International Journal of Mathematical, Engineering and Management Sciences
Vol. 11, No. 4, 1743-1769, 2026
https://doi.org/10.33889/IJMEMS.2026.11.4.071

A Weighted Ensemble Neural Network Method for Singularly Perturbed Parabolic Convection-Diffusion Equations with Robin Boundary Conditions

P. Deepika
Department of Mathematics,
Vellore Institute of Technology, 632014, Vellore, Tamil Nadu, India.
E-mail: deepika.p2021@vitstudent.ac.in

L. Mohana Sundari
Department of Software Systems,
Vellore Institute of Technology, 632014, Vellore, Tamil Nadu, India.
Corresponding author: mohanasundari.l@vit.ac.in

(Received on November 3, 2025; Revised on January 25, 2026 & May 7, 2026 & June 3, 2026; Accepted on June 20, 2026)

Abstract
Parabolic convection diffusion equations with small diffusion effects play a central role in modeling transport dominated processes arising in heat transfer, fluid flow, and mass transport. When diffusion is very small, these models exhibit sharp boundary layers near the spatial boundaries, which significantly complicates numerical approximation and often leads to loss of accuracy for conventional discretization techniques. Motivated by these challenges, this work focuses on a singularly perturbed parabolic convection diffusion model with Robin type boundary conditions, which naturally arise in problems involving convective heat exchange and surface interactions. To address the multiscale nature of the solution, a weighted ensemble neural network approach is proposed. The method combines several feedforward neural networks, each trained to approximate different localized features of the solution, and blends their outputs using smoothly varying spatial weight functions. A purely supervised learning framework is adopted, where training data are obtained from known exact solutions. Initial and boundary conditions are enforced through data based loss terms, while spatial derivatives are used only to ensure consistency at the boundaries. Unlike physics informed neural network approaches, the governing equation is not imposed as a residual constraint during training. By embedding the expected boundary layer behavior into the network structure and employing ensemble weighting, the proposed method accurately captures sharp solution gradients without the need for mesh refinement or domain decomposition. Numerical experiments for a range of perturbation parameters demonstrate that the method delivers stable and accurate approximations and generally outperforms single network models in resolving boundary layer effects.

Keywords- Weighted ensemble neural network, Singular perturbation problems, Parabolic convection–diffusion equations, Robin boundary conditions, Supervised learning.

1. Introduction

Heat transfer processes involving convection and weak diffusion arise frequently in engineering and applied sciences, particularly in the thermal analysis of thin rods, fins, and micro-scale devices. When heat conduction is relatively small compared to convection effects, the governing equations exhibit singular perturbation behavior, leading to the formation of sharp boundary layers near the ends of the spatial domain. Accurate numerical treatment of such problems requires careful consideration of both the convection dominance and the associated boundary conditions.

In this work, we consider a one-dimensional heat transfer model describing the temperature distribution in a thin rod subjected to convective heat transport and heat exchange with the surrounding environment. The temperature evolution is governed by a singularly perturbed parabolic convection-diffusion (SPPCD) equation of the form

$$\frac{\partial u}{\partial t} - \varepsilon \frac{\partial^2 u}{\partial x^2} + a(x,t)\frac{\partial u}{\partial x} + c(x,t)u = f(x,t), \quad (x,t) \in (0,1) \times (0,T] \tag{1}$$

where, $u(x,t)$ denotes the temperature, ε is a small positive diffusion parameter, $a(x,t)$ represents the convective velocity, and $f(x,t)$ is a given heat source term. The smallness of ε reflects low thermal diffusivity or high convective transport, which gives rise to boundary layers in the spatial domain.

The interaction between the rod and the surrounding medium is modeled through Robin boundary conditions (RBCs), which represent convective heat exchange according to Newton's law of cooling. Specifically, the boundary conditions are given by

$$\alpha_0 u(0,t) - \beta_0 u_x(0,t) = g_0(t), \quad \alpha_1 u(1,t) + \beta_1 u_x(1,t) = g_1(t), \quad t \in (0,T],$$

where, the coefficients α_i and β_i characterize the heat transfer properties at the boundaries. Unlike Dirichlet or Neumann conditions, RBC**s** provide a more realistic description of heat exchange processes in practical applications.

The presence of the small diffusion parameter ε combined with RBCs significantly complicates both the analytical and numerical treatment of the problem. Standard numerical schemes often fail to capture the boundary layer behavior uniformly with respect to ε, resulting in loss of accuracy for small values of the perturbation parameter. This motivates the development of techniques capable of resolving the sharp gradients induced by convection-dominated heat transfer.

Motivating through such application as in Equation (1), this work focuses on analyzing a singularly perturbed problem (SPP) defined over the domain $\Omega = \Omega_x \times \Lambda_t$, where, $\Omega_x = (0,1)$ denotes the spatial interval and $\Lambda_t = (0,T]$ represents the temporal range. ε satisfies $0 < \varepsilon \ll 1$. The boundary of the space-time domain Ω is denoted by $\Gamma = \Gamma_x \cup \Gamma_0 \cup \Gamma_1$, where $\Gamma_0 = \{(0,t): t \in [0,T]\}$ and $\Gamma_1 = \{(1,t): t \in [0,T]\}$ correspond to the lateral boundaries, while $\Gamma_x = \{(x,0): x \in [0,1]\}$ defines the initial boundary. The term *lateral boundaries* refer specifically to the spatial boundaries of the domain over the time interval $(0,T]$. More precisely, the lateral boundaries correspond to the sets

$$x = 0 \times (0,T] \; and \; x = 1 \times (0,T],$$

which represent the left and right spatial boundaries evolving in time. These are distinguished from the *initial boundary* $(0,1) \times \{t = 0\}$, where the initial condition is prescribed.

Consider the following problem (Gelu and Duressa, 2022):

$$L_\varepsilon u(x,t) \equiv \left(\varepsilon \frac{\partial^2 u}{\partial x^2} + b(x,t)\frac{\partial u}{\partial x} - c(x,t)u - \frac{\partial u}{\partial t}\right)(x,t) = f(x,t), \; (x,t) \in \Omega,$$

$$u(x,0) = s(x), \; (x,0) \in \Gamma_x,$$

$$B_{0,\varepsilon} u(0,t) \equiv \left(u - \varepsilon \frac{\partial u}{\partial x}\right)(0,t) = q_0(t), \; (0,t) \in \Gamma_0,$$

$$B_{1,\varepsilon} u(1,t) \equiv \left(u + \varepsilon \frac{\partial u}{\partial x}\right)(1,t) = q_1(t), \; (1,t) \in \Gamma_1.$$

It is assumed that the functions $b(x,t)$, $c(x,t)$, and $f(x,t)$, as well as the initial condition $s(x)$ and boundary data $q_0(t)$, $q_1(t)$, are smooth and bounded. Additionally, $b(x,t) \geq \beta > 0$, and the reaction term satisfies $c(x,t) \geq 0$, which ensures the well-posedness and uniqueness of solution to problem defined above. The initial data is further assumed to have higher-order derivatives that vanish appropriately. For

$b(x,t) > 0$, a boundary layer is expected near x=0, whereas for $b(x,t) < 0$, the layer forms near $x = 1$.

As ε becomes smaller, the solution of the problem develops a boundary layer near the inflow boundary Γ_0. This highlights the significant influence of ε on the behavior of the solution near the boundary.

Historically, numerous numerical techniques have been developed to address the difficulties associated with SPPs. Early approaches often employed uniform meshes, which proved inadequate for capturing sharp gradients without excessive computational cost. To overcome this, layer-adapted meshes like Shishkin and Bakhvalov meshes were introduced, providing finer discretization near layers and coarser grids elsewhere. Kadalbajoo and Patidar (2006) developed parameter-uniform numerical methods using fitted mesh techniques to handle boundary layers effectively. Similarly, Kadalbajoo and Gupta (2010) proposed robust numerical schemes that maintain uniform convergence irrespective of the perturbation parameter's magnitude. In Temel and Çakır (2024), the authors proposed an efficient numerical approach for solving singularly perturbed nonlinear ordinary differential equations with integral boundary conditions. These methods often involve sophisticated mesh generation and require prior knowledge of layer locations. In this work, Prince et al. (2025) developed a uniformly convergent numerical scheme for singularly perturbed Fredholm integro-differential equations using a non-standard finite difference method and Richardson extrapolation, achieving higher-order accuracy validated by numerical examples.

Other significant contributions include Linß (2007), who investigated the uniform convergence of upwind difference schemes for coupled systems, and O'Riordan and Stynes (2009), who studied upwind schemes for singularly perturbed strongly coupled ordinary differential equations. Liu and Chen (2015) introduced mesh equidistribution techniques to achieve ε-uniform convergence in singularly perturbed systems.

Although fitted-mesh techniques such as Shishkin and Bakhvalov-type methods have achieved remarkable success in obtaining parameter-uniform approximations for SPPs, their implementation often relies on problem-specific mesh construction and detailed knowledge of boundary-layer characteristics. In recent years, the rapid development of machine learning and deep neural networks has motivated the investigation of data-driven approaches for solving differential equations. In particular, neural-network-based methods offer a mesh-free approximation framework capable of representing highly complex solution structures. This has led to growing interest in applying machine learning techniques, including Physics-Informed Neural Networks (PINNs), SPPs where traditional numerical methods may require sophisticated layer-adapted discretizations. In recent years, the emergence of machine learning techniques has introduced innovative approaches to solving differential equations. PINNs, introduced by Raissi et al. (2019) allows for mesh-free solutions and has shown promise in various applications, including fluid mechanics, quantum mechanics, and material science. Basha and Shanthi (2015) proposes a numerical approach for coupled convection-diffusion problems with RBCs and discontinuous sources. Shan and Li (2024) proposed asymptotic PINNs that combine prior knowledge from asymptotic analysis with neural networks to solve SPPs effectively. Cao et al. (2023) implemented a parameter asymptotic strategy in PINNs to learn singularly perturbed convection-dominated problems. These studies demonstrate the versatility of PINNs in handling complex differential equations. However, challenges remain, particularly in capturing sharp boundary layers and ensuring convergence for small ε values.

Applying PINNs to SPPCD equations with RBCs presents unique challenges. They are: standard PINNs may struggle to resolve sharp gradients without adaptive sampling or specialized architectures; incorporating mixed boundary conditions into the loss function requires accurate estimation of both function values and derivatives. While PINNs have demonstrated considerable promise for solving differential equations, their application to singularly perturbed convection-diffusion problems remain

challenging due to the presence of sharp boundary layers and highly localized solution gradients. The proposed Weighted Ensemble Neural Network (WENN) framework is specifically designed to address the approximation difficulties associated with such localized boundary-layer structures by employing specialized subnetworks and smooth Gaussian blending. In particular, the method aims to improve the representation of steep solution variations that are often difficult for a single global neural network to capture accurately. However, it should be noted that the present framework does not address all challenges associated with PINNs, such as training directly from partial differential equation (PDE) residuals without solution data or eliminating optimization difficulties inherent in neural network training. Instead, the primary focus of this work is to investigate whether a weighted ensemble architecture can provide improved approximation accuracy for singularly perturbed problems under a supervised learning setting. The proposed WENN framework specifically aims to improve the approximation of sharp boundary-layer structures and reduce the training difficulties associated with highly localized gradients. Liu et al. (2024) proposed adaptive sampling methods to improve PINN performance in multi-scale problems. Wu et al. (2023) introduced residual-based adaptive refinement strategies to enhance PINN accuracy in SPPs. Yadav and Rai (2020) develop a parameter-uniform numerical method for parabolic turning point problems with twin boundary layers. Beguinet et al. (2023) conducted an exhaustive numerical study comparing PINNs with classical numerical approach for convection-diffusion SPPs. These advancements highlight the ongoing efforts to enhance PINN capabilities for complex differential equations. Rao and Chawla (2020) address weakly coupled convection-diffusion equations with RBCs and discontinuous sources. Kumar et al. (2023) analyzes the convergence of a domain decomposition method for SPPs under RBCs. Kumar and Deswal (2022) utilizes wavelet-based methods to approximate solutions of two-parameter SPPs with RBCs. Chawla et al. (2023) presents a fitted mesh numerical method for coupled systems exhibiting boundary and internal layers under RBCs. Gupta et al. (2023) presents a numerical method using hybrid spline difference schemes on adaptive meshes to solve parabolic reaction–diffusion SPPs with RBCs. The insights into handling boundary layers and Robin conditions can inform the design of neural network architectures for similar problems. Yadav and Mukherjee (2024) develop numerical methods for semi linear parabolic PDEs with convection-diffusion characteristics. Berrone et al. (2023) compares methods to enforce Dirichlet boundary conditions in PINNs, highlighting techniques that lead to more efficient solvers. Ahmed (2024) presents a high-accuracy method for coupled convection-diffusion equations under RBCs. Omkar and Phaneendra (2022) developed a numerical method based on exponential splines to solve singularly perturbed delay differential equations with large delays.

Recent developments in PINNs have focused on improving their performance for SPPs characterized by sharp boundary layers. For instance, Wang et al. (2024) proposed a modified PINN architecture incorporating asymptotic expansions to better capture boundary-layer behavior. Similarly, Gie et al. (2024) introduced semi-analytic PINN approaches to enhance solution accuracy for SPPs.

Several studies have identified fundamental limitations of standard PINNs in such settings. In particular, PINNs suffer from training stiffness and difficulty in resolving sharp gradients associated with small perturbation parameters (Sharma et al., 2023). To address this, recent approaches have incorporated asymptotic strategies and pre-training mechanisms, such as the parameter-asymptotic PINN and multistep asymptotic PINN frameworks, which improve convergence but increase model complexity (Cao et al., 2024).

In addition, large-scale benchmarking studies have emphasized that no single neural network approach performs well across all PDE classes, highlighting the need for problem-specific architectures (Wei et al., 2024).

Recent years have witnessed significant progress in the development of highly accurate numerical techniques for differential equations involving Robin and mixed boundary conditions. In particular, Ahmed has contributed several efficient spectral and operational matrix based approaches for solving a variety of boundary value problems. In Ahmed (2023), a highly accurate numerical method was proposed for solving boundary value problems subject to RBCs, demonstrating improved stability and accuracy compared to conventional approaches. Later, Ahmed (2025) developed new Jacobi-Galerkin operational matrices for fractional-order nonlinear two-point boundary value problems with RBCs, showing the effectiveness of spectral techniques for fractional models. More recently, Ahmed (2026a) introduced a Laguerre-Galerkin operational matrix framework for the fractional Bagley-Torvik equation with variable coefficients and RBCs, achieving high-order accuracy. In another study, Ahmed (2026c) proposed an accurate computational method for Poisson and Helmholtz equations with RBCs, highlighting the robustness of the approach for elliptic problems. Furthermore, an advanced numerical framework for Fredholm-Volterra integro-differential equations with mixed boundary conditions was developed in Ahmed (2026b), extending the applicability of operational matrix techniques to more complex integro-differential models.

In contrast to these approaches, the proposed WENN framework avoids PDE residual minimization and instead embeds the boundary-layer structure directly into the model through a weighted ensemble formulation. This leads to a simpler and more stable training process while maintaining high accuracy for SPPs. These studies demonstrate the growing interest in developing accurate and stable numerical methodologies for problems involving RBCs. However, most existing approaches are based on classical discretization or spectral frameworks and require carefully constructed basis functions and matrix formulations.

In this study, we develop a robust numerical framework termed the WENN for solving SPPCD equations with RBCs. The proposed approach is designed to accurately capture sharp boundary layers by combining multiple feedforward neural subnetworks, each responsible for approximating localized solution features. These subnetworks are smoothly blended using spatially dependent Gaussian weight functions, yielding a globally accurate approximation without explicit domain decomposition or adaptive meshing.

In contrast to PINNs, which incorporate the governing PDE residual into the training objective, the proposed WENN framework adopts a supervised learning strategy. The training dataset is generated from available analytical solutions, and the network parameters are optimized by directly minimizing the discrepancy between the predicted and reference solution values. Initial and RBCs are enforced through data-consistency loss terms, where spatial derivatives appearing in the Robin conditions are evaluated using automatic differentiation. These derivative evaluations are used solely to compute boundary mismatches and do not constitute enforcement of the governing PDE.

The proposed WENN framework is fundamentally different from conventional PINNs because the governing PDE residual is not incorporated into the training objective. Instead, the model is trained using supervised data generated from known analytical solutions. Nevertheless, automatic differentiation is employed to evaluate derivative terms appearing in the RBCs, thereby incorporating limited physical information associated with boundary enforcement. Consequently, the proposed approach is more appropriately viewed as a supervised data-driven framework with physics-aware boundary treatment rather than a fully PINN.

Although the proposed WENN framework shares high-level similarities with mixture-of-experts (MoE), domain decomposition neural networks, and enriched PINNs, it differs fundamentally in the following aspects:

i) Classical MoE architectures (Nguyen and Chamroukhi, 2018) employ a trainable gating network to adaptively route inputs to individual expert networks. This gating mechanism introduces additional trainable parameters and can lead to training instability, especially in the small-data regime. By contrast, the proposed WENN uses analytically prescribed, fixed Gaussian weight functions centered at pre-determined spatial locations. The weighting is entirely non-trainable and depends only on the spatial coordinate x, not on the network output or any learned representation. This design eliminates the overhead of gating network training, guarantees smooth blending across the domain, and avoids the winner-takes-all degeneracy often observed in MoE training.
ii) Domain decomposition approaches partition the computational domain into non-overlapping or minimally overlapping subdomains and enforce interface continuity conditions (e.g., matching of function values and derivatives at interfaces) as additional loss terms. This requires explicit interface identification and adds interfacial constraints that can be difficult to satisfy for stiff problems. WENN deliberately avoids hard domain decomposition. Instead, it introduces soft spatial localization through smooth Gaussian weights with fixed centers at $\mu_1 = 0.25$ and $\mu_2 = 0.75$, with $\sigma = 0.15$ controlling the spread. Subnetworks contribute smoothly across the entire domain; there are no artificial interfaces, no interface loss terms, and no explicit partitioning. This design is significantly simpler and avoids the conditioning issues associated with interface enforcement.
iii) Asymptotic and enriched PINN approaches (e.g., Gie et al., 2024; Shan and Li, 2024) incorporate boundary-layer structure by enriching the network with corrector functions derived from singular perturbation asymptotics, but they continue to enforce the governing PDE residual in the loss function. This residual-based training is known to suffer from loss imbalance and stiffness for small epsilon. The proposed WENN also embeds asymptotic boundary-layer structure analytically (via the exponential term $e^{-\alpha(1-x)/\varepsilon}$), but crucially, it does not include any PDE residual in the loss. Training is purely supervised using exact solution data, relying only on initial condition and RBC misfits as loss components. This purely data-driven strategy is conceptually and computationally distinct from enriched PINNs and eliminates the numerical stiffness associated with residual-based methods for SPPs.

These three distinguishing features - fixed non-trainable Gaussian weighting, absence of domain decomposition and interface constraints, and a purely supervised (non-residual) training strategy - constitute the core novelty of the proposed WENN framework.

Although ensemble neural networks, MoE models, and localized neural architectures have been extensively studied in the machine-learning literature, the proposed WENN framework differs in both motivation and construction. The primary objective of WENN is not merely to combine multiple subnetworks, but to develop a boundary-layer-aware approximation framework tailored to SPPCD problems with RBCs. **Table 1** summarizes the main methodological differences between the proposed approach and related neural-network-based methodologies.

As shown in **Table 1**, the proposed WENN framework combines localized subnetworks through predefined Gaussian blending functions without introducing trainable gating networks or interface constraints. Furthermore, unlike conventional PINNs, the governing PDE residual is not included in the loss function, while boundary-layer behavior is explicitly incorporated into the approximation architecture. These characteristics distinguish WENN from existing ensemble and domain-decomposition neural-network approaches and motivate its application to singularly perturbed convection-diffusion problems.

Table 1. Comparison of the proposed WENN framework with related neural-network-based approaches.

Feature	Standard neural network	PINN	MoE	Localized / Domain-decomposition	Proposed WENN
Multiple subnetworks	No	No	Yes	Yes	Yes
Spatial specialization	No	Limited	Yes	Yes	Yes
Trainable gating network	No	No	Yes	No	No
Gaussian weighting/blending	No	No	Rarely	No	Yes
PDE residual in loss function	No	Yes	Optional	Often	No
Exact solution training	Optional	No	Optional	Optional	Yes
Automatic differentiation for boundary terms	No	Yes	Optional	Optional	Yes
Interface continuity constraints	No	No	No	Usually required	No
Layer-adapted mesh required	No	No	No	Sometimes	No
Explicit boundary-layer enrichment	No	Limited	Limited	Partial	Yes
Designed specifically for SPPs	No	Limited	No	Partial	Yes
RBC treatment	Problem dependent	Via residual terms	Problem dependent	Problem dependent	Explicitly incorporated
Computational complexity	Low	Moderate–High	High	Moderate–High	Moderate
Main objective	General approximation	Physics-constrained learning	Expert specialization	Regional decomposition	Boundary-layer-aware approximation
Present work's contribution	–	Baseline comparison	Related architecture	Related architecture	Gaussian-weighted ensemble architecture for SPPCD equations with RBCs

The effectiveness of the proposed WENN framework is demonstrated through numerical experiments for a range of small perturbation parameters. The results confirm that the method accurately resolves boundary layers and achieves stable, high-quality approximations, making it a reliable supervised neural-network-based alternative for solving SPPCD equations with RBCs.

The paper is structured as follows: Section 2 presents the heuristic motivation underlying the proposed WENN framework. Section 3 details the proposed numerical method, including the formulation of the WENN. Section 4 discusses the numerical illustrations and their results. Section 5 summarizes the main findings of the study and outlines possible avenues for future investigation.

Throughout this paper, the terms Maximum Absolute Error (MAE) and L_∞ error are used interchangeably. Both quantities represent the maximum absolute difference between the exact solution and the numerical approximation evaluated over all discrete spatial–temporal grid points. For clarity and consistency, the L_∞ error reported throughout the paper corresponds exactly to the MAE.

2. Heuristic Justification and Motivating Analysis

In this section, we present a heuristic and structural justification of the proposed WENN framework based on classical singular perturbation theory and approximation arguments. The purpose of this section is to provide qualitative insight into the design of the method and to motivate why the proposed architecture is effective for singularly perturbed problems.

We emphasize that no rigorous convergence proof or parameter-uniform error estimate is claimed. All conclusions drawn in this section should be interpreted as formal or heuristic observations, whose validity is subsequently supported by numerical evidence in Section 4.

2.1 Asymptotic Structure of the Solution

For sufficiently small perturbation parameter ε, solutions of SPPCD problems are known to admit a decomposition of the form

$$u(x,t) = u_0(x,t) + A(x,t)\, e^{-a(1-x)/\varepsilon}.$$

where, $u_0(x,t)$ denotes a smooth outer solution and the second term represents a boundary-layer correction localized near the inflow boundary $x = 1$.

Such decompositions are well established in classical singular perturbation theory (see, e.g., Kadalbajoo and Patidar, 2006; Kadalbajoo and Gupta, 2010). The exponential term captures the steep spatial gradients induced by small ε, while both u_0 and A remain smooth functions with respect to x and t.

Motivated by this structure, the WENN framework embeds the boundary-layer behavior explicitly into the solution representation:

$$u_{\text{WENN}}(x,t) = u_{\theta_0}(x,t) + e^{-a(1-x)/\varepsilon}\, u_{\phi_1}(x,t) \tag{2}$$

where, u_{θ_0} and u_{ϕ_1} are neural network approximations of smooth components. This design ensures that the networks are not required to learn steep gradients directly, thereby reducing stiffness in the learning process.

2.2 Approximation Capability

Let N_{θ_0} and N_{ϕ_1} denote the sets of functions representable by the subnetworks parameterized by θ_0 and ϕ_1, respectively. According to the universal approximation theorem (Hornik, 1991), for any continuous target functions $u_0(x,t)$ and $A(x,t)$ defined on compact domains, there exist neural networks such that

$$||u_0 - u_{\theta_0}||_\infty \le \delta \quad , ||A - u_{\phi_1}||_\infty \le \delta, \ \forall\, \delta > 0.$$

Hence, the subnetworks u_{θ_0} and u_{ϕ_1} can approximate these functions arbitrarily well in the uniform norm. We use this result here as a motivating argument to justify the expressive capacity of the proposed subnetworks. Substituting these approximations into the representation (2), we formally obtain

$$||u - u_\theta||_\infty \le ||u_0 - u_{\theta_0}||_\infty + e^{-a(1-x)/\varepsilon}\, ||A - u_{\phi_1}||_\infty.$$

This estimate should be interpreted as a heuristic indication of the approximation behavior, rather than a rigorous parameter-uniform error bound.

This argument suggests that the proposed representation is well-suited to approximate solutions of SPPs, provided the boundary-layer structure is appropriately captured. While this does not constitute a rigorous parameter-uniform error bound, it motivates the observed robustness with respect to small perturbation parameters.

2.3 Weighted Ensemble Stability

The WENN model combines multiple subnetworks $\{u_i(x,t)\}_{i=1}^m$ using normalized Gaussian weights

$$w_i(x) = \frac{\exp\left[-(x-\mu_i)^2/(2\sigma^2)\right]}{\sum_{j=1}^m \exp\left[-(x-\mu_j)^2/(2\sigma^2)\right]}, \quad \sum_{i=1}^m w_i(x) = 1.$$

The final ensemble approximation is given by

$$u_{\text{WENN}}(x,t) = \sum_{i=1}^{m} w_i(x)\, u_i(x,t).$$

Since the weights form a smooth partition of unity, the ensemble output is a convex combination of subnetwork predictions. Consequently, the ensemble inherits the qualitative stability of the individual subnetworks and avoids artificial discontinuities at subdomain interfaces. This argument is qualitative in nature and serves to motivate the use of Gaussian weighting rather than hard domain decomposition.

2.4 Remarks on Parameter Dependence

Due to the explicit inclusion of the exponential boundary-layer term in the network representation, the dominant ε-dependent behavior is analytically prescribed rather than learned. This design reduces the burden on the neural networks and mitigates numerical stiffness for small ε.

While no rigorous statement on parameter-uniform convergence is established, the numerical experiments in Section 4 demonstrate that the approximation error remains stable and bounded across several orders of magnitude of ε. These observations support the practical robustness of the proposed method, though they are reported strictly as empirical findings rather than theoretical guarantees.

3. WENN Framework

This section describes the proposed WENN method for approximating solutions of singularly perturbed parabolic convection--diffusion equations with Robin boundary conditions. The emphasis here is on the numerical construction and training procedure; theoretical motivation has already been discussed in Section 2.

3.1 Training Data and Input Representation

The neural networks take the spatial - temporal coordinates $(x,t) \in \Omega \times [0,\mathrm{T}]$ as input variables. Let $M = N_x \times N_t$ denote the total number of training points, where N_x and N_t represent the number of spatial and temporal sampling points, respectively. Training data are generated from known exact solutions and corresponding initial and boundary data, resulting in a purely supervised learning framework. It should be noted that the training data employed in the present study are generated from known analytical solutions. This choice was made deliberately to isolate and evaluate the approximation capability of the proposed WENN architecture under controlled benchmark conditions. Consequently, the current framework should be viewed primarily as a supervised learning approach for benchmark validation rather than a fully general-purpose PDE solver. While this setting enables a clear assessment of the network's ability to capture sharp boundary-layer structures, it also limits direct applicability to problems for which exact solutions are unavailable. Future work will focus on extending the framework toward physics-informed and hybrid formulations that incorporate governing equation residuals, sparse observational data, or both.

3.2 Subnetwork Architecture

The WENN architecture consists of two independent feedforward neural subnetworks, denoted by Net_1 and Net_2. Each subnetwork has the following structure:

1) an input layer with two neurons corresponding to (x,t),
2) two hidden layers with 32 neurons each,
3) an output layer with one neuron representing the solution value.

The hidden layers employ the hyperbolic tangent sigmoid activation function (tansig), while the output layer uses a linear activation function (purelin).

Each subnetwork is trained independently using the Levenberg-Marquardt optimization algorithm (trainlm) with a mean squared error (MSE) loss function. Training is performed for up to 2000 epochs with an initial learning rate of 0.003. The outputs of the subnetworks are denoted by

$$u_1(x,t) = \mathrm{Net}_1(x,t), \quad u_2(x,t) = \mathrm{Net}_2(x,t).$$

3.3 Gaussian Weighting and Ensemble Construction

To combine the subnetwork predictions, spatially varying Gaussian weight functions are introduced:

$$w_i(x) = \frac{e^{\left(-\frac{(x-\mu_i)^2}{2\sigma^2}\right)}}{e^{\left(-\frac{(x-\mu_1)^2}{2\sigma^2}\right)} + e^{\left(-\frac{(x-\mu_2)^2}{2\sigma^2}\right)}}, \; i = 1, 2.$$

Here, $\mu_1 = 0.25$, $\mu_2 = 0.75$ are fixed centers, and $\sigma = 0.15$ controls the spread of the Gaussian functions. The normalization ensures that

$$w_1(x) + w_2(x) = 1 \quad \text{for all } x \in [0,1].$$

The WENN approximation is defined as the convex combination

$$u_{\mathrm{WENN}}(x,t) = w_1(x) \cdot u_1(x,t) + w_2(x) \cdot u_2(x,t).$$

These smooth spatial weights allow each subnetwork to contribute more strongly in different regions of the domain while avoiding sharp interfaces or explicit domain decomposition.

3.4 Boundary-Layer Representation

In order to incorporate known structural features of singularly perturbed problems, the WENN approximation is expressed in the form

$$u^{\Theta}(x,t) = u_0^{\theta}(x,t) + u_1^{\phi}(x,t) \cdot e^{-a(1-x)/\varepsilon},$$

where, u_0^{θ} and u_1^{ϕ} are neural network outputs representing the smooth component and the boundary-layer amplitude, respectively, and $\Theta = \{\theta, \phi\}$ denotes the collection of all trainable parameters.

The exponential term encodes the dominant boundary-layer behavior analytically, reducing the need for the neural networks to learn steep gradients directly. This representation is motivated by classical singular perturbation theory and is used here as a modeling assumption rather than a source of rigorous error estimates.

3.5 Loss Function and Enforcement of Constraints

The proposed method does not include the governing PDE residual in the loss function. Instead, only data-based constraints are imposed. The total loss is defined as

$$\mathcal{L}(\Theta) = \mathcal{L}_{\mathrm{ic}}(\Theta) + \mathcal{L}_{\mathrm{bc}}(\Theta).$$

The initial condition loss is given by

$$\mathcal{L}_{\mathrm{ic}}(\Theta) = \frac{1}{N_{ic}} \sum_{i=1}^{N_{ic}} \left(u^{\Theta}(x_i, 0) - s(x_i)\right)^2,$$

where, $s(x)$ denotes the prescribed initial data. The Robin boundary condition loss is defined as

$$\mathcal{L}_{\mathrm{bc}}(\Theta) = \frac{1}{N_{bc}}\sum_{i=1}^{N_{bc}}\left(u^{\Theta}(0,t_i) - \varepsilon\frac{\partial u^{\Theta}}{\partial x}(0,t_i) - q_0(t_i)\right)^2 + \left(u^{\Theta}(1,t_i) + \varepsilon\frac{\partial u^{\Theta}}{\partial x}(1,t_i) - q_1(t_i)\right)^2.$$

Spatial derivatives appearing in the boundary conditions are computed using automatic differentiation. These derivatives are used solely to evaluate boundary mismatches and do not correspond to enforcement of the governing PDE.

3.6 Error Measures

After training, the predicted solution is evaluated on a test grid and reshaped into an array of size $N_t \times N_x$. The MAE is computed as

$$\mathrm{MAE} = \max_{(x,t)} |u_{WENN}(x,t) - u_{exact}(x,t)|.$$

We have also computed L_2 error that measures the average error over the entire domain. The L_2 error is computed as

$$||u_{exact} - u_{WENN}||_{L_2} = \left(\frac{1}{N}\sum_{i=1}^{N}|u_{exact}(x_i,t_i) - u_{WENN}(x_i,t_i)|^2\right)^{1/2}.$$

Errors are reported for multiple values of the perturbation parameter ε in order to assess the empirical robustness of the method.

4. Numerical Results

Example 1:

Let us examine the following SPPCD as presented in Gelu and Duressa (2022):

$$\left(\varepsilon\frac{\partial^2 u}{\partial x^2} + 2\frac{\partial u}{\partial x} - \frac{\partial u}{\partial t}\right)(x,t) = f(x,t), \quad (x,t) \in (0,1)\times(0,1],$$

$$u(x,0) = \theta(x), \quad x \in [0,1],$$

$$\left(u - \varepsilon\frac{\partial u}{\partial x}\right)(0,t) = \eta(t), \quad t \in [0,1],$$

$$\left(u + \varepsilon\frac{\partial u}{\partial x}\right)(1,t) = \varphi(t), \quad t \in [0,1].$$

In this case, function $f(x,t)$ and the prescribed initial and boundary conditions are carefully chosen such that the analytical solution is explicitly known:

$$u(x,t) = \frac{e^{-2x/\varepsilon} - e^{-2/\varepsilon}}{1 - e^{-2/\varepsilon}} + e^{-t}.$$

Two separate feedforward neural networks are constructed and trained independently using the same dataset generated from the PDE problem. Each neural network has the following architecture:

- **Input layer:** 2 neurons corresponding to spatial and temporal variables (x,t).
- **Hidden layers:** Two layers with 32 neurons each.
- **Activation functions:**

  - Hidden layers: tansig (hyperbolic tangent sigmoid)
  - Output layer: purelin (linear function)
- **Output layer:** 1 neuron representing the approximated solution $u(x,t)$.

The networks are trained using the Levenberg–Marquardt backpropagation algorithm (trainlm) to minimize the MSE. The training is terminated upon reaching either the maximum number of iterations (1000 for each network) or the minimum gradient threshold.

For $\boldsymbol{\varepsilon = 0.1}$ and $\boldsymbol{\varepsilon = 0.0001}$, the training was executed in two stages for the ensemble networks. We have:

- Maximum epochs (for each network): 1000
- Performance goal: 1×10^{-7}
- Initial learning rate: 0.003
- Gradient threshold for stopping: 1×10^{-7}
- Training function: trainlm
- Performance function: MSE
- Computational engine: MEX

For $\boldsymbol{\varepsilon = 0.1}$, we have:

First Network Training:

- Number of epochs: 723
- Training time: 4 minutes and 41 seconds
- Final performance (MSE): 1.12×10^{-10}
- Final gradient: 1.00×10^{-7}
- Training termination reason: Minimum gradient reached

Second Network Training:

- Number of epochs: 402
- Training time: 3 minutes and 12 seconds
- Final performance (MSE): 2.63×10^{-10}
- Final gradient: 1.00×10^{-7}
- Training termination reason: Minimum gradient reached

For $\boldsymbol{\varepsilon = 0.0001}$, we have:

First Network Training:

- Number of epochs: 851
- Training time: 5 minutes and 21 seconds
- Final performance (MSE): 1.12×10^{-10}
- Final gradient: 1.00×10^{-7}
- Training termination reason: Minimum gradient reached

Second Network Training:

- Number of epochs: 440
- Training time: 3 minutes and 44 seconds
- Final performance (MSE): 2.63×10^{-10}
- Final gradient: 1.00×10^{-7}
- Training termination reason: Minimum gradient reached

These results indicate that both networks achieved convergence well before the maximum epoch limit, with the gradients reaching their defined thresholds. The performance values demonstrate high accuracy in the network approximations.

Figures 1 and **2** shows the WENN predicted solution of example 1 for various values of ε with a spatial grid of $N = 64$ points and a time step size of $\Delta t = 1/64$ clearly demonstrating the method's ability to capture sharp boundary-layer behavior without spurious oscillations, while the corresponding error distributions are presented in **Figures 3** and **4** respectively, where the pointwise errors remain uniformly small even as ε decreases, confirming the stability of the approach.

Quantitative error measures are summarized in **Table 2**, which reports the MAEs and L_2 error for multiple ε values, along with their logarithmic measures, highlighting the weak dependence of the error on ε. **Table 3** presents the observed empirical slopes of the MAE with respect to ε, serving as trend indicators rather than strict convergence rates.

Figures 5 and **6** show log–log plots of the L_∞ and L_2 errors versus the perturbation parameter ε. The error remains bounded and of comparable magnitude across several orders of ε, indicating ε - robust performance of the proposed WENN method. The results in **Table 4** indicate a consistent reduction in both L_∞ and L_2 errors as the mesh are refined, confirming the convergence behavior of the WENN method for fixed ε.

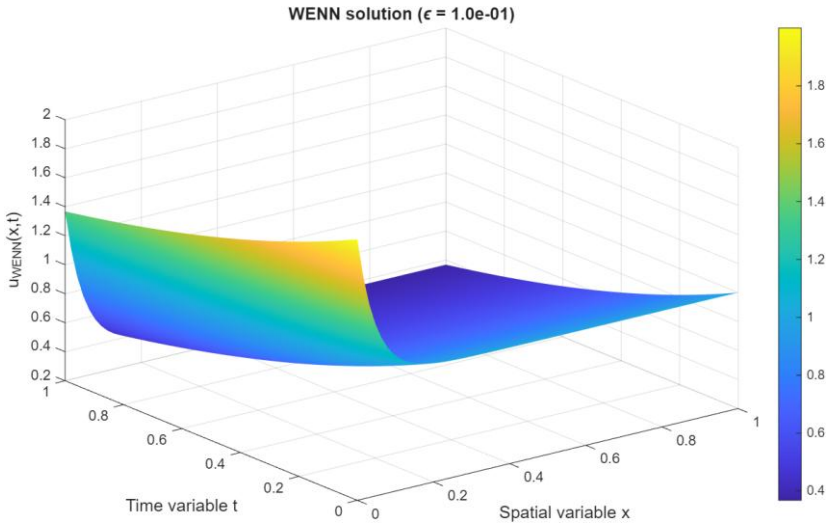

Figure 1. The WENN predicted solution for example 1 is computed using $\varepsilon = 0.1$.

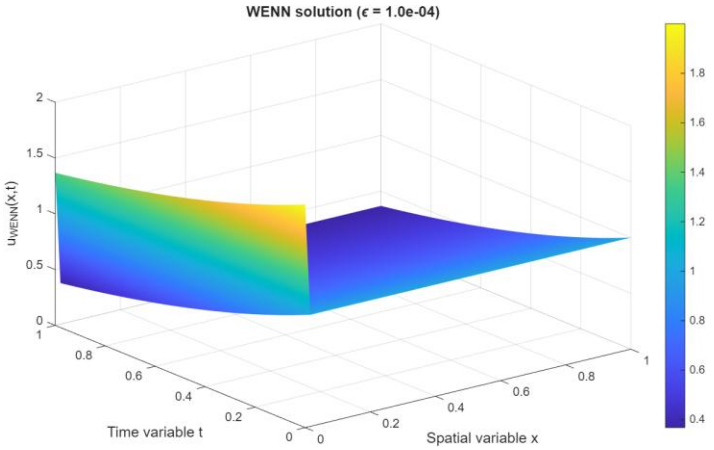

Figure 2. The WENN predicted solution for example 1 is computed using $\varepsilon = 0.0001$.

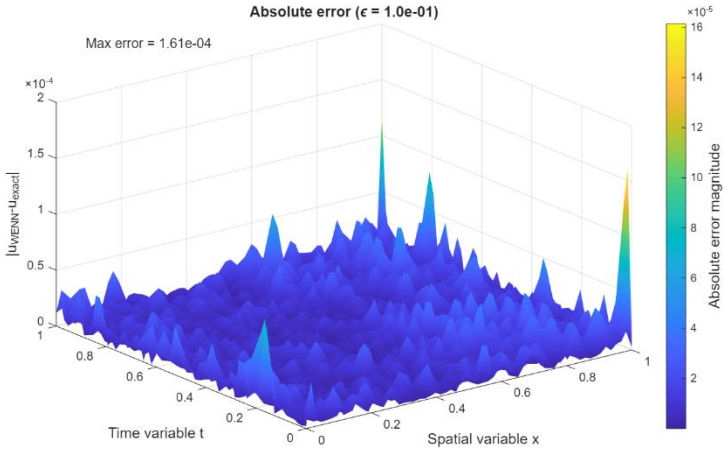

Figure 3. The error plot for example 1 is computed using $\varepsilon = 0.1$.

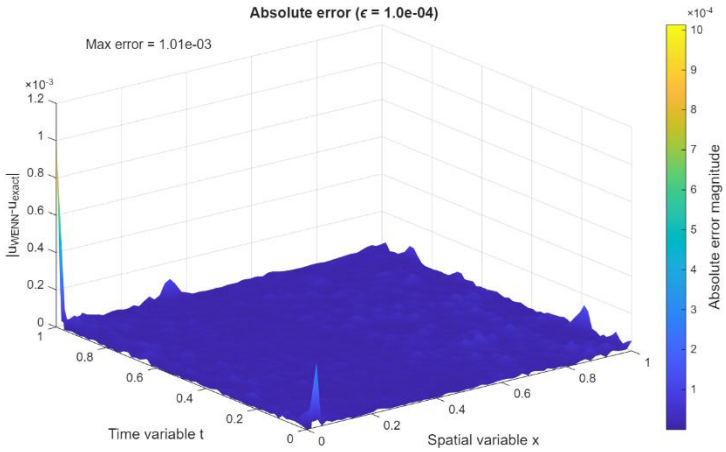

Figure 4. The error plot for example 1 is computed using $\varepsilon = 0.0001$.

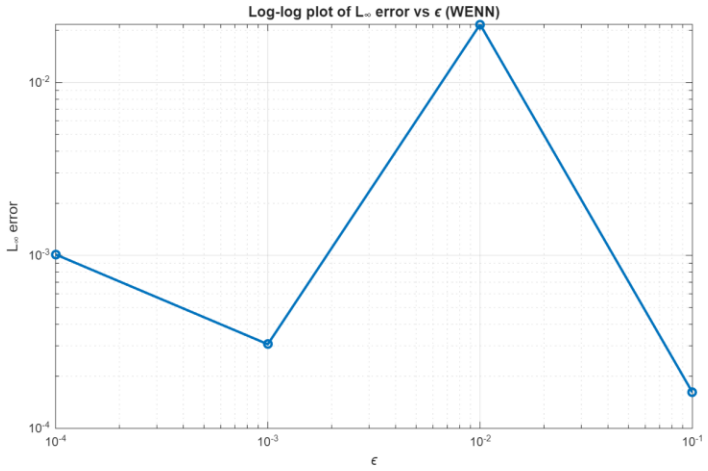

Figure 5. Log–log plot of L_∞- error versus ε for example 1 using the WENN method.

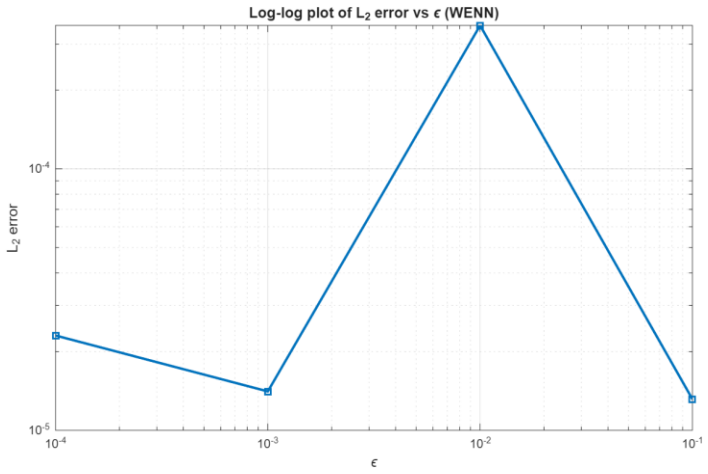

Figure 6. Log–log plot of L_2 - error versus ε for example 1 using the WENN method.

Table 2. Maximum pointwise and L_2 errors of the WENN, including logarithmic error measures for example 1.

Various values of perturbation parameter	**WENN**		
ε	**MAE**	L_2 **Error**	$\log_{10}(MAE)$
0.1	1.6143×10^{-4}	1.3148×10^{-5}	-3.792
0.01	2.1655×10^{-2}	3.5350×10^{-4}	-1.664
0.001	3.0647×10^{-4}	1.4083×10^{-5}	-3.514
0.0001	1.0131×10^{-3}	2.3042×10^{-5}	-2.994

Table 3. Observed empirical slopes of the MAE error with respect to ε for example 1.

ε_k	ε_{k+1}	**Observed slope**
10^{-1}	10^{-2}	-2.128
10^{-2}	10^{-3}	1.849
10^{-3}	10^{-4}	-0.519

Table 4. Convergence study for example 1 with $\varepsilon = 1.0 \times 10^{-3}$ using the WENN method.

$N_x = N_t$	L_∞ Error	L_2 Error	$\log_{10}(L_\infty)$
16	7.4025×10^{-1}	8.2116×10^{-2}	-0.1306
32	1.3566×10^{-2}	7.6811×10^{-4}	-1.8675
64	3.6812×10^{-3}	6.2359×10^{-5}	-2.4340
128	1.5493×10^{-4}	1.1667×10^{-5}	-3.8099

Example 2:
Consider SPPCD equation in Gelu and Duressa (2022):

$$\left(\varepsilon\frac{\partial^2 u}{\partial x^2} + \frac{\partial u}{\partial x} - \frac{\partial u}{\partial t}\right)(x,t) = f(x,t), \quad (x,t) \in (0,1) \times (0,1],$$
$$u(x,0) = \theta(x), \quad x \in [0,1],$$
$$\left(u - \varepsilon\frac{\partial u}{\partial x}\right)(0,t) = \eta(t), \quad t \in [0,1],$$
$$\left(u + \varepsilon\frac{\partial u}{\partial x}\right)(1,t) = \varphi(t), \quad t \in [0,1].$$

Here, the function $f(x,t)$, along with the initial condition $\theta(x)$ and boundary conditions $\varphi(t)$ and $\eta(t)$, are appropriately defined to ensure that the analytical solution is known and takes the form:

$$u(x,t) = \frac{e^{-x/\varepsilon} - e^{-1/\varepsilon}}{1 - e^{-1/\varepsilon}}\sin(2t) + 2x\cos\left(\frac{\pi x}{2}\right)\sin(t).$$

Two separate feedforward neural networks are constructed and trained independently using the same dataset generated from the PDE problem. Each neural network has the following architecture:

- **Input layer:** 2 neurons corresponding to spatial and temporal variables (x,t).
- **Hidden layers:** Two layers with 32 neurons each.
- **Activation functions:**
  - Hidden layers: tansig (hyperbolic tangent sigmoid)
  - Output layer: purelin (linear function)
- **Output layer:** 1 neuron representing the approximated solution $u(x,t)$.

The networks are trained using the Levenberg–Marquardt backpropagation algorithm (trainlm) to minimize the MSE. The training is terminated upon reaching either the maximum number of iterations (1000 for each network) or the minimum gradient threshold.

For $\varepsilon = 0.1$ and $\varepsilon = 0.0001$, the training was executed in two stages for the ensemble networks. We have:

- Maximum epochs (for each network): 1000
- Performance goal: 1×10^{-7}
- Initial learning rate: 0.003
- Gradient threshold for stopping: 1×10^{-7}
- Training function: trainlm
- Performance function: MSE
- Computational engine: MEX

For $\varepsilon = 0.1$, we have:

First Network Training:

- Number of epochs: 747
- Training time: 4 minutes and 14 seconds
- Final performance (MSE): 1.14×10^{-10}
- Final gradient: 1.01×10^{-7}
- Training termination reason: Minimum gradient reached

Second Network Training:

- Number of epochs: 629
- Training time: 3 minutes and 32 seconds
- Final performance (MSE): 2.07×10^{-10}
- Final gradient: 1.00×10^{-7}
- Training termination reason: Minimum gradient reached

For $\varepsilon = 0.0001$, we have:

First Network Training:

- Number of epochs: 851
- Training time: 4 minutes and 21 seconds
- Final performance (MSE): 1.12×10^{-10}
- Final gradient: 1.00×10^{-7}
- Training termination reason: Minimum gradient reached

Second Network Training:

- Number of epochs: 489
- Training time: 3 minutes and 44 seconds
- Final performance (MSE): 2.63×10^{-10}
- Final gradient: 1.00×10^{-7}
- Training termination reason: Minimum gradient reached

These results indicate that both networks achieved convergence well before the maximum epoch limit, with the gradients reaching their defined thresholds. The performance values demonstrate high accuracy in the network approximations.

Figures 7 and **8** illustrate the WENN-predicted solutions for example 2 corresponding to different values of ε with a spatial grid of $N = 64$ points and a time step size of $\Delta t = 1/64$. These figures clearly show that the proposed method accurately resolves sharp boundary-layer features without introducing nonphysical oscillations. The associated pointwise error distributions are displayed in **Figures 9** and **10**, where the errors remain uniformly small as ε decreases, indicating the numerical stability of the method.

A quantitative summary of the error behavior is provided in **Table 5**, which lists the MAE and L_2 error norms for various values of ε, together with their logarithmic values, demonstrating a weak dependence of the error on the perturbation parameter. Furthermore, **Table 6** reports the empirically observed slopes of the MAE with respect to ε, which are included as qualitative indicators of error trends rather than formal convergence rates.

Figures 11 and **12** show log-log plots of the L_∞ and L_2 errors versus the perturbation parameter ε. The error remains bounded and of comparable magnitude across several orders of ε, indicating ε - robust performance of the proposed WENN method. Finally, the results reported in **Table 7** demonstrate a systematic reduction in both L_∞ and L_2 errors with mesh refinement, confirming the convergence behavior of the WENN method.

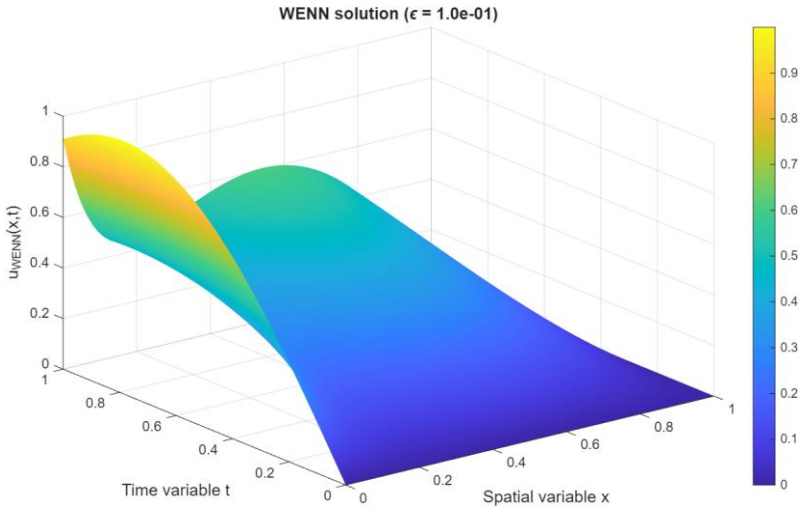

Figure 7. The WENN predicted solution for example 2 is computed using $\varepsilon = 0.1$.

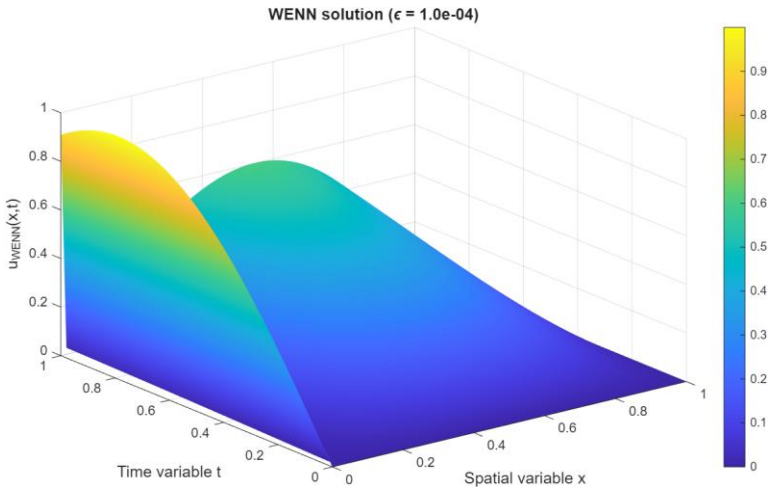

Figure 8. The WENN predicted solution for example 2 is computed using $\varepsilon = 0.0001$.

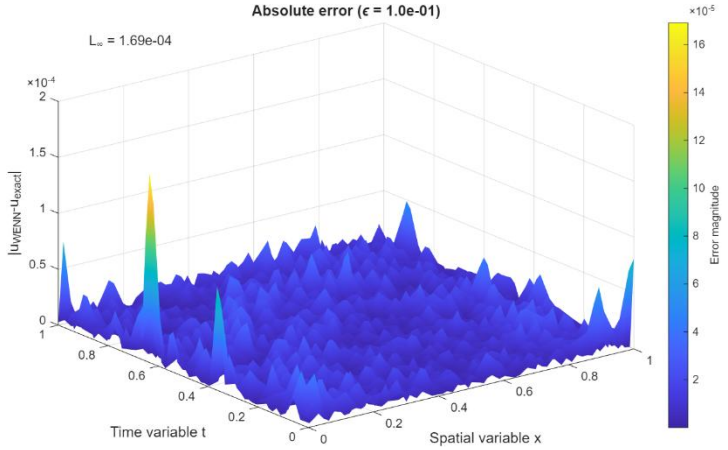

Figure 9. The error plot for example 2 is computed using $\varepsilon = 0.1$.

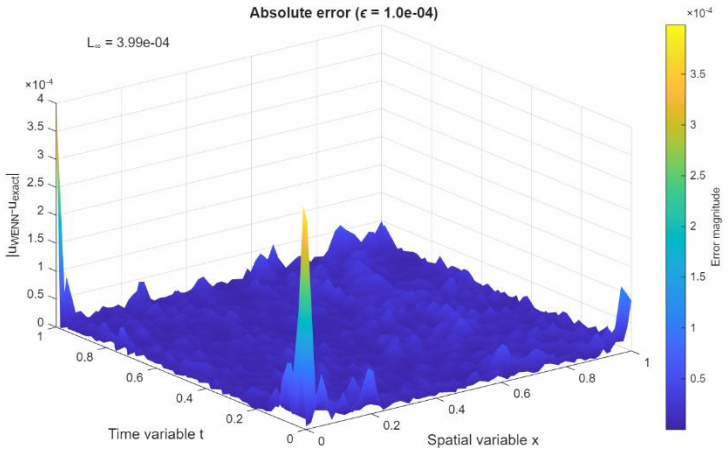

Figure 10. The error plot for example 2 is computed using $\varepsilon = 0.0001$.

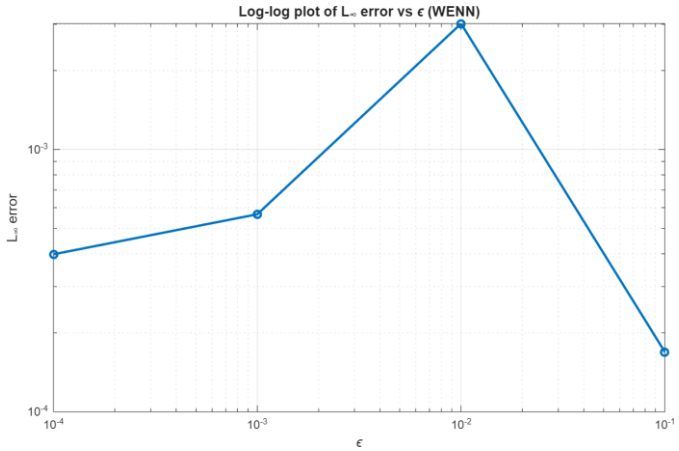

Figure 11. Log–log plot of L_∞ - error versus ε for example 2 using the WENN method.

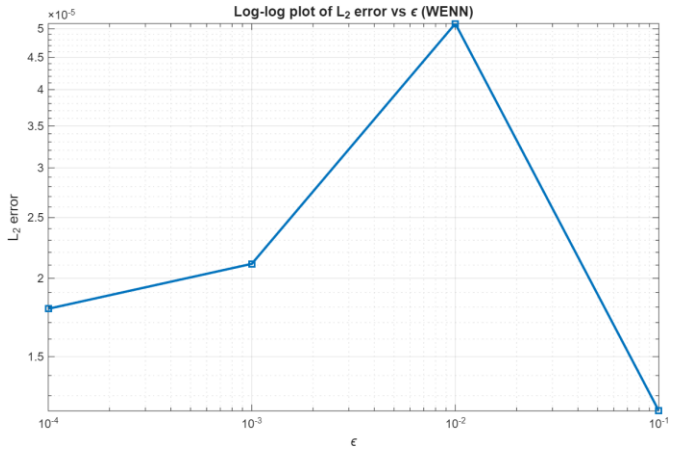

Figure 12. Log–log plot of L_2 - error versus ε for example 2 using the WENN method.

Table 5. Maximum pointwise and L_2 errors of the WENN, including logarithmic error measures for example 2.

Various values of perturbation parameter	WENN		
ε	MAE	L_2 Error	$\log_{10}(MAE)$
0.1	1.6914×10^{-4}	1.2290×10^{-5}	-3.7718
0.01	3.0172×10^{-3}	5.0969×10^{-5}	-2.5204
0.001	5.6628×10^{-4}	2.1093×10^{-5}	-3.2470
0.0001	3.9861×10^{-4}	1.7883×10^{-5}	-3.3995

Table 6. Observed empirical slopes of the MAE error with respect to ε for example 2.

ε_k	ε_{k+1}	Observed slope
10^{-1}	10^{-2}	-1.25140
10^{-2}	10^{-3}	0.72657
10^{-3}	10^{-4}	0.15248

Table 7. Convergence study for example 2 with $\varepsilon = 1.0 \times 10^{-3}$ using the WENN method.

$N_x = N_t$	L_∞ error	L_2 error	$\log_{10}(L_\infty)$
16	7.4025×10^{-1}	8.2116×10^{-2}	-0.1306
32	1.3566×10^{-2}	7.6811×10^{-4}	-1.8675
64	3.6812×10^{-3}	6.2359×10^{-5}	-2.4340
128	1.5493×10^{-4}	1.1667×10^{-5}	-3.8099

4.1 Comparison with Existing Methods

The results in **Tables 8** and **9** demonstrates that the proposed WENN method consistently yields lower MAE and L_2 errors than the Single Network for all tested values of ε. While the Single Network suffers from accuracy deterioration as ε decreases, WENN preserves stable and significantly reduced errors, confirming its effectiveness in resolving boundary layers. Also, we compared absolute error of Single Network and WENN technique, near boundary with $\varepsilon = 0.001$ in **Figure 13** for example 1 and in **Figure 14** for example 2.

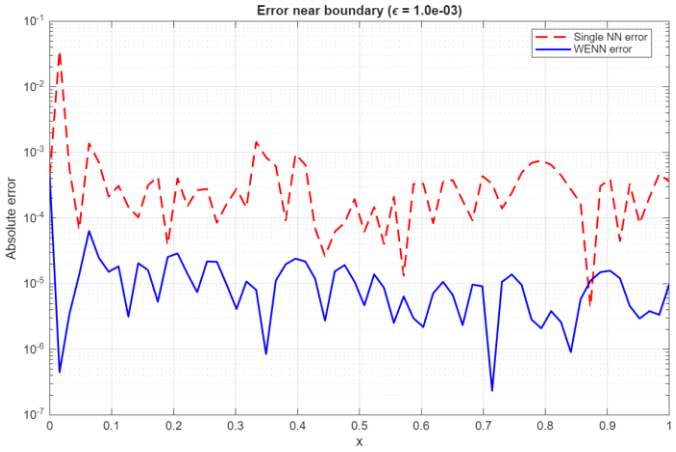

Figure 13. Comparison of absolute error for example 1 with $\varepsilon = 0.001$.

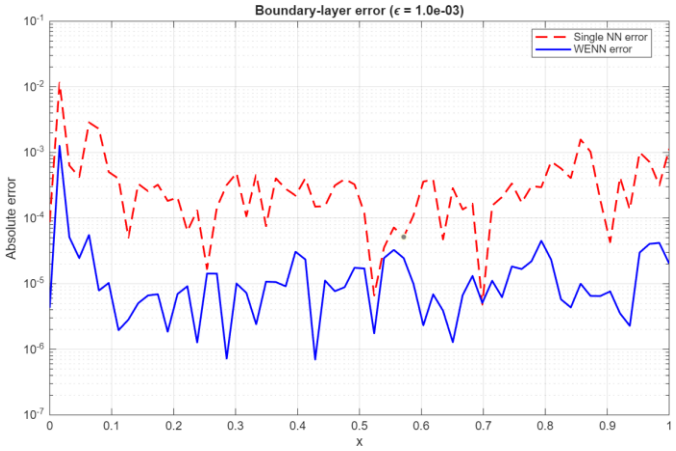

Figure 14. Comparison of absolute error for example 2 with $\varepsilon = 0.001$.

Table 8. Comparison of MAE and L_2 errors for single network and WENN methods for example 1.

ε	MAE		L_2 Error	
	Single Network	WENN	Single Network	WENN
10^{-1}	2.3137×10^{-3}	1.6143×10^{-4}	2.3248×10^{-4}	1.3148×10^{-5}
10^{-2}	2.4380×10^{-2}	2.1655×10^{-2}	5.4301×10^{-4}	3.5350×10^{-4}
10^{-3}	3.6117×10^{-2}	3.0647×10^{-4}	7.5288×10^{-4}	1.4083×10^{-5}
10^{-4}	8.7578×10^{-3}	1.0131×10^{-3}	4.7688×10^{-4}	2.3042×10^{-5}

Table 9. Comparison of MAE and L_2 errors for single network and WENN methods for example 2.

ε	MAE		L_2 Error	
	Single Network	WENN	Single Network	WENN
10^{-1}	2.2871×10^{-3}	1.6914×10^{-4}	3.3773×10^{-4}	1.2290×10^{-5}
10^{-2}	5.6914×10^{-3}	3.0172×10^{-3}	4.3860×10^{-4}	5.0969×10^{-5}
10^{-3}	1.1596×10^{-2}	5.6628×10^{-4}	4.9787×10^{-4}	2.1093×10^{-5}
10^{-4}	7.3627×10^{-3}	3.9861×10^{-4}	4.8289×10^{-4}	1.7883×10^{-5}

To further assess the effectiveness of the proposed WENN framework, we present a quantitative comparison with three widely used approaches for SPPs:

- PINNs,
- Shishkin mesh method,
- Bakhvalov-Shishkin mesh method.

***Differences in formulation*:**
The PINN approach enforces the governing PDE through residual minimization, which often leads to training stiffness for small perturbation parameters. In contrast, the proposed WENN method adopts a purely supervised learning framework and avoids PDE residual constraints. On the other hand, Shishkin and Bakhvalov-Shishkin methods are classical fitted mesh techniques that rely on layer-adapted grids. These methods require prior knowledge of boundary layer behavior and involve careful mesh construction to achieve parameter-uniform convergence. In contrast, the WENN framework is mesh-free and embeds the boundary layer structure directly into the neural network architecture.

***Computational aspects*:**
PINNs typically require repeated evaluation of PDE residuals and their derivatives, leading to increased computational cost and slower convergence. Fitted mesh methods require mesh generation and refinement strategies that depend on ε. The proposed WENN method avoids both residual computation and mesh construction, resulting in a simpler and computationally efficient framework.

***Quantitative performance*:**
Tables 10 and **11** present the comparison of MAE error for different values of ε with a spatial grid of $N = 64$ points and a time step size of $\Delta t = 1/64$. It is observed that the proposed WENN method consistently achieves lower error levels compared to fitted mesh methods (Gelu and Duressa, 2022).

Furthermore, while the accuracy of PINNs deteriorates due to training instability and fitted mesh methods depend strongly on mesh design, the WENN method maintains stable and accurate approximations across all tested parameter values.

Table 10. Comparison of MAE error for WENN and existing methods for example 1.

ε	WENN	Bakhvalov-Shishkin mesh	Shishkin mesh	PINN
10^{-2}	2.1655×10^{-2}	9.4854×10^{-2}	2.9693×10^{-1}	3.2050×10^{-1}
10^{-4}	1.0131×10^{-3}	9.4877×10^{-2}	2.9694×10^{-1}	9.0050×10^{-1}

Table 11. Comparison of MAE error for WENN and existing methods for example 2.

ε	WENN	Bakhvalov-Shishkin mesh	Shishkin mesh	PINN
10^{-2}	3.0172×10^{-3}	3.9327×10^{-2}	1.0809×10^{-1}	3.1150×10^{-1}
10^{-4}	3.9861×10^{-4}	4.1349×10^{-2}	1.0819×10^{-1}	8.0150×10^{-1}

These results clearly demonstrate that the proposed method provides an effective alternative for solving SPPs, combining the advantages of mesh-free approximation with improved accuracy and robustness.

It is observed that the reported errors do not always vary monotonically with respect to the perturbation parameter ε. This behavior is not unexpected for neural-network-based approximation methods. Unlike classical finite difference or finite element schemes, where the error is primarily controlled by discretization

parameters, the performance of neural networks depends on a combination of factors including weight initialization, optimization trajectory, network expressiveness, and the complexity of the target solution.

As ε decreases, the boundary layer becomes increasingly sharp, making the approximation problem more challenging. However, the final error is influenced not only by the perturbation parameter but also by the effectiveness of the training process in capturing localized solution features. Consequently, moderate fluctuations in error may occur across different ε values. The observed behavior should therefore be interpreted as a manifestation of optimization sensitivity rather than numerical instability. Despite these variations, the results demonstrate that the proposed WENN framework maintains bounded errors and stable approximation quality over a wide range of perturbation parameters.

Neural-network training inherently involves non-convex optimization, and therefore some sensitivity to initialization and hyperparameter selection is unavoidable. Although the numerical results presented in this study were found to be reproducible across repeated runs with similar settings, slight variations in the final approximation error may occur due to optimization-related factors.

The reported training times should be interpreted as offline computational costs associated with neural-network optimization rather than direct solution-evaluation costs. Although the benchmark problems considered in this work are one-dimensional, the training process involves repeated parameter updates over a large number of epochs to accurately capture sharp boundary-layer structures. Consequently, the training times may appear relatively high when compared with classical numerical solvers specifically designed for low-dimensional problems.

However, the proposed WENN framework avoids several procedures commonly required by traditional singular perturbation methods, including layer-adapted mesh generation, mesh refinement strategies, and problem-specific discretization design. In addition, unlike PINNs, the present approach does not require repeated evaluation of PDE residuals through automatic differentiation over interior collocation points. Once training has been completed, the resulting neural-network approximation can be evaluated rapidly at arbitrary spatial and temporal locations.

Nevertheless, for the relatively simple one-dimensional benchmark problems considered in this study, classical finite difference and fitted-mesh methods remain computationally more efficient than neural-network-based approaches. Therefore, the primary objective of the proposed framework is not to compete with established numerical solvers in terms of computational speed, but rather to investigate the effectiveness of ensemble neural architectures for representing singularly perturbed solution structures. Improving computational efficiency and extending the framework to higher-dimensional problems constitute important directions for future work.

5. Conclusions

In this research, we proposed a WENN framework incorporating Kindred Subnetworks for accurately solving SPPCDs with RBC. The method effectively addressed the numerical challenges posed by boundary layers and sharp gradients by decomposing the domain into regions, with each subnetwork trained independently to capture local solution features. These subnetworks were adaptively blended using spatially dependent Gaussian weights, allowing the ensemble model to represent the global solution with high fidelity.

Unlike conventional PINNs, our approach adopted a supervised learning strategy based on exact solution data, avoiding the complexities associated with physics-based loss formulations. The proposed method

demonstrated consistent accuracy across a wide range of perturbation parameters, without relying on mesh refinement or domain-specific decomposition techniques. Numerical results confirmed the capability of the WENN approach to resolve boundary layers effectively and achieve low approximation error. Our proposed method demonstrates relatively lower errors in comparison to the results reported in Gelu and Duressa (2022).

To assess the benefit of the ensemble strategy, a baseline comparison with a single neural network of comparable size is presented. The results indicate that the single-network model fails to accurately resolve boundary layers for small values of the perturbation parameter, leading to loss stagnation and significant error growth. In contrast, the proposed WENN approach maintains stable accuracy across all tested ε values.

Despite these promising outcomes, this work focused exclusively on one-dimensional problems with known exact solutions. A key limitation of the present work is the reliance on exact solution data for training, which restricts direct applicability to benchmark problems. However, the proposed framework can be extended to hybrid physics-informed formulations, as demonstrated in this study, enabling its application to problems where analytical solutions are unavailable. Additionally, experiments with sparse data indicate that the method retains reasonable accuracy even under limited data conditions.

The current formulation of WENN relies on supervised training using exact solution data. This restricts its direct applicability to benchmark problems where analytical solutions are available.

However, the framework can be naturally extended in the following ways:

(i) Hybrid WENN (Physics + Data):
A PDE residual term can be incorporated into the loss function to enable training without exact solutions, leading to a hybrid physics-informed WENN formulation.

(ii) Sparse data learning:
The model can be trained using limited observational data combined with boundary conditions, making it applicable to real-world scenarios.

(iii) Noisy data handling:
Regularization techniques can be incorporated to improve robustness under noisy measurements.

These extensions will be explored in future work.

Conflicts of Interest
The authors declare no competing interest.

Acknowledgments
This study was conducted without financial support from any public, commercial, or non-profit funding bodies. The authors express their sincere appreciation to the editor and anonymous reviewers for their valuable suggestions that enhanced the quality of this paper.

AI Disclosure
The author(s) declare that no assistance is taken from generative AI to write this article.

References

Ahmed, H.M. (2023). Highly accurate method for boundary value problems with Robin boundary conditions. *Journal of Nonlinear Mathematical Physics*, *30*(3), 1239-1263. https://doi.org/10.1007/s44198-023-00124-6

Ahmed, H.M. (2024). Highly accurate method for a singularly perturbed coupled system of convection–diffusion equations with Robin boundary conditions. *Journal of Nonlinear Mathematical Physics*, *31*(1), 17. https://doi.org/10.1007/s44198-024-00182-4

Ahmed, H.M. (2025). New Jacobi Galerkin operational matrices of derivatives: a highly accurate method for solving two-point fractional-order nonlinear boundary value problems with robin boundary conditions. *Fractal and Fractional*, *9*(11), 686. https://doi.org/10.3390/fractalfract9110686

Ahmed, H.M. (2026a). A new Laguerre-Galerkin operational matrix: highly accurate method for the fractional bagley–torvik equation of variable coefficients with robin boundary conditions. *Journal of Nonlinear Mathematical Physics*, *33*(1), 36. https://doi.org/10.1007/s44198-026-00382-0

Ahmed, H.M. (2026b). An advanced approach for numerical solution of a class of Fredholm–Volterra integro-differential equations with mixed boundary conditions. *International Journal of Modern Physics C*, *37*(09), 2550154. https://doi.org/10.1142/S0129183125501542

Ahmed, H.M. (2026c). Highly accurate method for Poisson's and the Helmholtz's equations subject to robin boundary conditions. *Journal of Nonlinear Mathematical Physics*, *33*(1), 18. https://doi.org/10.1007/s44198-025-00371-9

Basha, P.M., & Shanthi, V. (2015). A numerical method for singularly perturbed second order coupled system of convection–diffusion robin type boundary value problems with discontinuous source term. *International Journal of Applied and Computational Mathematics*, *1*(3), 381-397. https://doi.org/10.1007/s40819-014-0021-7

Beguinet, A., Ehrlacher, V., Flenghi, R., Fuente, M., Mula, O., & Somacal, A. (2023). Deep learning-based schemes for singularly perturbed convection-diffusion problems. *ESAIM*: *Proceedings and Surveys*, *73*, 48-67. https://doi.org/10.1051/proc/202373048

Berrone, S., Canuto, C., Pintore, M., & Sukumar, N. (2023). Enforcing dirichlet boundary conditions in physics-informed neural networks and variational physics-informed neural networks. *Heliyon*, *9*(8), e18820. https://doi.org/10.1016/j.heliyon.2023.e18820

Cao, F., Gao, F., Guo, X., & Yuan, D. (2023). Physics-informed neural networks with parameter asymptotic strategy for learning singularly perturbed convection-dominated problem. *Computers & Mathematics with Applications*, *150*, 229-242. https://doi.org/10.1016/j.camwa.2023.09.030

Cao, F., Gao, F., Yuan, D., & Liu, J. (2024). Multistep asymptotic pre-training strategy based on PINNs for solving steep boundary singular perturbation problems. *Computer Methods in Applied Mechanics and Engineering*, *431*, 117222. https://doi.org/10.1016/j.cma.2024.117222

Chawla, S., Urmil, & Singh, J. (2023). Fitted mesh numerical method for a coupled system of singularly perturbed reaction-diffusion robin boundary value problem having boundary and internal layers. *Indian Journal of Pure and Applied Mathematics*, *54*(3), 675-688. https://doi.org/10.1007/s13226-022-00285-y

Gelu, F.W., & Duressa, G.F. (2022). A novel numerical approach for singularly perturbed parabolic convection-diffusion problems on layer-adapted meshes. *Results in Mathematics*, *9*(1), 2020400. https://doi.org/10.1080/27658449.2021.2020400

Gie, G.-M., Hong, Y., & Jung, C.-Y. (2024). Semi-analytic PINN methods for singularly perturbed boundary value problems. *Applied Analysis*, *103*(14), 2554-2571. https://doi.org/10.1080/00036811.2024.2302405

Gupta, A., Kaushik, A., & Sharma, M. (2023). A higher-order hybrid spline difference method on adaptive mesh for solving singularly perturbed parabolic reaction–diffusion problems with robin boundary conditions. *Numerical Methods for Partial Differential Equations*, *39*(2), 1220-1250. https://doi.org/10.1002/num.22931

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, *4*(2), 251-257. https://doi.org/10.1016/0893-6080(91)90009-T

Kadalbajoo, M.K., & Gupta, V. (2010). A brief survey on numerical methods for solving singularly perturbed problems. *Applied Mathematics and Computation*, *217*(8), 3641-3716. https://doi.org/10.1016/j.amc.2010.09.059

Kadalbajoo, M.K., & Patidar, K.C. (2006). ε-Uniformly convergent fitted mesh finite difference methods for general singular perturbation problems. *Applied Mathematics and Computation*, *179*(1), 248-266. https://doi.org/10.1016/j.amc.2005.11.096

Kumar, D., & Deswal, K. (2022). Wavelet-based approximation for two-parameter singularly perturbed problems with Robin boundary conditions. *Journal of Applied Mathematics and Computing*, *68*, 125-149. https://doi.org/10.1007/s12190-021-01511-2

Kumar, S., Aakansha, Singh, J., & Ramos, H. (2023). Parameter-uniform convergence analysis of a domain decomposition method for singularly perturbed parabolic problems with robin boundary conditions. *Journal of Applied Mathematics and Computing*, *69*, 2239-2261. https://doi.org/10.1007/s12190-022-01832-w

Linß, T. (2007). Analysis of an upwind finite-difference scheme for a system of coupled singularly perturbed convection-diffusion equations. *Computing*, *79*, 23-32. https://doi.org/10.1007/s00607-006-0215-x

Liu, L.-B., & Chen, Y. (2015). An adaptive moving grid method for a system of singularly perturbed initial value problems. *Journal of Computational and Applied Mathematics*, *274*, 11-22. https://doi.org/10.1016/j.cam.2014.06.022

Liu, Y., Chen, L., Ding, J., & Chen, Y. (2024). An adaptive sampling method based on expected improvement function and residual gradient in PINNs. *IEEE Access*, *12*, 92130-92141. https://doi.org/10.1109/ACCESS.2024.3422224

Nguyen, H.D., & Chamroukhi, F. (2018). Practical and theoretical aspects of mixture-of-experts modeling: an overview. *WIREs Data Mining and Knowledge Discovery*, *8*(4), e1246. https://doi.org/10.1002/widm.1246

O'Riordan, E., & Stynes, M. (2009). Numerical analysis of a strongly coupled system of two singularly perturbed convection–diffusion problems. *Advances in Computational Mathematics*, *30*(2), 101-121. https://doi.org/10.1007/s10444-007-9058-z

Omkar, R., & Phaneendra, K. (2022). Numerical simulation of singularly perturbed delay differential equations with large delay using an exponential spline. *International Journal of Analysis and Applications*, *20*, 63. https://doi.org/10.28924/2291-8639-20-2022-63

Prince, P.A., Govindarao, L., & Elango, S. (2025). Richardson extrapolation for singularly perturbed Fredholm integro-differential equations. *International Journal of Mathematical, Engineering and Management Sciences, 10*(6), 2023-2039. 0.33889/IJMEMS.2025.10.6.094

Raissi, M., Perdikaris, P., & Karniadakis, G.E. (2019). 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. https://doi.org/10.1016/j.jcp.2018.10.045

Rao, S.C.S., & Chawla, S. (2020). Robin boundary value problems for a singularly perturbed weakly coupled system of convection–diffusion equations having discontinuous source term. *The Journal of Analysis*, *28*(1), 305-321. https://doi.org/10.1007/s41478-019-00172-6

Shan, B., & Li, Y. (2024). Asymptotic physics-informed neural networks for solving singularly perturbed problems. In *International Conference on Big Data and Security* (pp. 15-26). Springer Nature, Singapore. https://doi.org/10.1007/978-981-97-4390-2_2

Sharma, P., Evans, L., Tindall, M., & Nithiarasu, P. (2023). Stiff-PDEs and physics-informed neural networks. *Archives of Computational Methods in Engineering*, *30*(5), 2929-2958. https://doi.org/10.1007/s11831-023-09890-4

Temel, Z., & Çakır, M. (2024). A robust numerical method for a singularly perturbed semilinear problem with integral boundary conditions. *Contemporary Mathematics* (*Singapore*), *5*(1), 446-464. http://doi.org/10.37256/cm.5120243020

Wang, L., Zhang, L., & He, G. (2024). Chien-physics-informed neural networks for solving singularly perturbed boundary-layer problems. *Applied Mathematics and Mechanics*, *45*(9), 1467-1480. https://doi.org/10.1007/s10483-024-3149-8

Wei, P., Liu, M., Cen, J., Zhou, Z., Chen, L., & Zou, Q. (2024). PDENNEval: a comprehensive evaluation of neural network methods for solving PDEs. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence* (pp. 5181-5189). IJCAI. Jeju, Korea. https://doi.org/10.24963/ijcai.2024/573

Wu, C., Zhu, M., Tan, Q., Kartha, Y., & Lu, L. (2023). A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, *403*(Part A), 115671. https://doi.org/10.1016/j.cma.2022.115671

Yadav, N.S., & Mukherjee, K. (2024). Efficient parameter-robust numerical methods for singularly perturbed semilinear parabolic PDEs of convection-diffusion type. *Numerical Algorithms*, *96*(2), 925-973. https://doi.org/10.1007/s11075-023-01670-2

Yadav, S., & Rai, P. (2020). A higher order numerical scheme for singularly perturbed parabolic turning point problems exhibiting twin boundary layers. *Applied Mathematics and Computation*, *376*, 125095. https://doi.org/10.1016/j.amc.2020.125095