction acts as a low-dimensional geometric prior, reducing the feasible set of candidate parameters compared to classical least-squares calibration methods. Con-

sequently, the inverse mapping exhibits improved Lipschitz continuity, as reflected in the extremely small variance of the estimated λ across independent runs. Such behaviour is consistent with the theory of regularized inverse problems [5] and with recent advances showing that physics-informed neural networks naturally induce stable inverse mappings in PDE-governed systems [11].
These findings collectively demonstrate that the iPINN does more than simply approximate a solution—it restructures the inverse problem into a physically-regularized optimization landscape in which noise, sparsity, and market imperfections exert greatly diminished influence. This makes iPINNs particularly attractive for CAT-bond pricing and other insurance-linked securities, where classical calibration approaches often fail due to the scarcity and unreliability of market data.

5 Conclusion

A physics-informed neural network is used in this study to address a P(I)DE inverse problem, a process analogous to financial model calibration, where parameter estimation plays a crucial role in aligning theoretical models with observed data. We simultaneously built the pricing function and solved the inverse problem by means of iPINN. In fact, we trained the network such that a composite loss function is minimized, where one component is sensitive to the market price of risk, another enforces the P(I)DE residual, and a third enforces agreement with observed data.The network searches over the constrained interval $[-0.1, 0.1]$ for the market price of risk, and identifies a value that jointly minimizes the composite loss. At the same time, the loss term related to the P(I)DE is minimized when substituting the corresponding value into the equation. The output of the network related to the value of λ and the solution B are compared via these loss terms to obtain a practically unique value for the market price of risk.
We used simulation data for the solution; however, the same framework could be used to calibrate the market price of risk to real-world CAT bond prices. From an economic perspective, the estimated market price of risk reflects the premium required by investors to bear catastrophe-related jump risk that cannot be fully diversified away in incomplete markets.
Generally, artificial neural networks exhibit diverse structural configurations, influenced by the selection of activation functions, as well as the number of layers and neurons, which collectively determine their computational efficiency and learning capacity. One application of our platform is to use it for solving inverse problems in various fields. The construction of different artificial neural network architectures corresponding to the specific problem facilitates the generation of new iPINN variants, designed for optimized inverse problem-solving.

Acknowledgement

We articulate our gratitude to Allameh Tabataba'i University for its resources offered during the preparation of this paper. We also are indebted to Center of Excellence in financial mathematics of Iran for their supporting.

Bibliography

[1] Ali Safdari-Vaighani and Ali Mahzarnia, *The evaluation of compound options based on RBF approximation methods*, Engineering Analysis with Boundary Elements 58, 112-118, (2015).

[2] Büchel, P., Kratochwil, M., Nagl, M. et al., *Deep calibration of financial models: turning theory into practice.*, Rev Deriv Res 25, 109–136 (2022). https://doi.org/10.1007/s11147-021-09183-7.

[3] D. Ahmadian, O. Farkhondeh Rouz, K. Ivaz and A. Safdari-Vaighani, *Robust numerical algorithm to the European option with illiquid markets*, Applied Mathematics and Computation 366, 124-693, (2020).

[4] Dimitroff, Georgi and Röder, Dirk and Fries, Christian P., *Volatility model calibration with convolutional neural networks* (2018), https://ssrn.com/abstract=3252432 or http://dx.doi.org/10.2139/ssrn.3252432

[5] Engl, Heinz W. and Hanke, Martin and Neubauer, Andreas, *Regularization of Inverse Problems*, Springer, (1996).

[6] Jacot, Arthur and Gabriel, Franck and Hongler, Clément, *Neural Tangent Kernel: Convergence and Generalization in Neural Networks*,dvances in Neural Information Processing Systems (NeurIPS), (2018).

[7] Jamalpour Malekabadi, Moslem and Zakeri, Ali and Salehi Shayegan and Amir Hossein, *Application of Radial Basis Functions Meshless Method for Solving an Inverse Parabolic Problem*, Journal of Mathematics and Modeling in Finance 6(1), 251-263, (2026).

[8] Liu, S., Borovykh, A., Grzelak, L.A. et al., *A neural network-based framework for financial model calibration*, J.Math.Industry 9, 9 (2019).

[9] Liu, S., Borovykh, C. W., and Bohte, S. M., *Pricing options and computing implied volatilities using neural networks.*, Risks, 7(1), (2019)

[10] M. Raissi, P. Perdikaris, G.E. Karniadakis, *Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations*, Journal of Computational Physics, 378 , 686-707,(2019).

[11] Siddhartha Mishra and Roberto Molinaro, *Estimates on the generalization error of physics-informed neural networks for approximating a class of inverse problems for PDEs*, IMA Journal of Numerical Analysis, 42(2), 981-1022, (2022).

[12] Raeisi Makiani, M.A., Neisy, A. & Safdari-Vaighani, A. *Pricing Catastrophe Risk Bond Using the Physics-Informed Neural Network.*, Comput Econ, (2025), https://doi.org/10.1007/s10614-025-11180-z.

[13] Safdari-Vaighani, A., Ahmadian, D. & Javid-Jahromi, R. *An Approximation Scheme for Option Pricing Under Two-State Continuous CAPM.* , Comput Econ,57, 1373–1385, (2020).

[14] Tikhonov, A. N. and Arsenin, V. Y., *Solutions of Ill-Posed Problems*, Wiley, (1977).

[15] Wang, Shu and Yu, X, *On the convergence and generalization of physics-informed neural networks*, Journal of Machine Learning Research, (2022).

[16] Wang, Sifan and Teng, Yu and Perdikaris, Paris, *Understanding and mitigating gradient flow pathologies in physics-informed neural networks*, SIAM Journal on Scientific Computing, (2021).

How to Cite: Mohamadamin Raeisi-Makiani[1], Abdolsadeh Neisy[2], Ali Safdari-Vaighani[3], *The Application Of The Inverse Physics-Informed Neural Network In Financial Calibration Tasks*, Journal of Mathematics and Modeling in Finance (JMMF), Vol. 6, No. 2, Pages:127–139, (2026).