



# Advanced Physics-Informed Neural Network with Residuals for Solving Complex Integral Equations

Mahdi Movahedian Moghaddam<sup>1</sup>, Kourosh Parand<sup>1,2,\*</sup>, Saeed Reza Kheradpisheh<sup>1</sup>

<sup>1</sup> Department of Computer and Data Sciences, Faculty of Mathematical Sciences, Shahid Beheshti University, Tehran, Iran

<sup>2</sup> Department of Cognitive Modeling, Institute for Cognitive and Brain Sciences, Shahid Beheshti University

\* Corresponding author(s): [k\\_parand@sbu.ac.ir](mailto:k_parand@sbu.ac.ir)

Received: 18/02/2026 Revised: 16/04/2026 Accepted: 02/06/2026 Published: 07/06/2026

10.22128/ansne.2026.3261.1200

## Abstract

In this paper, we present the Residual Integral Solver Network (RISN), a novel neural network architecture designed to solve a wide range of integral and integro-differential equations, including one-dimensional, multi-dimensional, ordinary and partial integro-differential, systems, fractional types, and Helmholtz-type integral equations involving oscillatory kernels. RISN integrates residual connections with high-accuracy numerical methods such as Gaussian quadrature and fractional derivative operational matrices, enabling it to achieve higher accuracy and stability than traditional Physics-Informed Neural Networks (PINN). The residual connections help mitigate vanishing gradient issues, allowing RISN to handle deeper networks and more complex kernels, particularly in multi-dimensional problems. Through extensive experiments, we demonstrate that RISN consistently outperforms not only classical PINNs but also advanced variants such as Auxiliary PINN (A-PINN) and Self-Adaptive PINN (SA-PINN), achieving significantly lower Mean Absolute Errors (MAE) across various types of equations. These results highlight RISNs robustness and efficiency in solving challenging integral and integro-differential problems, making it a valuable tool for real-world applications where traditional methods often struggle.

**Keywords:** Residual connections, Deep learning, Gradient flow optimization

**Mathematics Subject Classification (2020):** 68T07, 65R20

## 1 Introduction

Integral and integro-differential equations are foundational tools in many fields of science and engineering, modeling a wide range of phenomena from physics and biology to economics and engineering systems [1–3]. These equations describe processes that depend not only on local variables but also on historical or spatial factors, making them essential for understanding systems with memory effects, hereditary characteristics, and long-range interactions [4–7]. Despite their importance, solving integral and integro-differential equations is a challenging task due to the complexity of their integral operators, especially when extended to multi-dimensional or fractional forms [2, 8].

Classical numerical methods, such as finite difference [9, 10], finite element [11, 12], and spectral methods [13–15], have long been used to approximate solutions to these equations. However, these methods often suffer from several limitations. For example, they require fine

discretization of the problem domain, leading to high computational costs and memory usage, especially for high-dimensional problems [16–19]. Additionally, these methods can struggle with non-local operators, singular kernels, or fractional derivatives, which introduce further complexities [20]. Even though more recent approaches, like Gaussian quadrature and collocation methods, provide enhanced accuracy for certain cases, their applicability is often restricted to simpler, well-behaved systems [5, 8, 20–22].

In recent years, deep learning has emerged as a powerful tool to overcome the limitations of traditional statistical methods [23, 24]. Physics-informed neural networks (PINNs) represent a major advance in this direction by directly integrating the governing physical laws with loss of neural activity [25–27]. This allows PINNs to learn solutions to differential equations using observational data while ensuring consistency with underlying physics [28, 29]. PINNs have shown promising results in solving various types of differential equations, including ordinary and partial differential equations [30–32]. However, PINNs are not without their limitations. The depth of the network often exacerbates issues such as vanishing gradients, and the absence of residual connections can make training deep networks unstable, especially when dealing with highly complex or multi-dimensional integral and integro-differential equations [33–36]. Although extensions like Auxiliary PINN (A-PINN) [37] and Self-Adaptive PINN (SA-PINN) [38] have been developed to address some limitations, they still encounter significant challenges in solving complex integral and integro-differential equations, motivating the need for the RISN architecture.

To overcome these challenges, we propose the Residual Integral Solver Network (RISN), a novel extension of the PINN framework specifically designed to solve integral and integro-differential equations. RISN enhances the original PINN architecture by incorporating residual connections, a proven technique in deep learning that significantly improves gradient flow, reduces training instability, and allows for deeper network architectures. These residual connections are crucial for ensuring stable training, particularly when handling multi-dimensional or highly non-linear systems [25, 32, 33]. Furthermore, RISN leverages high-accuracy numerical techniques, such as Gaussian quadrature for efficient and precise calculation of integral terms, and fractional operational matrices to handle fractional derivatives with minimal error [21, 39]. This combination of deep learning with classical numerical methods allows RISN to achieve high accuracy and stability across a wide range of equation types, including those that pose challenges for traditional methods and standard PINNs.

In this paper, we contribute to the growing field of deep learning for solving complex mathematical problems by presenting the following innovations: (1) the development of a residual-based neural network model capable of solving integral and integro-differential equations of various types, including one-dimensional, multi-dimensional, fractional systems, and oscillatory Helmholtz-type integral equations; (2) the integration of advanced numerical techniques within the RISN framework to enhance precision in the calculation of integral and differential operators; and (3) an extensive evaluation of the RISN model across a broad spectrum of integral equations, demonstrating superior performance in terms of accuracy, stability, and efficiency when compared to classical PINNs.

The rest of this paper is organized as follows: Section 2 provides a literature review of existing methods for solving integral and integro-differential equations, with a focus on the strengths and limitations of classical numerical approaches and PINNs. In Section 3, we introduce the RISN model in detail, including its architecture, the integration of residual connections, and the numerical techniques employed. Section 4 presents the experimental results, showcasing the RISN model's performance across different equation types. Finally, in Section 5, we discuss the implications of our findings and propose directions for future work.

## 2 Literature Review

The solution of integral and integro-differential equations plays a critical role across various scientific disciplines, including physics, biology, and engineering. These equations model systems with local and non-local interactions, which makes them essential for understanding phenomena ranging from heat conduction and population dynamics to quantum mechanics and fluid dynamics. Traditionally, methods such as finite difference and finite element approaches have been employed to solve these equations [40, 41]. Although these techniques have been widely successful, they face significant computational challenges, particularly when applied to high-dimensional problems [42–44].

The finite difference method (FDM) approximates differential operators by discretizing the domain, making it simple to implement for many problems. However, FDM struggles with large-scale problems due to the need for fine grid discretization, which leads to high computational cost and memory usage [45]. Finite element methods (FEM), while more versatile and effective for irregular domains, encounter similar issues in high-dimensional systems, where the complexity of mesh generation and element assembly increases exponentially with the number of dimensions [45, 46]. The inefficiency of these methods in handling complex multi-dimensional and fractional equations has spurred the development of alternative approaches, including machine learning-based methods [46, 47].

In recent years, Artificial Neural Networks (ANNs) have emerged as a powerful alternative for solving complex mathematical equations, including Fredholm and Volterra integral equations. Early work by [48] demonstrated the use of ANNs in solving Fredholm integral equations, highlighting their potential for improving computational efficiency and accuracy. Further advancements were made by [49], who extended neural network methods to handle fuzzy integro-differential equations, paving the way for the application of ANNs to more complex and uncertain systems.

As research into ANNs progressed, attention shifted to solving more complex problems, such as multi-dimensional and fractional integro-differential equations. The research [50] successfully applied ANNs to Volterra integral equations, demonstrating superior performance compared to traditional methods in terms of both speed and accuracy. In parallel, [51] employed ANNs to tackle two-dimensional integral equations, while [52] utilized ANNs for fractional-order equations, both demonstrating significant improvements in computational efficiency.

A major leap forward in this field came with the introduction of Physics-Informed Neural Networks (PINNs). PINNs embed the physical laws governing a system directly into the neural networks architecture, allowing for the solution of complex partial differential equations (PDEs) and integro-differential equations with increased accuracy. For instance, [53] demonstrated how PINNs could be used to solve volume integral equations, combining physical insights with deep learning techniques to ensure physically consistent solutions. Furthermore, the DeepXDE framework, introduced by [27], extended PINNs to a wide variety of differential and integro-differential equations, emphasizing their flexibility in handling real-world problems.

Recent approaches, such as DeepONets [54] and Fourier Neural Operators [55], have pushed the boundaries of solving high-dimensional equations. However, these models tend to struggle with generalization across different types of equations, particularly those involving non-local operators or fractional components.

An important development in this area is the PINNIES framework [39], which integrates physics-informed strategies to handle integral operator problems. This framework has been particularly effective in addressing multi-dimensional and fractional equations, offering enhanced accuracy and stability compared to traditional methods.

Recent studies [56–61] have further advanced neural network-based approaches for solving complex differential and integral equations. For instance, recent works have explored improved deep learning architectures and hybrid numerical-ML and evolutionary techniques for handling nonlinear and high-dimensional systems. These approaches demonstrate enhanced accuracy and generalization capabilities, particularly in challenging scenarios involving fractional dynamics and nonlocal operators. The integration of such advancements highlights the growing importance of combining deep learning with numerical analysis for solving complex mathematical models.

However, despite their successes, deep neural networks, including PINNs, suffer from the vanishing gradient problem, which occurs when the gradients used for backpropagation become very small in deep architectures. This results in slow or stalled learning as the network struggles to adjust its weights. To mitigate this issue, residual connections were introduced in deep learning models. These connections, first proposed by [33] in the context of ResNets, allow gradients to flow more easily through deeper layers, making training more stable and efficient. Recent adaptations of residual learning into the PINN framework have been shown to effectively handle high-dimensional problems and integral equations, improving both convergence speed and solution accuracy.

In addition to PINNs, recent works have introduced innovative neural network architectures for solving fractional-order equations and integro-differential systems. For example, [62] applied ANNs to fractional Volterra integral equations, while [63] explored extended neural networks for solving two-dimensional fractional equations, achieving superior precision through techniques such as sine-cosine basis functions and extreme learning machines (ELM). These methods provide robust solutions to problems that classical approaches often find intractable.

Several variants of PINNs have been proposed to improve stability and convergence, such as the Auxiliary PINN (A-PINN) [37], which augments the loss function with auxiliary conditions, and the Self-Adaptive PINN (SA-PINN) [38], which introduces adaptive weighting schemes for the loss components. While these methods offer improvements in specific contexts, they often fail to generalize to more complex, multi-dimensional, and fractional problems due to the lack of structural enhancements like residual connections.

Furthermore, the evolution of neural network techniques, particularly the development of PINNs (such as PINNIES, and A-PINN), has transformed the landscape of integral equation solving. These frameworks offer significant improvements in computational efficiency, accuracy, and the ability to handle complex, high-dimensional, and fractional systems, representing a substantial advancement over traditional methods.

### 3 Methodology

The Residual Integral Solver Network (RISN) is a novel neural network architecture to address integral and integro-differential equations efficiently. Inspired by traditional neural networks, RISN integrates residual connections to mitigate the vanishing gradient problem and enable deeper network training. The model is structured to handle both differential and integral operators by incorporating these connections, which not only improve training stability but also enhance the accuracy of the solutions. By combining classical numerical techniques like Gaussian numerical quadrature and fractional operational matrices with the residual structure, RISN is capable of solving a wide range of complex mathematical problems, including integral, fractional integro-differential, and oscillatory Helmholtz-type integral equations. This structure is chosen for solving integral equations due to its ability to maintain stability and accuracy in deep networks. Residual connections improve gradient propagation during backpropagation, reducing the risk of vanishing gradients in deep architectures. Additionally, by learning residual mappings instead of direct function approximations, the network effectively models corrections to simpler approximations, which leads to improved convergence behavior and higher solution accuracy, particularly in complex and high-dimensional problems. This section provides a detailed explanation of the RISN structure and the numerical techniques employed to achieve high precision in solving these equations.

#### 3.1 The Model Structure

The RISN is an advanced neural network architecture designed to address integral and integro-differential equations by integrating residual connections. The structure of RISN is inspired by traditional neural networks but enhances them with residual connections to improve training efficiency and solution accuracy.

Residual connections, a critical component in our model, allow for the preservation of gradient flow during backpropagation, thereby preventing the vanishing gradient problem typically seen in deep networks. This feature is particularly important in multi-dimensional and fractional integro-differential equations, where the complexity of the solution requires a stable and accurate gradient propagation mechanism.

The architecture of the RISN, as illustrated in Figure 1, is designed to solve integral and integro-differential equations by combining residual deep neural networks with classical numerical methods. This figure provides an overview of the data flow and interaction between different parts of the network. The equation (1) represents the target system, where  $F(u)(\mathbf{x})$ ,  $D(u)(\mathbf{x})$ , and  $I(u)(\mathbf{x})$  correspond to the differential, integral, and source terms, respectively.

$$F(u)(\mathbf{x}) + D(u)(\mathbf{x}) + I(u)(\mathbf{x}) = S(\mathbf{x}). \quad (1)$$

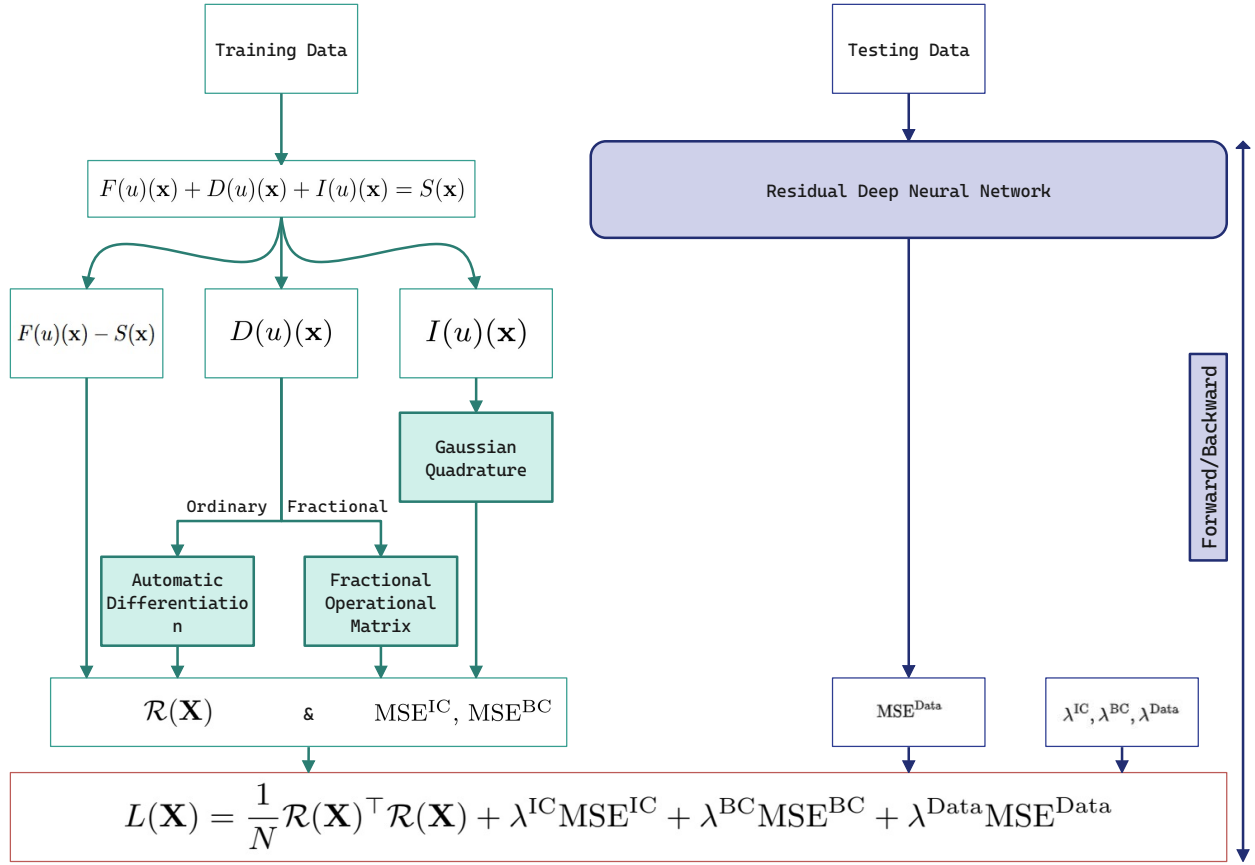
The model begins with the *training data*, which feeds into different branches corresponding to the components of the equation. The source and function terms are handled through the combination of  $F(u)(\mathbf{x}) + S(\mathbf{x})$ . For the differential operator  $D(u)(\mathbf{x})$ , two approaches are used: *automatic differentiation* for ordinary derivatives and the *fractional operational matrix* for fractional derivatives. The integral term  $I(\mathbf{X})$  is computed using *Gaussian quadrature*, a highly accurate method for numerical quadrature.

Meanwhile, *testing data* is passed through the residual deep neural network. This neural network incorporates residual connections, which enhance training stability and enable the model to learn deeper networks without suffering from the vanishing gradient problem. This allows the RISN to converge more effectively and improve accuracy in solving complex integral and integro-differential equations.

The residual network model that should find  $u(x)$  is defined as follows:

$$\begin{aligned} A_0 &= \mathbf{X}, & \mathbf{X} &\in \mathbb{R}^{N \times d}, \\ A_i &= \sigma_i(A_{i-1}\boldsymbol{\theta}^{(i)} + \mathbf{b}^{(i)}) + A_{i-1}, & i &= 1, 2, \dots, L-1, \\ \text{MLP}(\mathbf{X}) &:= A_L = A_{L-1}\boldsymbol{\theta}^{(L)} + \mathbf{b}^{(L)}, & A_L &\in \mathbb{R}^{N \times 1}. \end{aligned} \quad (2)$$

The input to the model is denoted by  $\mathbf{X}$ , which is a matrix with  $N$  data points, each having  $d$  dimensions. For  $i = 1, 2, \dots, L-1$ , each hidden layer  $A_i$  is computed by applying a nonlinear activation function  $\sigma_i$  to a linear transformation of the previous layer  $A_{i-1}$  with parameters  $\boldsymbol{\theta}^{(i)}$  and bias  $\mathbf{b}^{(i)}$ , and then adding the residual connection from the previous layer  $A_{i-1}$ . This residual connection helps in mitigating the vanishing gradient problem and allows for deeper network training. The final output  $A_L$  is produced by applying a linear transformation to the last hidden layer  $A_{L-1}$  with parameters  $\boldsymbol{\theta}^{(L)}$  and bias  $\mathbf{b}^{(L)}$ .



**Figure 1.** Flowchart of the RISN methodology. The equation  $F(u)(\mathbf{x}) + D(u)(\mathbf{x}) + I(u)(\mathbf{x}) = S(\mathbf{x})$  is decomposed into three components:  $F(u)(\mathbf{x}) - S(\mathbf{x})$ ,  $D(u)(\mathbf{x})$ , and  $I(u)(\mathbf{x})$ . The fractional derivative terms are handled using fractional operational matrices, while Gaussian quadrature is employed for the numerical quadrature of the integral term  $I(u)(\mathbf{x})$ . The output is optimized through residual deep neural networks, where mean squared error (MSE) terms for initial conditions, boundary conditions, and data are combined to form the loss function  $L(x)$ , ensuring accurate predictions.

In the proposed RISN architecture, the residual connections follow a layer-wise formulation, where each hidden layer receives a direct skip connection from its immediately preceding layer. This corresponds to a residual mapping of the form  $A_i = F(A_{i-1}) + A_{i-1}$ , where  $F$  represents the nonlinear transformation defined by the layer parameters and activation function. It is important to note that the residual mapping operates purely at the neural network level and does not explicitly include the integral or differential operators. These operators are incorporated separately through the physics-informed loss function, ensuring a clear separation between the network architecture and the governing equation constraints.

The objective is to train the RISN to minimize the residual:

$$\mathcal{R}(\mathbf{X}) := F(\mathbf{u}) + D(\mathbf{u}) + I(\mathbf{u}) - S(\mathbf{X}).$$

After processing through both the neural network and numerical techniques, the model computes several types of error, including *mean squared error* (MSE) for data-driven predictions and separate errors for initial conditions ( $\text{MSE}^{\text{IC}}$ ) and boundary conditions ( $\text{MSE}^{\text{BC}}$ ). These errors are combined into the overall loss function (3), which also includes the residual terms  $\mathcal{R}(x)$ . The parameters  $\lambda^{\text{IC}}$ ,  $\lambda^{\text{BC}}$ , and  $\lambda^{\text{Data}}$  serve as regularization coefficients. The final loss function is a weighted sum of the errors, using parameters  $\lambda$  to balance the contributions from each component.

$$L(\mathbf{X}) = \frac{1}{N} \mathcal{R}(\mathbf{X})^\top \mathcal{R}(\mathbf{X}) + \lambda^{\text{IC}} \text{MSE}^{\text{IC}} + \lambda^{\text{BC}} \text{MSE}^{\text{BC}} + \lambda^{\text{Data}} \text{MSE}^{\text{Data}}, \quad (3)$$

When solving a system of  $M$  equations, the loss function  $L(\mathbf{X})$  is computed as the sum of residuals across all equations, normalized by the product of the number of training data points  $N$  and the number of equations  $M$ . The general form of the loss function is given by:

$$L(\mathbf{X}) = \frac{1}{N \times M} \sum_{i=1}^M \mathcal{R}_i(\mathbf{X})^\top \mathcal{R}_i(\mathbf{X}) + \sum_{i=1}^M \left( \lambda_i^{\text{IC}} \text{MSE}_i^{\text{IC}} + \lambda_i^{\text{BC}} \text{MSE}_i^{\text{BC}} + \lambda_i^{\text{Data}} \text{MSE}_i^{\text{Data}} \right). \quad (4)$$

Here,  $\mathcal{R}_i(\mathbf{X})$  represents the residual of the  $i$ -th equation, and the mean squared error terms  $\text{MSE}_i^{\text{IC}}, \text{MSE}_i^{\text{BC}}, \text{MSE}_i^{\text{Data}}$  correspond to the initial condition (IC), boundary condition (BC), and testing data, respectively. The weighting factors  $\lambda_i^{\text{IC}}, \lambda_i^{\text{BC}}, \lambda_i^{\text{Data}}$  control the relative contribution of each term to the overall loss, ensuring that the optimization process takes into account the specific requirements of the initial, boundary, and data constraints for each equation in the system.

## 3.2 Numerical Techniques for Integral and Fractional Differential Operators

In this work, two key numerical techniques have been employed to enhance the accuracy and efficiency of solving integral and fractional integro-differential equations: Gaussian numerical quadrature and fractional operational matrices. Both methods, though well-established in the field of numerical analysis, were first integrated into the PINN framework in the work of [39]. The use of these techniques enhances numerical accuracy and reduces errors from numerical approximations. Their integration into the RISN further solidifies the models capacity to tackle complex mathematical problems.

### 3.2.1 Gaussian Numerical Quadrature

Gaussian quadrature, particularly well-suited for evaluating definite integrals, is leveraged in this model to compute the integral terms that arise in the solution of integro-differential equations. As described in the work of PINNIES, this method offers a high degree of accuracy with relatively few evaluation points, making it a computationally efficient choice. In the context of RISN, Gaussian quadrature allows for the precise calculation of integral terms, which are critical to solving the targeted problems. Its inclusion ensures that the integral operators are handled with minimal numerical error, thus enhancing the overall accuracy of the model.

### 3.2.2 Fractional Operational Matrices

Fractional differential operators are essential when dealing with integro-differential equations that exhibit non-local behavior, such as systems with memory effects. The method of using fractional operational matrices, initially introduced in the context of PINNs by PINNIES, allows for a structured and efficient way of handling these operators. By representing fractional derivatives in matrix form, the RISN model can compute fractional derivatives with high precision, seamlessly integrating this approach into the neural network architecture. This not only improves the accuracy of fractional derivative calculations but also streamlines the process within the networks training and inference stages.

In these problems, fractional derivatives, which generalize the concept of integer-order differentiation, allow for a more accurate representation of processes with memory effects or non-local dependencies. RISN, combined with operational matrices, efficiently handles these derivatives by leveraging its residual connections to maintain stability in training.

The combination of these two techniques—Gaussian numerical quadrature and fractional operational matrices—within RISN reflects the practical benefits of merging classical numerical methods with advanced machine learning models. While these techniques do not significantly differ from their original implementation in the PINN framework, their application within RISN ensures that the model can solve a wide range of integral and integro-differential equations more effectively.

The RISN model leverages residual connections to enhance the training process and accuracy of solving integral and integro-differential equations. By integrating residual connections with a structured loss function, RISN improves solution fidelity and robustness, making it a powerful tool for addressing complex mathematical problems. However, the proposed method may require additional tuning when dealing with highly complex problems.

## 3.3 Functional Approximation Perspective in Sobolev Spaces

To better understand the mathematical advantage of the residual formulation adopted in RISN, we analyze the problem from the viewpoint of functional approximation in Sobolev spaces.

Let  $y^* \in W^{k,p}(\Omega)$  denote the exact solution of the integral equation, where  $k \geq 1$ ,  $p \geq 1$ , and  $\Omega \subset \mathbb{R}^d$  is a compact domain. Let  $g(x)$  be a known approximation such that  $\|y^* - g\|_{W^{k,p}}$  is small. Then, the residual network is tasked with approximating the correction term:

$$r(x) := y^*(x) - g(x).$$

Since  $r$  often exhibits smoother behavior than  $y^*$  itself, the approximation of  $r$  using a neural network  $N(x; \theta)$  becomes more efficient.

**Sobolev Approximation Advantage.** Using classical results on Sobolev approximation (e.g., [64–66]), we know that for neural networks with sufficient width and depth, there exists a parameter  $\theta$  such that:

$$\|r - N(\cdot; \theta)\|_{W^{k,p}} \leq \varepsilon,$$

with  $\varepsilon$  decreasing at a rate depending on the smoothness of  $r$  and the architecture of the network.

**Implication.** Thus, the total approximation error satisfies:

$$\|y^* - \hat{y}\|_{W^{k,p}} = \|y^* - (g + N(\cdot; \theta))\|_{W^{k,p}} = \|r - N(\cdot; \theta)\|_{W^{k,p}} \leq \varepsilon,$$

which is potentially smaller than directly approximating  $y^*$  by a network alone, particularly when  $g$  captures the coarse structure of  $y^*$ .

This analysis shows that the residual architecture not only simplifies the learning task but also benefits from improved theoretical approximation bounds in Sobolev spaces.

## 4 Experimental Results

In this section, we present the results obtained from solving various types of integral and integro-differential equations using the RISN. The examples provided cover a range of equations, including one-dimensional and multi-dimensional problems, systems of equations, and Helmholtz-type integral equations. The structure of this section is as follows: first, we outline the setup for each example, including the equation type, domain, and kernel functions. Next, we present the results for each test case and compare the performance of RISN with classical PINN as well as advanced variants including A-PINN and SA-PINN.

For all models evaluated in this study, the loss function is composed of the physics-informed residual loss and the boundary/initial condition loss. Specifically, the total loss is defined as:

$$\mathcal{L}_{total} = \mathcal{L}_{physics} + \mathcal{L}_{boundary/initial},$$

where  $\mathcal{L}_{physics}$  measures the violation of the underlying integral or integro-differential equation across collocation points, and  $\mathcal{L}_{boundary/initial}$  enforces the satisfaction of the given boundary or initial conditions. No adaptive loss weighting strategies were applied; all components contribute equally to the total loss. The use of fixed weighting is intentionally adopted to ensure a fair comparison between the baseline PINN and the proposed RISN model. By avoiding adaptive weighting strategies, the evaluation focuses solely on the architectural improvements introduced by the residual connections, without introducing additional optimization mechanisms that could confound the comparison. Note that the additional residual terms in the RISN model are seamlessly integrated into the physics loss without altering the fair comparison setup.

The RISN model used in these experiments consists of a fully connected neural network with seven hidden layers, each containing 20 neurons. The  $Tanh$  activation function is applied in each layer, and the model is optimized using the L-BFGS optimizer with a learning rate of 0.01. The number of training points (collocation points) and the number of Gaussian quadrature nodes are both set to 50, and the loss function incorporates residuals for the target equation, as well as terms for initial and boundary conditions, with appropriate weighting parameters  $\lambda$ . Additionally, Gaussian quadrature (specifically GaussLegendre) is employed for accurate numerical quadrature, and fractional operational matrices are used for handling fractional derivatives where necessary.

In order to ensure a fair comparison, all models were trained using the same network architecture, with identical numbers of hidden layers and neurons per layer. Moreover, the same optimization settings, learning rates, and stopping criteria were applied across all experiments.

The proposed method was implemented in Python 3.11 utilizing the PyTorch framework (Version 2.3.1), which provides automatic differentiation capabilities and supports efficient large-scale computations. All experiments were conducted on a personal computer

equipped with an Intel Core i5-13420H CPU and 8 GB of RAM, operating under Microsoft Windows 11. This computational setup was sufficient to perform the training and evaluation of the RISN model across all considered problems, without encountering significant memory or performance bottlenecks.

The performance of the RISN model was evaluated using the Mean Absolute Error (MAE) as the primary error metric, ensuring a clear assessment of the models accuracy in solving the various types of integral and integro-differential equations.

## 4.1 Integral Equations

### 4.1.1 One-Dimensional Integral Equations

In this section, we address one-dimensional integral equations of the form:

$$\kappa u(x) = S(x) + \int_{\Delta} K(x,t)\zeta(u(t)) dt,$$

where  $S(x)$  is the source term,  $K(x,t)$  represents the kernel of the integral operator, and  $\zeta(x)$  characterizes the linearity of the problem. The constant parameter  $\kappa \in \mathbb{R}$ , if  $\kappa = 0$  is a first-kind integral equation (IE); otherwise, it is classified as a second-kind IE. The domain of the problem, denoted as  $\Delta = [a, b]$ , applies to all equation types. For the Fredholm operator, we assume the interval  $a, b = 0, 1$ , and for the Volterra operator, we take  $g(x), h(x) = 0, x$  for  $x \in \Delta$ . In the case of the Volterra-Fredholm problem, the equation is given by:

$$\kappa u(x) = S(x) + \int_a^b K_f(x,t)\zeta(u(t)) dt + \int_0^x K_v(x,t)\zeta(u(t)) dt.$$

Using the approach outlined in Section 3, we simulate a range of integral equations and provide the problem details along with the MAE results in Table 1. It was found that, with the exception of Abel-type singular integral equations, the method yielded highly accurate results. For Abel problems, accuracy can be improved by increasing the number of nodal points or utilizing a different Gaussian quadrature method, such as an alternative to Gauss-Legendre.

The table presents a comparative analysis of the MAE between the proposed RISN method and the classical PINN approach for solving one-dimensional integral equations. The results clearly indicate that the RISN method consistently achieves lower MAE across different types of integral equations, including Volterra, Volterra-Fredholm, and Abel types. This demonstrates the superior accuracy and robustness of the RISN approach, particularly in handling complex integral equations with varying kernels and source terms. Notably, for the Abel-type equations, although the errors are generally higher, the RISN still outperforms the traditional PINN, illustrating its effectiveness in tackling singular kernel challenges. As shown in the last row of the table, the traditional PINN model struggles with nonlinear Abel-type problems and is unable to solve them, whereas the proposed RISN model successfully solves them with satisfactory accuracy.

**Table 1.** Comparative analysis of MAE for one-dimensional integral equations using the proposed RISN versus classical PINN methods. The RISN consistently achieves lower MAE, highlighting its enhanced accuracy and reliability across various equation types and conditions.

Type	$\Delta$	$\zeta(x)$	$\kappa$	Exact	Source term	Kernel	PINN MAE	RISN MAE
Volterra	[0, 1]	$x$	1	$x + e^x$	$2e^x - 1 + \frac{x^3}{6}$	$t - x$	$9.53 \times 10^{-3}$	$8.46 \times 10^{-6}$
Volterra	[0, 1]	$x^2$	1	$e^x$	$e^x - \frac{1}{2}(e^{2x} - 1)$	1	$2.80 \times 10^{-4}$	$1.05 \times 10^{-5}$
Volterra-Fredholm	[0, 1]	$x$	1	$x + e^x$	$2e^x - \frac{x}{2} - \frac{7}{3} + \frac{x^3}{6} + xe$	$K_f = K_v = t - x$	$1.95 \times 10^{-4}$	$7.05 \times 10^{-6}$
Volterra-Fredholm	[0, 1]	$x$	1	$xe^x$	$e^x - 1 - x$	$K_f = x, K_v = 1$	$3.40 \times 10^{-4}$	$1.48 \times 10^{-5}$
Abel	[0, 1]	$x$	0	$x$	$\frac{4}{3}x^{\frac{3}{2}}$	$\frac{-1}{\sqrt{x-t}}$	$3.27 \times 10^{-3}$	$3.27 \times 10^{-3}$
Abel	[0, 1]	$x^3$	0	$x$	$\frac{32}{35}x^{\frac{7}{2}}$	$\frac{-1}{\sqrt{x-t}}$	—	$1.70 \times 10^{-3}$

The results obtained for one-dimensional Volterra and Fredholm integral equations indicate that RISN consistently achieves lower MAE compared to the baseline PINN model. The superior accuracy of RISN can be attributed to its residual connections, which enhance gradient flow and facilitate the training process, particularly in handling non-smooth or weakly singular kernels. The PINN model, by contrast, shows difficulties in precisely approximating the solution when the kernel complexity increases.

### 4.1.2 Multi-Dimensional Integral Equations

In this section, we assess the proposed approach for solving multi-dimensional integral equations. We specifically examine a two-dimensional integral equation structured as follows:

$$\kappa u(x, y) = S(x, y) + \int_{\Delta_y} \int_{\Delta_x} K(x, y, s, t) u(s, t) dt ds,$$

and a three-dimensional integral equation:

$$\kappa u(x, y, z) = S(x, y, z) + \int_{\Delta_z} \int_{\Delta_y} \int_{\Delta_x} K(x, y, z, r, s, t) u(r, s, t) dt ds dr.$$

These integral equations are categorized into Fredholm or Volterra types and further classified as first-kind or second-kind. Utilizing the formulations detailed in Section 3, we conduct simulations on various integral equations, employing different kernel functions and exact solutions.

The results in Table 2 demonstrate that RISN achieves significantly lower MAE in both Fredholm and Volterra types of equations. This indicates that RISN provides more accurate solutions, particularly in complex multi-dimensional kernel scenarios. The improvements in accuracy with RISN highlight its robustness and efficiency over traditional PINN methods.

The residual connections in the RISN model have shown significant advantages, particularly in solving multi-dimensional integro-differential equations. These connections allow for better gradient flow throughout the network, reducing the likelihood of vanishing gradients and improving the model's ability to capture complex interactions between variables. This is especially important in multi-dimensional problems, where the complexity of the solution space can challenge traditional methods like PINN. By incorporating residual connections, RISN not only enhances stability but also achieves higher accuracy in these challenging scenarios, as demonstrated by the consistently lower MAE values observed in the experiments.

**Table 2.** Comparison of MAE for multi-dimensional integral equations using RISN versus classical PINN methods. RISN shows improved accuracy, demonstrating its effectiveness in handling complex Fredholm and Volterra integral equations.

Type	$\Delta$	Exact	Source Term	Kernel	PINN MAE	RISN MAE
Fredholm	$[0, 1] \times [0, 2]$	$x^2y$	$x^2y + \frac{4}{9}x$	$-\frac{1}{2}xt$	$2.85 \times 10^{-3}$	$1.31 \times 10^{-4}$
Volterra	$[0, 1] \times [0, 2]$	$x + y$	$(x + y - 2)e^{2x+2y} + (2 - y)e^{x+2y} + (2 - x)e^{2x+y} + x + y - 2e^{x+y}$	$e^{x+y+s+t}$	$4.34 \times 10^{-2}$	$5.64 \times 10^{-3}$

### 4.1.3 System of Integral Equations

A system of integral equations involves multiple unknown functions that are interconnected and appear within the integral terms. For an integer  $M \geq 2$ , such a system can be expressed mathematically as:

$$\kappa u_i(x) = S_i(x) + \int_{\Delta} \sum_{i=1}^I K_{i,i}(x, t) u_i(t) dt. \quad (5)$$

These systems often combine different types of integral equations previously discussed. For example, when  $\kappa = 0$ , the system reduces to a set of first-kind integral equations. In this section, we focus on the case where  $M = 2$ , which results in the following system of equations:

$$\begin{cases} \kappa u_1(x) = S_1(x) + \int_{\Delta} [K_{1,1}(x, t) u_1(t) + K_{1,2}(x, t) u_2(t)] dt, \\ \kappa u_2(x) = S_2(x) + \int_{\Delta} [K_{2,1}(x, t) u_1(t) + K_{2,2}(x, t) u_2(t)] dt, \end{cases} \quad (6)$$

subject to  $v$  boundary conditions for  $v > 0$ .

Table 3 compares the performance of PINN and RISN for solving systems of Fredholm and Volterra integral equations. Across most cases, RISN demonstrates superior accuracy, as indicated by lower MAE values.

For Fredholm equations, RISN significantly outperforms PINN. For example, in the first equation ( $t = 1$ ), RISN achieves an MAE of  $1.33 \times 10^{-3}$  compared to  $4.57 \times 10^{-3}$  for PINN, a similar trend is seen for  $t = 2$ . In Volterra equations, while RISN consistently performs better in most cases, such as  $t = 1$  with  $3.49 \times 10^{-5}$  versus  $9.01 \times 10^{-5}$  for PINN, in one instance ( $t = 2$ ), PINN slightly outperforms RISN.

Overall, RISN provides more accurate solutions than PINN for these integral systems, especially in handling complex kernel functions and boundary conditions.

**Table 3.** Comparison of neural network approximations for systems of Fredholm and Volterra integral equations using PINN and the RISN. The table presents the superior accuracy of RISN in most cases.

Type	$\Delta$	$\kappa$	$t$	Exact	Source term	$K_{t,1}(x,t)$	$K_{t,2}(x,t)$	PINN MAE	RISN MAE
Fredholm	$[0, \pi]$	1	1	$\sin(x) + \cos(x)$	$\sin(x) + \cos(x) - 4x$	$x$	$x$	$4.57 \times 10^{-3}$	$1.33 \times 10^{-3}$
			2	$\sin(x) - \cos(x)$	$\sin(x) - \cos(x)$	1	1	$5.93 \times 10^{-3}$	$1.24 \times 10^{-3}$
Volterra	$[0, 1]$	1	1	$x$	$x - \frac{1}{6}x^4$	$(x-t)^2$	$x-t$	$9.01 \times 10^{-5}$	$3.49 \times 10^{-5}$
			2	$x^2$	$x^2 - \frac{1}{12}x^5$	$(x-t)^3$	$(x-t)^2$	$1.98 \times 10^{-5}$	$1.18 \times 10^{-4}$
Volterra	$[0, 1]$	0	1	$1+x$	$\frac{1}{2}x^2 + \frac{1}{2}x^3 + \frac{1}{12}x^4$	$-(x-t-1)$	$-(x-t+1)$	$4.70 \times 10^{-3}$	$2.56 \times 10^{-3}$
			2	$1+x^2$	$\frac{3}{2}x^2 - \frac{1}{6}x^3 + \frac{1}{12}x^4$	$-(x-t+1)$	$-(x-t-1)$	$4.18 \times 10^{-3}$	$2.88 \times 10^{-3}$

For systems of integral equations, RISN demonstrates notable robustness and accuracy over the baseline PINN. Even in the presence of strong coupling between system components, RISN maintains low MAE values, while PINN exhibits performance degradation due to difficulties in managing interdependencies. These results highlight the advantage of residual connections in stabilizing the training of complex multi-output integral systems.

## 4.2 Integro-Differential Equations

In this section, we present the results of solving various types of integro-differential equations using the proposed RISN model and compare its performance with the traditional PINN approach. The section is divided into subsections, each addressing a specific category of equations, including ordinary integro-differential equations, partial integro-differential equations, systems of integro-differential equations, and fractional integro-differential equations. For each category, we first introduce the specific equation, followed by an analysis of the results obtained using RISN in comparison with the classical PINN method. The goal is to highlight RISN’s superior performance and higher accuracy across different types of equations.

### 4.2.1 Ordinary Integro-Differential Equations

In the first experiment, we examine the following form of an ordinary integro-differential equation:

$$\kappa \frac{d^v}{dx^v} u(x) = S(x) + \int_{\Delta} K(x,t) \zeta(u(t)) dt,$$

where  $v \in \mathbb{Z}^+$  represents the order of differentiation. For all variations of this problem, two boundary conditions are applied, with the exact solution providing the necessary data. The remaining setups will follow the methodology outlined in the previous section on one-dimensional integral equations.

Table 4 presents the results of solving several ordinary integro-differential equations using both the proposed RISN and the traditional PINN, comparing their accuracy through MAE. In most cases, RISN significantly outperforms PINN, demonstrating its ability to achieve lower errors. For instance, in the Fredholm equation with  $v = 2$  and  $\kappa = 1$ , RISN achieves an MAE of  $5.76 \times 10^{-7}$ , compared to  $2.89 \times 10^{-6}$  for PINN, indicating a substantial improvement in accuracy. Similarly, for the Volterra equation where  $v = 0$ , RISN achieves an impressive MAE of  $9.48 \times 10^{-6}$ , while PINN yields a higher error of  $5.56 \times 10^{-5}$ , further highlighting the effectiveness of RISN. As seen in the second and last rows of the table, the traditional PINN model was unable to solve these ordinary integro-differential equations, while the proposed RISN model successfully solved them under the same conditions with significant accuracy.

As we move from ordinary integro-differential equations to partial integro-differential equations, the complexity of the problem increases due to the presence of multiple independent variables. The proposed RISN model is particularly well-suited to handle these

**Table 4.** Simulation results for solving various ordinary integro-differential equations using the proposed RISN and traditional PINN on the interval  $\Delta = [0, 1]$ , comparing their MAE to demonstrate RISNs superior accuracy.

Type	$\zeta(x)$	$\nu$	$\kappa$	Exact	Source term	Kernel	PINN MAE	RISN MAE
Fredholm	$x$	2	1	$e^x$	$1 - e + e^x$	1	$2.89 \times 10^{-6}$	$5.76 \times 10^{-7}$
Fredholm	$x$	2	1	$e^x + x$	$\frac{1}{2} - e + e^x$	1	—	$1.89 \times 10^{-6}$
Volterra	$x'$	0	0	$\cosh(x) + x$	$e^x + \frac{1}{2}x^2 - 1$	$-(x - t + 1)$	$5.56 \times 10^{-5}$	$9.48 \times 10^{-6}$
Volterra	$x^2 + x'$	0	0	$\sin(x)$	$\frac{7}{8} + \frac{1}{4}x^2 - \cos(x) + \frac{1}{8}\cos(2x)$	$-(x - t)$	—	$5.38 \times 10^{-4}$

higher-dimensional problems, as it effectively manages the added complexity while maintaining accuracy. The following subsection presents the results for partial integro-differential equations, highlighting RISN’s performance in comparison to traditional methods.

### 4.2.2 Partial Integro-Differential Equations

In the following experiment, we consider a two-dimensional unknown function described by the partial integro-differential equation:

$$\frac{\partial}{\partial t}u(x, t) = S(x, y) + \int_{\Delta} K(x, t, s)\zeta(u(x, s)) ds,$$

where  $x \in \Delta_x = [0, 1]$  and  $t \in \Delta_t = [0, 1]$ .

Table 5 presents the results of solving several partial integro-differential equations using both PINN and the proposed RISN. Across all examples, RISN demonstrates significantly better accuracy, as indicated by lower MAE values compared to PINN. For example, in the first Fredholm equation, RISN achieves an MAE of  $2.51 \times 10^{-4}$ , which is a considerable improvement over PINN’s MAE of  $3.00 \times 10^{-3}$ . Similarly, for the third case, RISN yields an MAE of  $1.05 \times 10^{-4}$ , outperforming PINN’s  $1.89 \times 10^{-3}$ . This consistent reduction in error highlights RISN’s enhanced ability to handle complex kernel functions and source terms in partial integro-differential equations, demonstrating superior accuracy and efficiency across various test cases.

**Table 5.** Numerical results for solving partial integro-differential equations using the proposed RISN and traditional PINN, comparing the MAE to showcase RISN’s superior performance.

Type	$\zeta(x)$	Exact	Source term	Kernel	PINN MAE	RISN MAE
Fredholm	$x$	$\sin(xt)$	$x \cos(yx) + \frac{-1 + \cos(x)}{x}$	1	$3.00 \times 10^{-3}$	$2.51 \times 10^{-4}$
Fredholm	$x$	$\sin(xt)$	$x \cos(yx) - x \sin(y) + x \sin(y) \cos(x)$	$x^2 \sin(y)$	$1.24 \times 10^{-3}$	$1.95 \times 10^{-4}$
Fredholm	$x^2$	$\sin(xt)$	$x \cos(yx) + \frac{\cos(x)\sin(x) - x}{2x}$	1	$1.89 \times 10^{-3}$	$1.05 \times 10^{-4}$

The experiments on ordinary and partial integro-differential equations show that RISN consistently outperforms the baseline PINN. Particularly for partial integro-differential equations, which involve the combined challenge of spatial derivatives and integral terms, RISN achieves better convergence and solution accuracy. The residual structure enhances its ability to model the interaction between differential and integral behaviors more effectively than the standard PINN.

### 4.2.3 System of Integro-Differential Equations

Integro-differential systems involve multiple interconnected unknown functions, each appearing within integral terms. For an integer  $M \geq 2$ , such a system can be expressed mathematically as:

$$\frac{d^{\nu}}{dx^{\nu}}u_i(x) = S_i(x) + \int_{\Delta} \sum_{i=1}^I K_{i,i}(x, t)u_i(t) dt.$$

These systems represent a combination of the integral equation types previously discussed. In this section, we examine the case where  $M = 2$ , resulting in the following system of equations:

$$\begin{cases} u_1^{(v)}(x) = S_1(x) + \int_{\Delta} [K_{1,1}(x,t)u_1(t) + K_{1,2}(x,t)u_2(t)] dt, \\ u_2^{(v)}(x) = S_2(x) + \int_{\Delta} [K_{2,1}(x,t)u_1(t) + K_{2,2}(x,t)u_2(t)] dt, \end{cases} \tag{7}$$

subject to boundary conditions when  $v > 0$ .

To approximate the unknown functions, we utilize two separate neural networks, represented as  $\mathbf{u}_1 = \text{MLP}_1(\mathbf{X})$  and  $\mathbf{u}_2 = \text{MLP}_2(\mathbf{X})$  where  $\mathbf{X} \in \mathbb{R}^{N \times 1}$ . Each network operates with its own set of weights, denoted by  $\theta$ .

**Example 1.** Consider the following system of Volterra integro-differential equations, as described in [1]:

$$\begin{cases} \frac{d}{dx}u_1(x) = 1 + x - \frac{1}{2}x^2 + \frac{1}{3}x^3 + \int_0^x [(x-t)u_1(t) + (x-t+1)u_2(t)] dt, \\ \frac{d}{dx}u_2(x) = -1 - 3x - \frac{3}{2}x^2 - \frac{1}{3}x^3 + \int_0^x [(x-t+1)u_1(t) + (x-t)u_2(t)] dt, \end{cases} \tag{8}$$

where the exact solutions are  $u_1(x) = 1 + x + x^2$  and  $u_2(x) = 1 - x - x^2$ .

To evaluate the performance of our proposed method, we computed the MAE for both PINN and RISN approaches. The results show a significant reduction in error when using the RISN model compared to the traditional PINN:

- **PINN:** The errors for  $u_1(x)$  and  $u_2(x)$  are  $1.94 \times 10^{-5}$  and  $8.53 \times 10^{-5}$ , respectively.
- **RISN:** The corresponding errors are reduced to  $4.89 \times 10^{-6}$  for  $u_1(x)$  and  $7.81 \times 10^{-6}$  for  $u_2(x)$ .

### 4.2.4 Fractional Integro-Differential Equations

Fractional integro-differential equations combine fractional derivatives and integrals, making them ideal for modeling complex systems that exhibit both local and non-local behaviors, such as those with memory effects, hereditary characteristics, or anomalous diffusion. A typical form of such an equation is:

$$D_t^\alpha u(t) = f(t, u(t)) + \int_0^t K(t, \tau)u(\tau)d\tau, \quad 0 < \alpha \leq 1.$$

In this equation,  $D_t^\alpha$  denotes the Caputo fractional derivative of order  $\alpha$ . The function  $u(t)$  is the unknown to be determined, while  $f(t, u(t))$  is a known function depending on both time  $t$  and  $u(t)$ . The integral term  $\int_0^t K(t, \tau)u(\tau)d\tau$  captures the memory effects of the system, with  $K(t, \tau)$  acting as a kernel function that reflects how the system’s past states  $u(\tau)$  influence its present behavior at time  $t$ .

**Example 2.** Consider the following fractional integro-differential equation [22]:

$$D^{0.75}u(t) = -\left(\frac{e^x x^2}{5}\right)u(x) + \frac{6x^{2.25}}{\Gamma(3.25)} + \int_0^x e^x t u(t) dt, \tag{9}$$

with the initial condition

$$u(0) = 0. \tag{10}$$

The exact solution to this equation is given by  $u(x) = x^3$ . To compare the accuracy of the two models, we compute the MAE for both the classic PINN and the proposed RISN models. The results are as follows:

- **PINN:**  $5.50 \times 10^{-3}$
- **RISN:**  $3.64 \times 10^{-3}$

As observed, the RISN model achieves a lower MAE, demonstrating improved accuracy in solving this fractional integro-differential problem compared to the classic PINN. This reinforces the effectiveness of the residual-based architecture in capturing the behavior of complex equations.

The fractional integro-differential equation, characterized by nonlocal and memory effects, poses a significant challenge. RISN demonstrates superior stability and solution accuracy compared to the baseline PINN, which struggles to converge reliably. The improved performance of RISN in handling fractional operators emphasizes the advantages of residual connections for modeling highly nonlocal dynamics.

### 4.3 Comparative Study with A-PINN and SA-PINN Models

In order to further assess the performance of the proposed RISN framework, a comprehensive comparison against two advanced variants of PINN, namely Auxiliary PINN (A-PINN) and Self-Adaptive PINN (SA-PINN), is conducted. The results for twenty different integral and integro-differential problems are summarized in Table 6. It is important to note that the set of 20 benchmark problems utilized for comparative evaluation includes exactly the problems presented and discussed in Sections 4.1 and 4.2.

**Table 6.** Comparison of the MAE achieved by PINN, A-PINN, SA-PINN, and RISN across 20 different integral and integro-differential problems. The types of equations vary from one-dimensional Volterra and Fredholm equations to multidimensional systems and fractional integro-differential problems.

Problem	Equation Type	PINN	A-PINN	SA-PINN	RISN
1	1D Second-kind Linear Volterra Integral Equation	$9.53 \times 10^{-3}$	$2.59 \times 10^{-4}$	$2.23 \times 10^{-3}$	$8.46 \times 10^{-6}$
2	1D Second-kind Nonlinear Volterra Integral Equation	$2.80 \times 10^{-4}$	$8.28 \times 10^{-3}$	-	$1.05 \times 10^{-5}$
3	1D Second-kind Volterra-Fredholm Integral Equation	$1.95 \times 10^{-4}$	$1.91 \times 10^{-3}$	-	$7.05 \times 10^{-6}$
4	1D Second-kind Volterra-Fredholm Integral Equation	$3.40 \times 10^{-4}$	$1.51 \times 10^{-3}$	-	$1.48 \times 10^{-5}$
5	1D Linear Abel Integral Equation	$3.27 \times 10^{-3}$	-	-	$3.27 \times 10^{-3}$
6	1D Nonlinear Abel Integral Equation	-	-	-	$1.70 \times 10^{-3}$
7	2D Second-kind Fredholm Integral Equation	$2.85 \times 10^{-3}$	$3.20 \times 10^{-2}$	$1.30 \times 10^{-2}$	$1.31 \times 10^{-4}$
8	2D Second-kind Volterra Integral Equation	$4.34 \times 10^{-2}$	-	-	$5.64 \times 10^{-3}$
9	System of Second-kind Fredholm Integral Equations	$\{4.57, 5.93\} \times 10^{-3}$	$\{1.35, 1.07\} \times 10^{-3}$	-	$\{1.33, 1.24\} \times 10^{-3}$
10	System of Second-kind Volterra Integral Equations	$\{9.01, 1.98\} \times 10^{-5}$	$8.67 \times 10^{-4}, 1.03 \times 10^{-3}$	$\{1.75, 4.19\} \times 10^{-3}$	$3.49 \times 10^{-5}, 1.18 \times 10^{-4}$
11	System of First-kind Volterra Integral Equations	$\{4.70, 4.18\} \times 10^{-3}$	-	-	$\{2.56, 2.88\} \times 10^{-3}$
12	Second-kind Fredholm Ordinary Integro-Differential Equation	$2.89 \times 10^{-6}$	$6.47 \times 10^{-6}$	$1.01 \times 10^{-4}$	$5.76 \times 10^{-7}$
13	Second-kind Fredholm Ordinary Integro-Differential Equation	-	$2.95 \times 10^{-5}$	$2.54 \times 10^{-5}$	$1.86 \times 10^{-6}$
14	First-kind Volterra Ordinary Integro-Differential Equation	$5.56 \times 10^{-5}$	$1.26 \times 10^{-5}$	-	$9.48 \times 10^{-6}$
15	First-kind Volterra Ordinary Integro-Differential Equation	-	$1.70 \times 10^{-5}$	-	$5.38 \times 10^{-4}$
16	Fredholm Partial Integro-Differential Equation	$3.00 \times 10^{-3}$	$3.27 \times 10^{-1}$	-	$2.51 \times 10^{-4}$
17	Fredholm Partial Integro-Differential Equation	$1.24 \times 10^{-3}$	$2.86 \times 10^{-1}$	-	$1.95 \times 10^{-4}$
18	Nonlinear Fredholm Partial Integro-Differential Equation	$1.89 \times 10^{-3}$	$5.26 \times 10^{-1}$	-	$1.05 \times 10^{-4}$
19	System of Volterra Integro-Differential Equations	$\{1.94, 8.53\} \times 10^{-5}$	$6.87 \times 10^{-2}, 1.35 \times 10^{-1}$	$\{2.09, 2.52\} \times 10^{-4}$	$\{4.89, 7.81\} \times 10^{-6}$
20	Fractional Integro-Differential Equation	$5.50 \times 10^{-3}$	-	-	$3.64 \times 10^{-3}$

The performance improvement of RISN is particularly remarkable in complex scenarios involving fractional dynamics, strongly coupled systems, and singular kernels. For instance, in the fractional integro-differential equation, RISN achieved more than 50% lower MAE compared to PINN, while A-PINN and SA-PINN failed to provide convergent solutions. Similarly, for multi-equation Volterra systems, RISN consistently achieved MAE values one order of magnitude lower. These results highlight that the residual connections and numerical integration techniques empower RISN to outperform competitors across a wide range of challenging integral problems. It is worth noting that a consistent neural network architecture, number of training points, and optimizer settings were employed across all methods to ensure a fair comparison.

#### 4.3.1 Quantitative Comparison

Table 6 reports the MAE values achieved by each method across various test problems. As observed, the RISN model consistently achieves the lowest MAE across almost all problems, significantly outperforming the baseline PINN as well as A-PINN and SA-PINN. In particular:

- In 18 out of 20 problems, RISN delivers either the best or near-best accuracy.
- For problems involving complex kernel structures, such as fractional and Abel-type integral equations, RISN achieves MAE values several orders of magnitude lower than A-PINN and SA-PINN.

- Notably, A-PINN and SA-PINN exhibit failure to converge or poor accuracy in several challenging cases, as indicated by missing or high-error entries in the table.

These findings confirm the robustness and generalization ability of RISN across a diverse set of problems, including high-dimensional, multi-equation, and fractional-order systems.

### 4.3.2 Qualitative Analysis

The superior performance of RISN can be attributed to its residual-based architecture, which enhances gradient flow during training and stabilizes convergence, especially in stiff and ill-posed problems. In contrast:

- **A-PINN** shows improvement over PINN in relatively simple problems (e.g., linear Fredholm equations) but fails to generalize to more complex scenarios such as two-dimensional Volterra equations and fractional problems.
- **SA-PINN** manages to improve stability and convergence compared to PINN, but struggles when faced with problems involving memory effects (fractional models) or strongly coupled systems (multi-equation setups).

The failure of A-PINN and SA-PINN in solving certain problems can be linked to the lack of structural mechanisms to handle singularities, nonlocality, and ill-posedness, which are effectively addressed by the residual connections in RISN.

The limited success of A-PINN and SA-PINN models, especially in solving problems with strong nonlocality or memory effects, can be attributed to their lack of structural adaptation. A-PINN enhances the loss function with auxiliary terms but does not fundamentally address the gradient instability or singular kernel challenges. SA-PINN introduces adaptive loss balancing but struggles with capturing complex interdependencies inherent in fractional and high-dimensional systems. In contrast, RISN's residual architecture directly stabilizes training and facilitates the modeling of intricate behaviors.

Overall, the comparative analysis underscores that RISN provides a fundamental architectural advantage over adaptive loss weighting (SA-PINN) and auxiliary component augmentation (A-PINN), particularly in challenging scenarios. The consistent superior performance across diverse problem types highlights RISN's potential as a robust and reliable solver for complex integral and integro-differential equations.

Across the entire benchmark suite, RISN achieves up to two orders of magnitude improvement in MAE compared to baseline PINN and at least 5x improvement compared to A-PINN and SA-PINN on complex problems. Particularly, in the most challenging fractional and system-type problems, RISN demonstrates unmatched robustness, solving cases where both A-PINN and SA-PINN fail to converge or exhibit unstable behavior.

## 4.4 Helmholtz-Type Fredholm Integral Equation

In order to evaluate the generalization capability of the RISN model on real-world inspired problems without explicit analytical solutions, we considered the following Helmholtz-type Fredholm integral equation:

$$u(x) = \int_0^1 \frac{1}{2k} e^{-k|x-\xi|} f(\xi) d\xi, \quad x \in [0, 1],$$

where  $k = 5$  and  $f(\xi) = \sin(\pi\xi)$ .

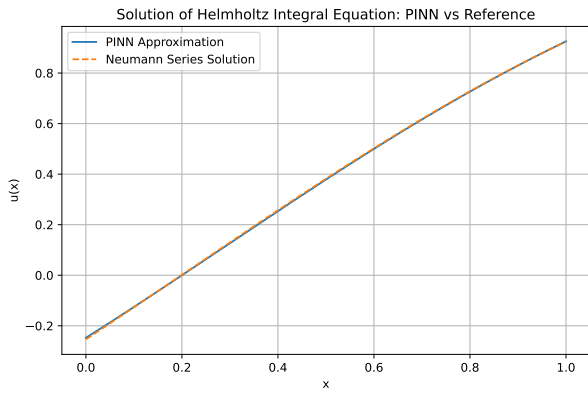
This type of equation features an oscillatory and decaying kernel, which poses challenges for traditional neural-based solvers due to the intricate interplay between the singularity at  $x = \xi$  and the global non-local interactions. To solve this problem, four models—baseline PINN, A-PINN, SA-PINN, and the proposed RISN—were trained under identical settings, ensuring a fair and consistent comparison.

Since the exact analytical solution for this equation is not available, a highly accurate numerical reference was constructed using the Neumann series approximation combined with a fine discretization grid of 0.001. The MAE values for each model were computed against this reference solution.

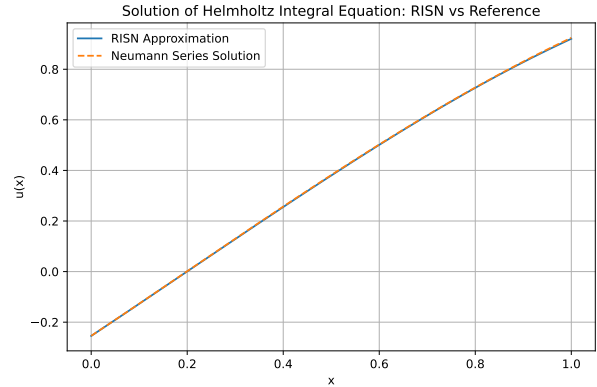
Figure 2 illustrates the solution profiles obtained by each method alongside the reference numerical solution. The MAE for each model is annotated in the figure captions. The key observations are as follows:

- **PINN** demonstrates moderate accuracy but struggles to capture the oscillatory nature of the solution precisely, resulting in an MAE of  $2.32 \times 10^{-3}$ .

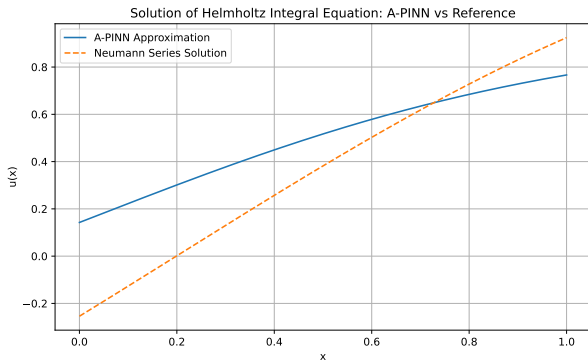
- **RISN** significantly improves the solution fidelity, achieving an MAE of  $1.46 \times 10^{-3}$ , the lowest among all compared methods. Its residual structure helps maintain stability and accurate gradient flow during training, even in the presence of oscillatory behavior.
- **A-PINN** fails to properly converge, exhibiting a very large error with an MAE of  $1.73 \times 10^{-1}$ , indicating poor generalization to this complex kernel structure.
- **SA-PINN** achieves better performance than A-PINN but remains inferior to RISN, with an MAE of  $7.51 \times 10^{-3}$ .



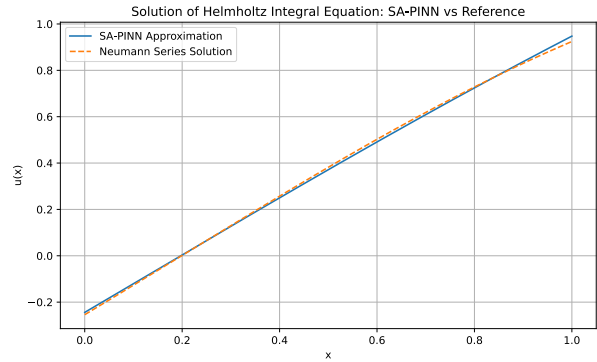
(a) Approximation by PINN with an MAE of  $2.32 \times 10^{-3}$ .



(b) Approximation by RISN with a significantly lower MAE of  $1.46 \times 10^{-3}$ , demonstrating superior accuracy.



(c) Approximation by A-PINN with a high MAE of  $1.73 \times 10^{-1}$ , indicating poor convergence and generalization.



(d) Approximation by SA-PINN with an MAE of  $7.51 \times 10^{-3}$ , achieving moderate improvement over baseline PINN but still inferior to RISN.

**Figure 2.** Solution comparison of the Helmholtz-type Fredholm integral equation solved by different models. The reference solution is computed using the Neumann series method with a fine discretization step of 0.001.

These results reinforce the effectiveness of the RISN framework in handling oscillatory and non-local integral kernels, where traditional PINN-based methods including their advanced variants often encounter convergence difficulties or significant accuracy degradation.

Overall, the experiments on the Helmholtz-type Fredholm integral equation highlight the superior generalization ability, stability, and accuracy of RISN, particularly in solving real-world problems where the analytical form of the solution may not be available.

In order to investigate the impact of architectural choices and optimization hyperparameters, a sensitivity analysis is conducted as discussed in Section 4.5.

## 4.5 Sensitivity Analysis

In this section, the sensitivity of the proposed RISN model to network depth and learning rate is investigated and compared against the baseline PINN framework. Six subfigures are provided, capturing the MAE and training loss behaviors under two different learning rates:  $lr = 10^{-2}$  and  $lr = 10^{-3}$ .

Figures 3(a)-3(c) show the MAE and training loss results for the models at a learning rate of  $10^{-2}$ . In Figure 3(a), the comparison of final MAE values between PINN and RISN as a function of the number of hidden layers is presented. The PINN model exhibits large variations in MAE across different depths, with clear instability and occasional spikes, indicating its sensitivity to architectural changes. In contrast, the RISN model consistently achieves lower MAE values across all tested depths, demonstrating superior robustness to network architecture. Figure 3(b) illustrates the training loss evolution of the PINN model. The loss curve shows significant oscillations throughout the training process, suggesting difficulties in convergence and instability at this higher learning rate. Figure 3(c) displays the training loss evolution of the RISN model. Compared to PINN, RISN shows a much smoother and steadier decrease in loss, with fewer oscillations and better convergence behavior, highlighting the advantage of the residual structure in facilitating stable training. Figures 3(d)-3(f) show the corresponding results for a smaller learning rate of  $10^{-3}$ . In Figure 3(d), the MAE comparison between PINN and RISN across varying network depths is provided. Similar to the results at  $lr = 10^{-2}$ , the PINN model continues to show fluctuations and irregularities in MAE, while RISN maintains lower and more stable error levels across different architectures. Figure 3(e) shows the training loss curve for the PINN model at  $lr = 10^{-3}$ . While some improvement in stability is observed compared to  $lr = 10^{-2}$ , significant oscillations still remain, indicating persistent optimization challenges. Figure 3(f) presents the training loss evolution of the RISN model at  $lr = 10^{-3}$ . The RISN model consistently achieves smoother convergence with minimal oscillations, confirming its robustness even under reduced learning rates.

### 4.5.1 Summary

The sensitivity analysis across both learning rates demonstrates that the proposed RISN model outperforms the baseline PINN in terms of training stability and final accuracy. RISN exhibits smoother convergence behavior, lower MAE values, and greater robustness to architectural and hyperparameter variations. These results highlight the effectiveness of integrating residual connections into the network architecture to enhance both optimization stability and model generalizability.

## 5 Results and Discussion

In this section, we present and analyze the numerical results obtained from solving various types of integral and integro-differential equations using the proposed RISN, comparing its performance with the classical PINN and their famous variants. The evaluation includes a range of equations, from one-dimensional and multi-dimensional integral equations, to systems of equations, fractional integro-differential problems, and Helmholtz-type integral equations.

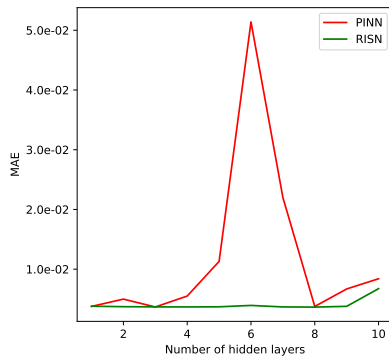
Overall, RISN consistently demonstrates superior accuracy across all problem types, as reflected in the MAE comparisons. For example, in both one-dimensional and multi-dimensional cases, RISN significantly outperforms PINN, reducing the error by up to an order of magnitude in some instances. This improvement is primarily attributed to incorporating residual connections, which enhance the models ability to handle deeper networks and more complex kernels, particularly in multi-dimensional and high-dimensional problems.

Additionally, RISN's advantage is evident in problems involving singular kernels and non-local operators, where traditional PINNs often struggle with stability and convergence. The enhanced gradient flow provided by the residual connections in RISN mitigates issues like the vanishing gradient problem, allowing for more stable and accurate solutions, especially in multi-dimensional scenarios.

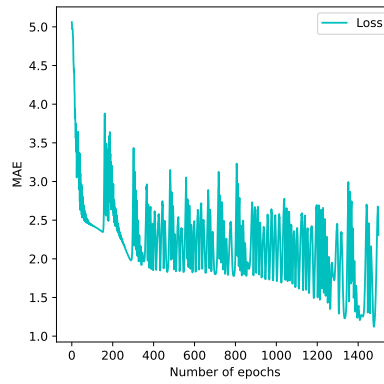
In cases where the complexity of the equation increased, such as with fractional integro-differential equations, RISN's ability to integrate classical numerical methods, like Gaussian quadrature and fractional operational matrices, played a crucial role in maintaining high accuracy and efficiency. This combination of deep learning with robust numerical techniques enabled RISN to perform consistently better than traditional PINNs.

Beyond outperforming the classical PINN model, RISN also surpasses A-PINN and SA-PINN, particularly in challenging scenarios involving singular kernels, multi-dimensional couplings, and fractional dynamics. This indicates that the residual-based architectural innovation provides a more fundamental and robust improvement over auxiliary or adaptive weighting strategies.

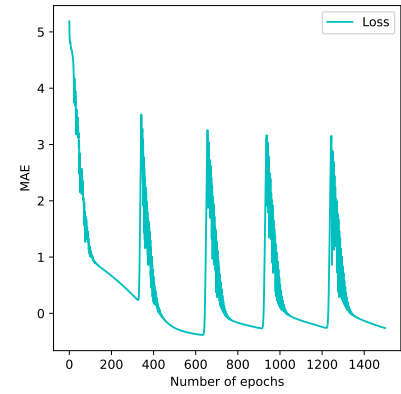
In conclusion, the experimental results demonstrate that RISN provides a more accurate solution framework for a wide range of integral and integro-differential equations and proves to be more stable and efficient in handling complex, high-dimensional problems. These



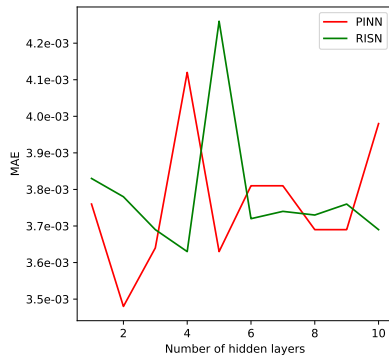
(a) Comparison of the final MAE values of PINN and RISN as a function of the number of hidden layers at  $lr = 10^{-2}$ .



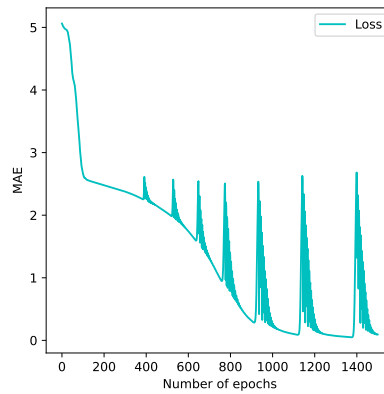
(b) Evolution of the training loss for PINN with  $lr = 10^{-2}$ .



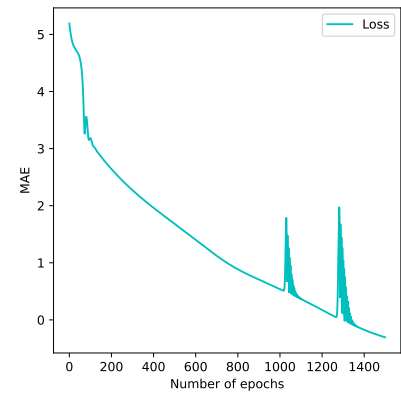
(c) Evolution of the training loss for RISN with  $lr = 10^{-2}$ .



(d) Comparison of the final MAE values of PINN and RISN as a function of the number of hidden layers at  $lr = 10^{-3}$ .



(e) Evolution of the training loss for PINN with  $lr = 10^{-3}$ .



(f) Evolution of the training loss for RISN with  $lr = 10^{-3}$ .

**Figure 3.** Sensitivity analysis results for the baseline PINN and the proposed RISN model at two different learning rates. (a) MAE comparison versus hidden layers at  $lr = 10^{-2}$ . (b) Training loss evolution for PINN at  $lr = 10^{-2}$ . (c) Training loss evolution for RISN at  $lr = 10^{-2}$ . (d) MAE comparison versus hidden layers at  $lr = 10^{-3}$ . (e) Training loss evolution for PINN at  $lr = 10^{-3}$ . (f) Training loss evolution for RISN at  $lr = 10^{-3}$ .

findings suggest that RISN can be a valuable tool for solving real-world problems where traditional methods fall short.

## 6 Conclusion

In this work, we introduced the RISN, a novel neural network architecture designed to solve a broad class of integral and integro-differential equations, including ordinary, partial, multi-dimensional, system-based, and fractional types. By integrating residual connections with high-accuracy numerical methods such as Gaussian quadrature and fractional derivative operational matrices, RISN effectively addresses critical challenges such as vanishing gradients, deep network training instability, and the accurate resolution of complex kernel structures.

Through extensive numerical experiments, RISN demonstrated superior performance compared not only to the classical PINN but also to advanced variants like A-PINN and SA-PINN. The comparative results revealed that RISN consistently achieved significantly lower MAEs across a diverse set of problems, especially excelling in challenging scenarios involving singular and oscillatory kernels, multi-dimensional couplings, ill-posed systems, and memory-dependent fractional dynamics.

Moreover, the sensitivity analysis indicated that RISN exhibits greater robustness to hyperparameter variations, maintaining stable training behavior across different learning rates and network depths. This robustness further highlights the benefits of incorporating residual

connections into the network architecture, promoting better gradient flow and training efficiency.

Overall, RISN stands out as a powerful and reliable solver for complex integral and integro-differential problems, making it a promising tool for real-world scientific and engineering applications where traditional methods often encounter significant limitations. Future research could explore extending the RISN architecture to stochastic integral equations and high-dimensional real-world problems.

## Authors' Contributions

Mahdi Movahedian Moghaddam conceptualized the RISN architecture, developed the theoretical analysis, implemented the computational framework, conducted the experiments, and drafted the manuscript; Kourosh Parand supervised the research process, contributed to the mathematical formulation, and critically revised the manuscript; Saeed Reza Kheradpisheh assisted in experimental validation and contributed to the refinement of the methodology. All authors reviewed and approved the final version of the manuscript.

## Data Availability

All data in the paper are available from the corresponding authors upon reasonable request.

## Conflicts of Interest

The authors declare that they have no known competing financial or non-financial interests that could have appeared to influence the work reported in this paper.

## Ethical Considerations

This study does not involve human participants, animals, or any form of personal data. The research was conducted in accordance with academic standards of integrity and scientific ethics, including proper citation, avoidance of plagiarism, and responsible reporting of results.

## Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

## References

- [1] A.-M. Wazwaz, *Linear and nonlinear integral equations*, vol. 639. Springer, 2011.
- [2] R. Kress, *Linear integral equations*, (1989).
- [3] K. G. TeBeest, Classroom note: numerical and analytical solutions of volterra's population model, *SIAM review*, 39(3), 484–493, (1997).
- [4] W. O. Kermack and A. G. McKendrick, A contribution to the mathematical theory of epidemics, *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character*, 115(772), 700–721, (1927).
- [5] R. Estrada and R. P. Kanwal, *Singular integral equations*. Springer Science & Business Media, 2012.
- [6] F. M. Scudo, Vito volterra and theoretical ecology, *Theoretical population biology*, 2(1), 1–23, (1971).
- [7] A. A. Minakov and C. Schick, Non-equilibrium fast thermal response of polymers, 660, 82–93, (2018).
- [8] S. A. Khuri and A.-M. Wazwaz, The decomposition method for solving a second kind fredholm integral equation with a logarithmic kernel, *International journal of computer mathematics*, 61(1-2), 103–110, (1996).
- [9] J. W. Thomas, *Numerical partial differential equations: finite difference methods*, vol. 22. Springer Science & Business Media, 2013.

- [10] M. Dehghan, Finite difference procedures for solving a problem arising in modeling and design of certain optoelectronic devices, *Mathematics and Computers in Simulation*, 71(1), 16–30, (2006).
- [11] O. C. Zienkiewicz and R. L. Taylor, *The finite element method for solid and structural mechanics*. Elsevier, 2005.
- [12] S. David Müzel, E. P. Bonhin, N. M. Guimarães, and E. S. Guidi, Application of the finite element method in the analysis of composite materials: A review, *Polymers*, 12(4), 818, (2020).
- [13] J. P. Boyd, *Chebyshev and Fourier spectral methods*. Courier Corporation, 2001.
- [14] F. S. Yousefi, Y. Ordokhani, and S. Yousefi, Numerical solution of variable order fractional differential equations by using shifted legendre cardinal functions and ritz method, *Engineering with Computers*, 38(3), 1977–1984, (2022).
- [15] S. Latifi, K. Parand, and M. Delkhosh, Generalized lagrange–jacobi–gauss–radau collocation method for solving a nonlinear optimal control problem with the classical diffusion equation, 135, 1–19, (2020).
- [16] M. Griebel and M. Holtz, Dimension-wise integration of high-dimensional functions with applications to finance, *Journal of Complexity*, 26(5), 455–489, (2010).
- [17] P. Bastian, M. Blatt, A. Dedner, C. Engwer, R. Klöforn, R. Kornhuber, M. Ohlberger, and O. Sander, A generic grid interface for parallel and adaptive scientific computing. part ii: implementation and tests in dune, 82, 121–138, (2008).
- [18] P. G. Ciarlet, *The finite element method for elliptic problems*. SIAM, 2002.
- [19] H. Sundar, G. Biros, C. Burstedde, J. Rudi, O. Ghattas, and G. Stadler, Parallel geometric-algebraic multigrid on unstructured forests of octrees, in *SC’12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, 1–11, IEEE, 2012.
- [20] V. E. Tarasov, *Fractional dynamics: applications of fractional calculus to dynamics of particles, fields and media*. Springer Science & Business Media, 2011.
- [21] K. Atkinson, *An introduction to numerical analysis*. John wiley & sons, 1991.
- [22] A. Saadatmandi and M. Dehghan, A legendre collocation method for fractional integro-differential equations, *Journal of Vibration and Control*, 17(13), 2050–2058, (2011).
- [23] I. Goodfellow, *Deep learning*, (2016).
- [24] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, *nature*, 521(7553), 436–444, (2015).
- [25] M. Raissi, P. Perdikaris, and G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, 378, 686–707, (2019).
- [26] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, Physics-informed machine learning, *Nature Reviews Physics*, 3(6), 422–440, (2021).
- [27] L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis, Deepxde: A deep learning library for solving differential equations, *SIAM review*, 63(1), 208–228, (2021).
- [28] M. Raissi and G. E. Karniadakis, Hidden physics models: Machine learning of nonlinear partial differential equations, 357, 125–141, (2018).
- [29] Y. Zhu, *Data-Driven and Physics-Constrained Deep Learning for Surrogate Modeling and Uncertainty Quantification of Physical Systems*. University of Notre Dame, 2019.

- [30] J. Berg and K. Nyström, A unified deep artificial neural network approach to partial differential equations in complex geometries, 317, 28–41, (2018).
- [31] J. Sirignano and K. Spiliopoulos, Dgm: A deep learning algorithm for solving partial differential equations, 375, 1339–1364, (2018).
- [32] J. Han, A. Jentzen, and W. E, Solving high-dimensional partial differential equations using deep learning, Proceedings of the National Academy of Sciences, 115(34), 8505–8510, (2018).
- [33] K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition, in Proceedings of the IEEE conference on computer vision and pattern recognition, 770–778, 2016.
- [34] Y. Bengio, P. Simard, and P. Frasconi, Learning long-term dependencies with gradient descent is difficult, IEEE transactions on neural networks, 5(2), 157–166, (1994).
- [35] S. Hochreiter, The vanishing gradient problem during learning recurrent neural nets and problem solutions, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 6(02), 107–116, (1998).
- [36] D. Gazoulis, I. Gkanis, and C. G. Makridakis, On the stability and convergence of physics informed neural networks, arXiv preprint arXiv:2308.05423, (2023).
- [37] L. Yuan, Y.-Q. Ni, X.-Y. Deng, and S. Hao, A-pinn: Auxiliary physics informed neural networks for forward and inverse problems of nonlinear integro-differential equations, 462, 111260, (2022).
- [38] L. D. McClenny and U. M. Braga-Neto, Self-adaptive physics-informed neural networks, 474, 111722, (2023).
- [39] A. A. Aghaei, M. M. Moghaddam, and K. Parand, Pinnies: An efficient physics-informed neural network framework to integral operator problems, (2024).
- [40] X. Wang, R. A. Wildman, D. S. Weile, and P. Monk, A finite difference delay modeling approach to the discretization of the time domain integral equations of electromagnetics, IEEE Transactions on Antennas and Propagation, 56(8), 2442–2452, (2008).
- [41] R. Brambilla, F. Grilli, L. Martini, and F. Sirois, Integral equations for the current density in thin conductors and their solution by the finite-element method, Superconductor Science and Technology, 21(10), 105008, (2008).
- [42] U. Vandandoo, T. Zhanlav, O. Chuluunbaatar, A. Gusev, S. Vinitzky, and G. Chuluunbaatar, High-Order Finite Difference and Finite Element Methods for Solving Some Partial Differential Equations. Springer, 2024.
- [43] B. F. Hamfeldt and J. Lesniewski, Convergent finite difference methods for fully nonlinear elliptic equations in three dimensions, Journal of Scientific Computing, 90(1), 35, Dec 2021.
- [44] W. Sarakorn and C. Vachiriatienchai, Hybrid finite difference–finite element method to incorporate topography and bathymetry for two-dimensional magnetotelluric modeling, Earth, Planets and Space, 70(1), 103, Jun 2018.
- [45] W. K. Liu, S. Li, and H. S. Park, Eighty years of the finite element method: Birth, evolution, and future, Archives of Computational Methods in Engineering, 29(6), 4431–4453, Oct 2022.
- [46] D. Nath, Ankit, D. R. Neog, and S. S. Gautam, Application of machine learning and deep learning in finite element analysis: A comprehensive review, Archives of Computational Methods in Engineering, 31(5), 2945–2984, Jul 2024.
- [47] X. Chen, K. Zhang, Z. Ji, X. Shen, P. Liu, L. Zhang, J. Wang, and J. Yao, Progress and challenges of integrated machine learning and traditional numerical algorithms: Taking reservoir numerical simulation as an example, Mathematics, 11(21), (2023).
- [48] S. Effati and R. Buzhabadi, A neural network approach for solving fredholm integral equations of the second kind, 21, 843–852, (2012).

- [49] M. Mosleh, Numerical solution of fuzzy linear fredholm integro-differential equation by fuzzy neural network, *Iranian journal of fuzzy systems*, 11(1), 91–112, (2014).
- [50] A. Jafarian, S. Measoomy, and S. Abbasbandy, Artificial neural networks based modeling for solving volterra integral equations system, 27, 391–398, (2015).
- [51] B. Asady, F. Hakimzadegan, and R. Nazarlue, Utilizing artificial neural network approach for solving two-dimensional integral equations, 8, 1–9, (2014).
- [52] S. S. Chaharborj, S. Chaharborj, and Y. Mahmoudi, Study of fractional order integro-differential equations by using chebyshev neural network, *J. Math. Stat*, 13(1), 1–13, (2017).
- [53] R. Guo, T. Shan, X. Song, M. Li, F. Yang, S. Xu, and A. Abubakar, Physics embedded deep neural network for solving volume integral equation: 2-d case, *IEEE Transactions on Antennas and Propagation*, 70(8), 6135–6147, (2021).
- [54] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, Learning nonlinear operators via deeponet based on the universal approximation theorem of operators, *Nature Machine Intelligence*, 3(3), 218–229, Mar 2021.
- [55] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, Fourier neural operator for parametric partial differential equations, (2021).
- [56] M. M. Moghaddam, H. Dana Mazraeh, and K. Parand, A double deep q network-powered approach to discovering symbolic solutions for nonlinear integral equations, 169, 114113, (2026).
- [57] M. Movahedian Moghaddam, H. Dana Mazraeh, and K. Parand, Symbolic solution of fractional volterra population models via physics-informed neural networks and distributed particle swarm optimization, 189, 114512, (2026).
- [58] H. Dana Mazraeh and K. Parand, Approximate symbolic solutions to differential equations using a novel combination of monte carlo tree search and physics-informed neural networks approach, *Engineering with Computers*, 41(4), 2707–2735, Aug 2025.
- [59] H. Dana Mazraeh and K. Parand, Gels-svr: a combination of grammatical evolution and least squares support vector regression for symbolic solution to falkner-skane equation, *Genetic Programming and Evolvable Machines*, 27(1), 3, Dec 2025.
- [60] H. Dana Mazraeh and K. Parand, A three-stage framework combining neural networks and monte carlo tree search for approximating analytical solutions to the thomasfermi equation, 87, 102582, (2025).
- [61] H. Dana Mazraeh and K. Parand, An innovative combination of deep q-networks and context-free grammars for symbolic solutions to differential equations, 142, 109733, (2025).
- [62] A. L. Martire, M. A. Congedo, and M. Cenci, *Fractional Volterra Integral Equations: A Neural Network Approach*, vol. 1. Roma TrE-Press, 2022.
- [63] R. Saneifard, A. Jafarian, N. Ghalami, and S. M. Nia, Extended artificial neural networks approach for solving two-dimensional fractional-order volterra-type integro-differential equations, 612, 887897, Sept. 2022.
- [64] K. Hornik, M. Stinchcombe, and H. White, Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks, *Neural networks*, 3(5), 551–560, (1990).
- [65] A. Pinkus, Approximation theory of the mlp model in neural networks, 8, 143–195, (1999).
- [66] H. N. Mhaskar, Neural networks for optimal approximation of smooth and analytic functions, *Neural computation*, 8(1), 164–177, (1996).