Matched Asymptotic Expansions-Based Transferable Neural Networks for Singular Perturbation Problems

Zhequan Shen[a], Lili Ju[b], Liyong Zhu[a,*]

[a]*School of Mathematical Sciences, Beihang University, Beijing 100191, China*
[b]*Department of Mathematics, University of South Carolina, Columbia, SC 29208, USA*

Abstract

In this paper, by utilizing the theory of matched asymptotic expansions, an efficient and accurate neural network method, named as "MAE-TransNet", is developed for solving singular perturbation problems in general dimensions, whose solutions usually change drastically in some narrow boundary layers. The TransNet is a two-layer neural networks with specially pre-trained hidden-layer neurons. In the proposed MAE-TransNet, the inner and outer solutions produced from the matched asymptotic expansions are first approximated by a TransNet with nonuniform hidden-layer neurons and a TransNet with uniform hidden-layer neurons, respectively. Then, these two solutions are combined with a matching term to obtain the composite solution, which approximates the asymptotic expansion solution of the singular perturbation problem. This process enables the MAE-TransNet method to retain the precision of the matched asymptotic expansions while maintaining the efficiency and accuracy of TransNet. Meanwhile, the rescaling of the sharp region allows the same pre-trained network parameters to be applied to boundary layers with various thicknesses, thereby improving the transferability of the method. Notably, for coupled boundary layer problems, a computational framework based on MAE-TransNet is also constructed to effectively addresses issues resulting from the lacking of relevant matched asymptotic expansion theory in such problems. Our MAE-TransNet is thoroughly compared with TransNet, PINN, and Boundary-Layer PINN (BL-PINN) on various benchmark problems including 1D linear and nonlinear problems with boundary layers, the 2D Couette flow problem, a 2D coupled boundary layer problem, and the 3D Burgers vortex problem. Numerical results demonstrate that MAE-TransNet significantly outperforms other neural network methods in capturing the characteristics of boundary layers, improving the accuracy, and reducing the computational cost.

Keywords: Singular perturbation problem, boundary layer, two-layer neural network, matched asymptotic expansion, transferability

1. Introduction

A singular perturbation generally occurs when a small parameter multiplies the highest derivative, leading to the formation of thin boundary layers in the solution. These boundary layers reveal

[*]Corresponding author.

Email addresses: zhequanshen@buaa.edu.cn (Zhequan Shen), ju@math.sc.edu (Lili Ju), liyongzhu@buaa.edu.cn (Liyong Zhu)

a wide range of natural phenomena such as high Reynolds number flows [1] and high Peclet number transfer [2] in fluid dynamics, oxygen concentration transport [3] in mathematical biology, and heat transfer simulation [4] in manufacturing technologies. In this paper, we consider the following singular perturbation problems with boundary layers:

$$\begin{cases} \mathcal{L}(\boldsymbol{u};\varepsilon) = f, & \boldsymbol{x} \in \Omega, \\ \mathcal{B}(\boldsymbol{u}) = g, & \boldsymbol{x} \in \partial\Omega, \end{cases} \tag{1}$$

where $\mathcal{L}$ denotes a differential operator defined on the domain (an open bounded set) $\Omega \in \mathbb{R}^d$, $\mathcal{B}$ denotes a boundary operator defined on the domain boundary $\partial\Omega$, and $0 < \varepsilon \ll 1$ is a small thickness parameter which is usually multiplied to the highest order derivative term (e.g., a small diffusion coefficient).

Accurately solving the singular perturbation problem (1) is quite challenging due to the large gradients within the boundary layers. Traditional numerical methods require extremely dense uniform meshes, leading to significant wastage of computational resources. Therefore, nonuniform mesh methods are developed to reduce computational costs by refining the meshes within the boundary layer [5, 6, 7, 8, 9]. Despite this, most classic numerical methods, such as finite difference methods (FDM) [10, 11, 12], finite element methods (FEM) [13, 14, 15, 16] and continuous internal penalty (CIP) method [17], still could be impractical when dealing with the case of very small singular perturbation parameters [18], especially for multi-dimensional multi-layer problems.

In recent years, numerous neural network methods have been developed to incorporate PDE information into the loss function and solve the corresponding problem through optimization techniques, such as Deep Ritz method (DRM) [19], Deep Galerkin method (DGM) [20] and Physics-informed neural networks (PINN) [21, 22]. Although these methods have presented good performance on many types of PDEs, it is challenging for them to capture the sharp changes within the boundary layers. Several PINN-based methods have been proposed to solve singular perturbation problems in [18, 23, 24, 25]. The method proposed in [26] includes computing an initial approximation to the problem using a simple neural network and estimating a finer correction in an iterative manner by solving the problem for the residual with a new network of increasing complexity. In the work of Boundary-Layer PINN (BL-PINN) [27], the solution to boundary layer problems in the singular perturbation theory is divided into solutions in inner and outer regions, and multiple asymptotic expansion basis functions in both regions are approximated by PINN. BL-PINN integrates perturbation methods with PINN, successfully approximate solutions for singular perturbation problems with boundary layers. However, the above-mentioned neural network methods usually have deep network architectures, which require a significant amount of computational cost due to the multitude of parameters in nonlinear and nonconvex optimization. Moreover, in addressing singular perturbation problems with boundary layers, these approaches often employ hyperparameters that are dependent on ε. Consequently, as ε changes, these hyperparameters must be readjusted, leading to additional computational costs for traning.

To overcome the challenge of high computational costs associated with deep neural networks, some researchers introduce shallow neural network (i.e, very few number of network layers) methods along with pre-trained parameters for the hidden layers, which only require solving a simple least squares problem concerning the parameters of the output layer, achieving greater accuracy for various PDEs when compared with the conventional neural network methods [28, 29, 30, 31]. Deep least-squares method proposed in [32] makes use of the neural network to approximate the solution of singularly perturbed reaction-diffusion equations through the compositional construction and employ first-order system least-squares functional as a loss function. For boundary layer problems in the singular perturbation theory, a numerical scheme based on the concept of Extreme Learning Machines (ELMs) has been proposed in [33]. Although these methods have demonstrated advantages in handling certain specific singular perturbation problems with boundary layers, they do not guarantee effectiveness for sufficiently small values of ε. Moreover, in practice, the hyperparameters in these methods usually lack interpretability, which increases the cost of selecting these parameters.

Matched asymptotic expansions (MAE) [34, 35, 36, 37] are used when the problem solution experiences a rapid change over a short interval (e.g., the solution to singular perturbation problems with boundary layers). MAE usually involve the construction of an expansion valid in the region of rapid change, called the inner solution, and an expansion valid outside this region, called the outer solution. The composite solution is consequently expressed as a sum of the inner solution, the outer solution, and a matching term which ensures the validity across the entire domain. Taking the case of Prandtl's matched asymptotic expansions, the error between the composite solution and the analytical solution achieves $O(\varepsilon)$ over the entire computational domain [34], which implies that the error decreases as ε diminishes. However, it is quite challenging to analytically solve the outer and inner solutions for the majority of problems [38].

Transferable neural network (TransNet) [39] is a two-layer neural network (i.e., one hidden layer and one output layer) with specially pre-trained hidden-layer neurons, and its key point is to construct a transferable neural feature space based on re-parameterization of the hidden layer neurons and approximation properties without using information from PDEs. It is noteworthy that the parameters in the neural feature space have explicit geometric meanings, which can guide parameter selection and thereby significantly reduce the cost of parameter tuning. The uniform neuron distribution theorem ensures the transferability of neurons in the hidden layer. The experimental results in [39] verify the excellent performance of TransNet, specifically, various problems with different types of PDEs, domains, or boundary conditions are solved with much greater accuracy than the state-of-the-art methods. Moreover, due to the utilization of pre-trained hidden layers, TransNet significantly outperforms PINN in terms of computational efficiency. However, when applied to singular perturbation problems, it remains difficult to effectively handle the sharp changes in the boundary layers.

Inspired by the work of BL-PINN, MAE and TransNet, in this paper we propose a matched asymptotic expansions-based transferable neural network (named as "MAE-TransNet") method for solving singular perturbation problems with boundary layers in general dimensions. Furthermore, we also develop a novel computational framework for solving coupled boundary layer problems based on MAE-TransNet, which addresses issues of lacking relevant matched asymptotic expansion theory in those problems. Our method integrates the traditional perturbation methods into neural network frameworks to enable the new network method to inherit the advantages of both approaches simultaneously. The proposed MAE-TransNet, rooted in the efficiency and accuracy of TransNet, achieves higher accuracy as the parameter ε decreases, consistent with the theory of matched asymptotic expansions.

Our method distinguishes itself from existing neural network approaches, such as the popular BL-PINN method, in the following ways: first, we compute both the inner and outer solutions across the entire computational domain. This contrasts with domain decomposition methods that separately compute inner expansions within boundary layer regions and outer expansions outside. Second, unlike approaches that simultaneously solve the solutions in both the inner and outer regions of the boundary layers, we first solve the outer solution over the entire computational domain, and then compute the inner solution with the matching principle. Third, instead of relying on PINNs, we employ the TransNet method, which utilizes a two-layer neural network with pre-determined hidden-layer neuron parameters to solve the simplified boundary value problems for inner and outer solutions. Fourth, diverging from methods that use identical networks for both the inner and outer solutions, we utilize nonuniformly and uniformly distributed hidden-layer neurons for the inner and outer solutions, respectively, to capture their distinct characteristics.

The rest of the paper is organized as follows. Section 2 briefly reviews the theory of MAE for singular perturbation problems and the TransNet method for solving PDEs. In Section 3, the MAE-TransNet method is developed with corresponding algorithms for both one-dimensional and multi-dimensional cases. Section 4 presents numerical results and comparisons on a series of benchmark experiments in various dimensional spaces, highlighting the superior performance of MAE-TransNet compared to TransNet, PINN and BL-PINN. Finally, Section 5 concludes this work and discusses some future research directions.

2. Related work

In this section, we present some related work including Prandtl's matched asymptotic expansions approach [34] and the TransNet method developed in [39], which collectively form the foundation of our methods.

2.1. Prandtl's matched asymptotic expansions

Prandtl's matched asymptotic expansions consist of a first-order outer expansion (*outer solution*) and a first-order inner expansion (*inner solution*). The outer expansion is valid outside the boundary layer and the inner expansion is valid in the sharp-change region, and then an approximate solution of the original singular perturbation problem problem (*composite solution*) can be written as a sum of the outer solution, the inner solution, and a matching term which cancels the outer expansion in the inner region and the inner expansion in the outer region.

For purpose of illustration, let us consider the following 1D singular perturbation problem in the domain $\Omega = (0, 1)$:

$$\begin{cases} \varepsilon \dfrac{\mathrm{d}^2 u(x)}{\mathrm{d}x^2} + \dfrac{\mathrm{d}u(x)}{\mathrm{d}x} + u(x) = 0, \quad x \in (0,1), \\ u(0) = 0, \quad u(1) = 1. \end{cases} \tag{2}$$

It is well known that one boundary layer occurs at $x = 0$ with the thickness ε for this problem. The outer, inner, composite, and exact solutions are denoted as $u^o(x), u^i(x), u^c(x)$, and $u_{exact}(x)$, respectively. As shown in Figure 1, the left plot presents the exact solutions of the problem (2) with different values of ε, while the right plot presents the outer, inner, and composite solutions for the problem (2) under $\varepsilon = 5 \times 10^{-3}$.

We first focus on the boundary value problem satisfied by the outer solution $u^o(x)$. Through analysis in [34], the boundary condition $u(0) = 0$ is to be be dropped, then as $\varepsilon \to 0$ the problem (2) reduces to the following boundary value problem:

$$\begin{cases} \dfrac{\mathrm{d}u^o(x)}{\mathrm{d}x} + u^o(x) = 0, \quad x \in (0,1), \\ u^o(1) = 1. \end{cases} \tag{3}$$

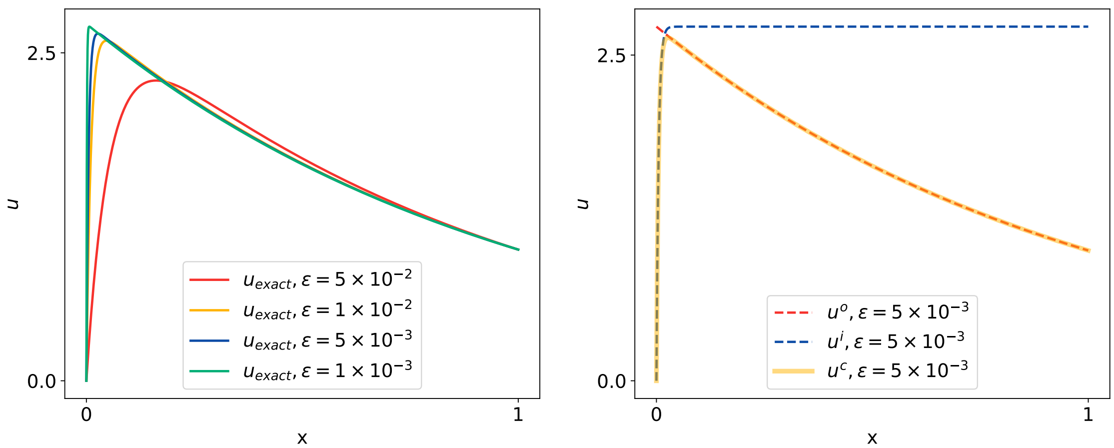

Figure 1: Left: The exact solutions of the 1D singular perturbation problem (2) with different values of ε. Right: The outer, inner and composite solutions for the problem (2) under $\varepsilon = 5 \times 10^{-3}$.

It is easy to find that the solution of problem (3) is given by

$$u^o(x) = e^{1-x}.$$

To obtain the inner solution $u^i(x)$, let us introduce the following scaling transformation to magnify the boundary layer

$$\zeta = \frac{x}{\delta(\varepsilon)} := \frac{x}{\varepsilon}, \tag{4}$$

where $\delta(\varepsilon) \approx O(\varepsilon)$ is usually determined by the settings of problem [34] and here we take $\delta = \varepsilon$. Then we have $u^i(x) = u^i(\varepsilon\zeta)$ and denote $\bar{u}^i(\zeta) = u^i(\varepsilon\zeta)$ with $\zeta \in [0, \frac{1}{\varepsilon}]$. Based on the problems (2) and (3), we know the boundary condition $\bar{u}^i(0) = 0$ should be imposed. In order to determine the other boundary condition, the *outer limit* (denoted by $u^{out}(x)$) and *inner limit* (denoted by $\bar{u}^{in}(\zeta)$) of a function $u(x;\varepsilon)$ are introduced:

$$u^{out}(x) = \lim_{\varepsilon \to 0,\ x \text{ fixed}} u(x;\varepsilon),$$
$$\bar{u}^{in}(\zeta) = \lim_{\varepsilon \to 0,\ \zeta \text{ fixed}} u(\varepsilon\zeta;\varepsilon),$$

and then *matching principle* is defined by

$$\lim_{x \to 0} u^{out}(x) = \lim_{\zeta \to \infty} \bar{u}^{in}(\zeta). \tag{5}$$

From the matching principle (5), the inner limit of the outer solution should be equal to the outer limit of the inner solution and we get the second boundary condition

$$\lim_{\varepsilon \to 0} \bar{u}^i(\frac{1}{\varepsilon}) = u^o(0) = e.$$

Therefore, as $\varepsilon \to 0$, the problem (2) after the scaling transformation by $1/\varepsilon$ reduces to the following boundary value problem:

$$\begin{cases} \dfrac{\mathrm{d}^2\bar{u}^i(\zeta)}{\mathrm{d}\zeta^2} + \dfrac{\mathrm{d}\bar{u}^i(\zeta)}{\mathrm{d}\zeta} = 0, \quad \zeta \in (0, \dfrac{1}{\varepsilon}), \\ \bar{u}^i(0) = 0, \quad \bar{u}^i(\dfrac{1}{\varepsilon}) = e. \end{cases} \tag{6}$$

Note the computational domain for $\bar{u}^i(\zeta)$ is $\Omega^\zeta = (0, \frac{1}{\varepsilon})$. It can be found that the solution of problem (6) is given by

$$\bar{u}^i(\zeta) = e - e^{1-\zeta} = e - e^{1-x/\varepsilon} = u^i(x). \tag{7}$$

Finally, a uniformly valid solution called composite solution $u^c(x)$ is obtained:

$$u^c(x) = u^o(x) + (u^i(x) - (u^o)^{in}(x)), \quad x \in [0, 1], \tag{8}$$

which is regarded as an approximate solution of the exact solution $u(x)$. For the problem (2), its composite solution is given by $u^c(x) = e^{1-x} - e^{1-x/\varepsilon}$ since $(u^o)^{in}(x) = e$ for any $x \in [0, 1]$. The error between $u^c(x)$ and $u(x)$ achieves a magnitude of $O(\varepsilon)$ over the entire computational domain [34], which implies that the accuracy improves as ε decreases.

Matched asymptotic expansions overcome the difficulty of straightforward expansions in simultaneously addressing sharp and gentle gradients by employing a magnified scale to capture steep changes and the original scale to characterize features outside the sharp-change regions. Thus, the matched asymptotic expansions have been used to solve various complex problems involving sharp gradients in solutions [40, 41]. However, the major drawback of the matched asymptotic expansions method lies in the necessity to explicitly solve both the outer and inner solutions, which can be impossible for a majority of problems [38]. Transferable neural network is one of the efficient numerical methods for addressing these issues.

2.2. *Transferable Neural Networks*

The TransNet [39] is a fully-connected two-layer neural network (one hidden layer and one output layer) with pre-trained hidden-layer neurons for solving PDEs of the following general formulation:

$$\begin{cases} \mathcal{L}(\boldsymbol{u}) = f, & \boldsymbol{x} \in \Omega, \\ \mathcal{B}(\boldsymbol{u}) = g, & \boldsymbol{x} \in \partial\Omega, \end{cases} \tag{9}$$

where Ω denotes the spatial domain in $\mathbb{R}^d$ (or the spatial-temporal domain in $\mathbb{R}^d \times (0, T]$ for time-dependent PDE problems). As shown in Figure 2, The computational procedure of the TransNet method consists of two stages: generating pre-trained hidden layer parameters, and optimizing the parameters of the output layer.

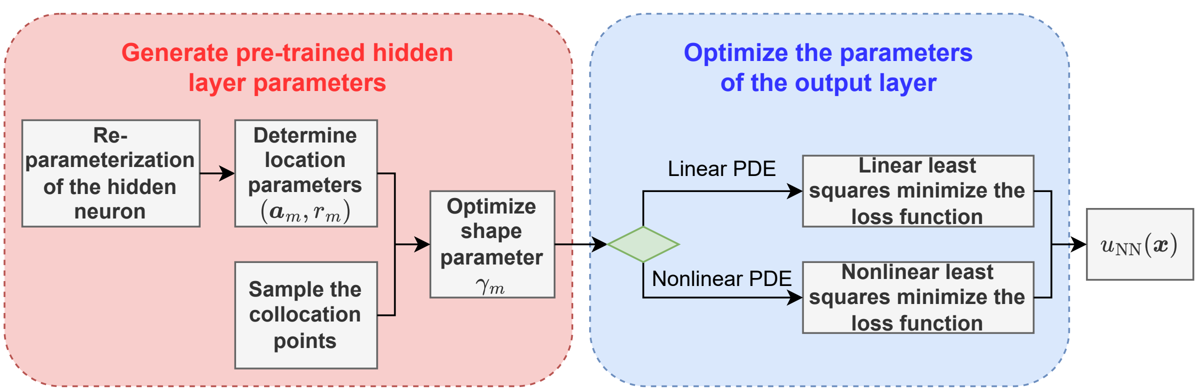

Figure 2: The flowchart of the transferable neural network method for solving PDEs.

In the first stage, the pre-trained model is the neural feature space $\mathcal{P}_{\text{NN}}$, expanded by hidden-layer neurons (or called neural basis functions) $\left\{\sigma\left(\boldsymbol{w}_m^\top \boldsymbol{x} + b_m\right)\right\}_{m=1}^M$:

$$\mathcal{P}_{\text{NN}} = \text{span}\left\{1, \sigma\left(\boldsymbol{w}_1^\top \boldsymbol{x} + b_1\right), \ldots, \sigma\left(\boldsymbol{w}_M^\top \boldsymbol{x} + b_M\right)\right\}, \tag{10}$$

where M is the number of hidden-layer neurons, $\sigma(\cdot)$ is the activation function, and $\boldsymbol{w}_m \in \mathbb{R}^d, b_m \in \mathbb{R}$ is the weight and bias of the m-th hidden-layer neuron. To achieve the uniform distribution of neurons in $\mathcal{P}_{\text{NN}}$, the initial step involves the re-parameterization of the hidden-layer neuron as:

$$\sigma\left(\boldsymbol{w}_m^\top \boldsymbol{x} + b_m\right) = \sigma\left(\gamma_m\left(\boldsymbol{a}_m^\top \boldsymbol{x} + r_m\right)\right),$$

where $\boldsymbol{a}_m \in \mathbb{R}^d$, $\|\boldsymbol{a}_m\|_2 = 1$ is the unit normal vector of the partition hyperplane corresponding to the m-th hidden-layer neuron, i.e.,

$$\boldsymbol{w}_m^\top \boldsymbol{x} + b_m = 0.$$

Here, $r_m \in \mathbb{R}$, $|r_m|$ represents the distance from the origin to the partition hyperplane, $(\boldsymbol{a}_m, r_m)$ are named as location parameters, while $\gamma_m \in \mathbb{R}_+$ is the shape parameter. Let us take $\sigma(\cdot) = \tanh(\cdot)$ throughout this paper, which is widely used for solving PDEs, the shape parameter γ_m controls the transition layer width of the activation function near the partition hyperplane. *Location parameter* $\{\boldsymbol{a}_m\}_{m=1}^M$ *are i.i.d., uniformly distributed on the d-dimensional unit sphere and* $\{r_m\}_{m=1}^M$ *are i.i.d., uniformly distributed in* $[0, 1]$. For simplicity, it is usually assumed that all hidden-layer neurons share the same shape parameter, i.e., $\gamma_m = \gamma$. The next step is then to optimize shape parameter γ through the residuals at the collocation points for approximating Gaussian random fields, which can be achieved by some simple line search algorithms (see [39] for details). At this stage, the construction of $\mathcal{P}_{\text{NN}}$ is completed. To simplify notation, we assume that solution $\boldsymbol{u}$ in (9) is a scalar function, then the network solution $u_{\text{NN}} \in \mathcal{P}_{\text{NN}}$ is expressed as

$$u_{\text{NN}}(\boldsymbol{x}) = \sum_{m=1}^{M} \alpha_m \sigma\left(\boldsymbol{w}_m^\top \boldsymbol{x} + b_m\right) + \alpha_0, \tag{11}$$

where $\alpha_1, \alpha_2, \ldots, \alpha_M$ are the weights of the output layer, and α_0 is the bias of that.

In the second stage, the parameters of the hidden layer are frozen, the weights and bias of the output layer, i.e., $\alpha_0, \alpha_1, \ldots, \alpha_M$ in (11) can be obtained through least squares solvers, which minimizes the PDE residual loss function:

$$\text{Loss} = \|\mathcal{L}(u(\boldsymbol{x})) - \mathcal{L}(u_{\text{NN}}(\boldsymbol{x}))\|_2^2 + \|\mathcal{B}(u(\boldsymbol{x})) - \mathcal{B}(u_{\text{NN}}(\boldsymbol{x}))\|_2^2. \tag{12}$$

Algorithm 1 summarizes the main steps of the TransNet method. In this paper, the optimal γ for each example can be efficiently determined via the golden section search with low computational cost as done in [42], and thus for simplicity, the value of γ is directly provided as an input.

Wide ranges of numerical experiments indicate that TransNet has much higher accuracy (numerically exponential-type convergence with respect the number of neurons) and lower computing cost compared to PINN for various PDE problems with relatively smooth solutions [39]. However, numerical experiments in Section 4 reveal TransNet's frequent inability to capture the highly sharp variations occurring within boundary layers of singular perturbation problems. In this paper, we address this limitation by employing matched asymptotic expansions to handle singular perturbation problems with boundary layers, and TransNet is then subsequently used to solve the corresponding simplified problems for the inner and outer solutions.

Algorithm 1: (TransNet method for solving PDE problems [39])

Input: The shape parameter γ and the number of hidden-layer neurons M

Output: The TransNet solution u_{NN}

1 Generate the location parameters $\{(\boldsymbol{a}_m, r_m)\}_{m=1}^{M}$ on Ω as the process described in [39] and form the set of hidden-layer neurons $\{\sigma\left(\gamma\left(\boldsymbol{a}_m^{\top}\boldsymbol{x} + r_m\right)\right)\}_{m=1}^{M}$ (uniformly distributed in Ω).

2 Generate collocation points in Ω and on $\partial\Omega$ by uniform sampling.

3 Optimize the parameters of the output layer $\boldsymbol{\alpha}$ by minimizing the loss function (12), which is associated with the PDE problem (9), over the collocation points using certain least squares solver.

4 Return u_{NN} based on (11).

3. The proposed method – matched asymptotic expansions-based transferable neural network

The TransNet method meets significant challenges when dealing with singular perturbation problems, as it struggles to capture large gradient information within the boundary layer. In this section, we present the MAE-TransNet method for both one-dimensional and multi-dimensional singular perturbation problems with boundary layers, which could effectively resolve such issue.

3.1. 1D singular perturbation problems with boundary layers

In this subsection, for 1D singular perturbation problems, we first present how the MAE-TransNet method captures a single boundary layer and subsequently extend the method to multiple boundary layers.

3.1.1. The case of single boundary layer

For purpose of illustration, let us again consider the 1D singular perturbation problem (2), whose solution has a boundary layer with width $O(\varepsilon)$ close the point $x = 0$. As shown in Figure 3, the schematic diagram of MAE-TransNet is divided into three modules: analysis, computation, and composition.

In the analysis module, the boundary value problem (3) for the outer solution $u^o(x)$ (defined in $\overline{\Omega} = [0, 1]$) is first derived, while the scaling transformation $\zeta = x/\delta(\varepsilon)$ (see (4) with $\delta(\varepsilon) = \varepsilon$) is employed to magnify the boundary layer and construct the corresponding boundary value problem (6) (defined in the scaled domain $\overline{\Omega^\zeta} = [0, 1/\varepsilon]$) for the inner solution $\bar{u}^i(\zeta)$. In the computation module, the variation of the $u^o(x)$ in Ω is relatively gentle, thus the TransNet with uniformly distributed hidden-layer neurons in Ω is able to yield an accurate network outer solution $u^o_{\text{NN}}(x)$. For $\bar{u}^i(\zeta)$, leveraging the existing u^o_{NN} and the matching principle outlined in (5), the inner limit of the outer solution $(u^o_{\text{NN}})^{in}$ is integrated to the problem (6) as a supplementary boundary condition. It is noteworthy that since the gradient information of $\bar{u}^i(\zeta)$ concentrates within the scaled boundary layer, inspired by [43], the nonuniformly distributed hidden-layer neurons are employed to compute the network inner solution $\bar{u}^i_{\text{NN}}(\zeta)$, in which the hidden-layer neurons are uniformly distributed within a small spherical region covering only the scaled boundary layer region (such as $(0, 1) \subset \Omega^\zeta$ in this case) rather than the entire scaled computational domain Ω^ζ. In the composite module, the network composite solution u^c_{NN} is finally obtained through the combination of u^o_{NN}, u^i_{NN} and the matching term $(u^o_{\text{NN}})^{in}$ according to (8).

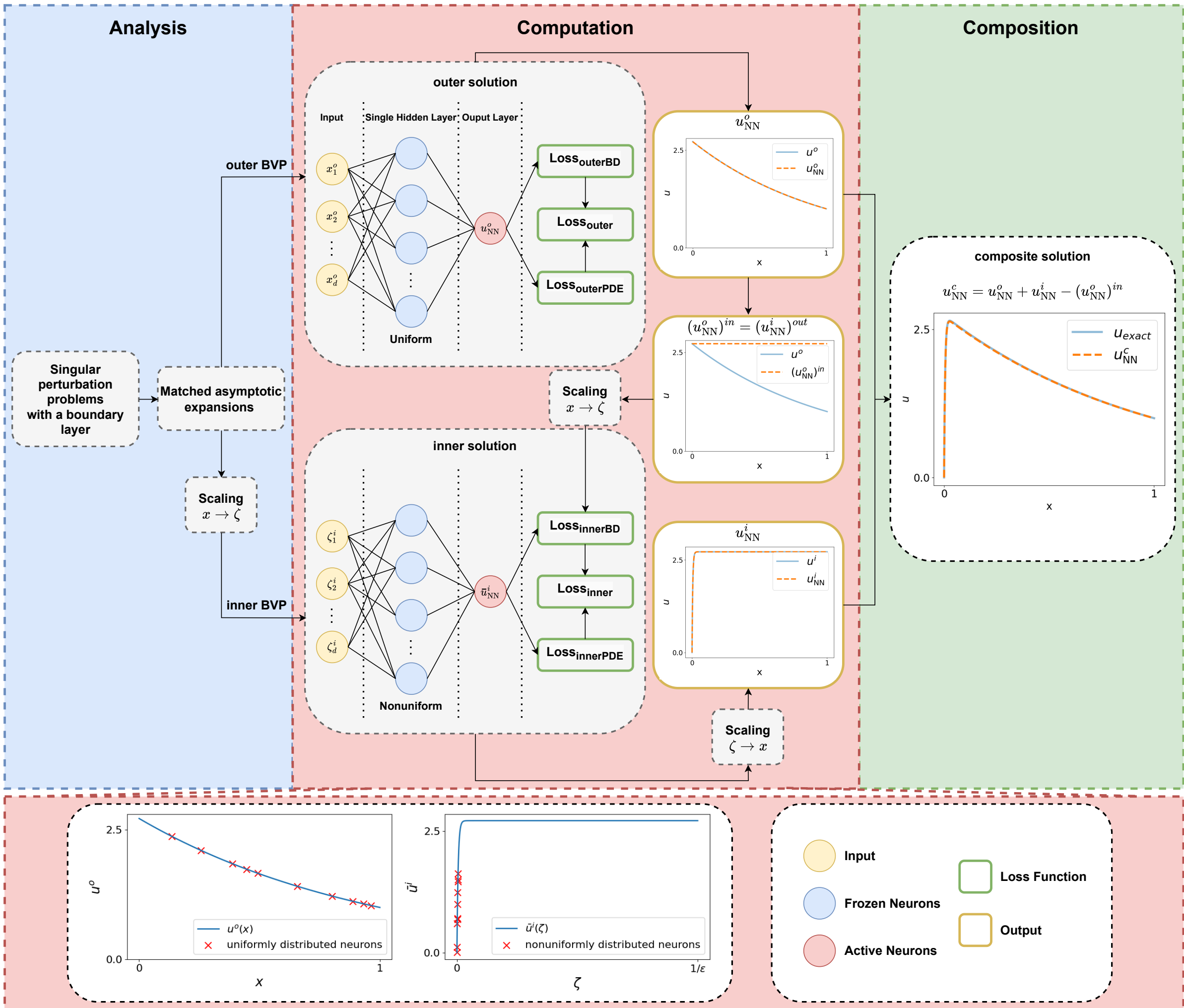

Figure 3: A schematic diagram of MAE-TransNet for 1D singular perturbation problems with single boundary layer.

As illustrated in Figure 4, the left panel presents two sets of hidden-layer neurons under different values of location parameters (a, r) in the scaled domain Ω^ζ, where the width of the transition region get larger along the decrease of the shape parameter γ. The right panel compares two distinct hidden-layer neuron distributions across Ω^ζ for solving the inner solution $\bar{u}^i(\zeta)$. A uniform neuron distribution in Ω^ζ fails to capture the large gradient information within the scaled boundary layer due to the mild variation of most neurons in this region. In contrast, a nonuniform distribution concentrates neurons inside the scaled boundary layer region of Ω^ζ, enabling effective resolution of rapid variations. Traditional methods such as FEM typically rely on local basis functions. When confined strictly within the boundary layer, these functions fail to capture information outside this region. However, since the inner solution remains approximately constant outside the boundary layer, and the neurons act as global basis functions that are nearly constant beyond the transition layer, neurons distributed within the scaled boundary layer can effectively capture the solution characteristics outside this region.

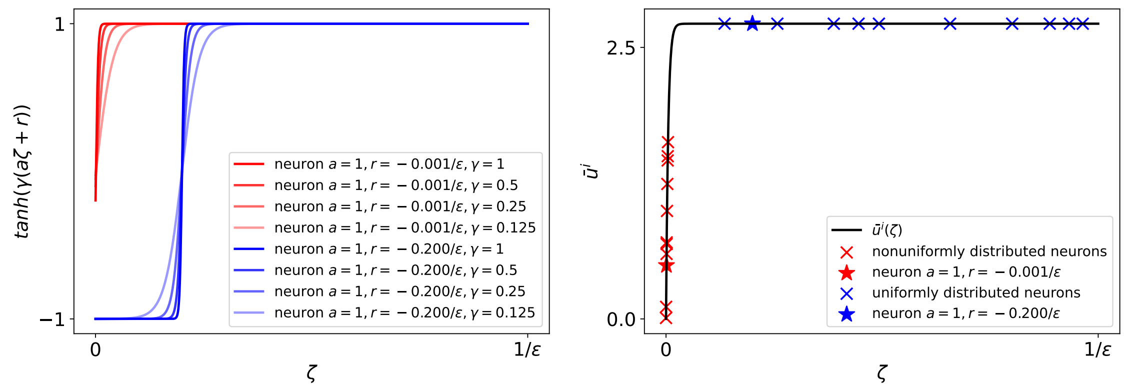

Figure 4: Left: Schematic illustration of the neuron geometry structure under two location parameters (a, r) and various shape parameter γ in the computational domain Ω^{ζ}. Right: Comparison of two distinct hidden-layer neuron distributions across Ω^{ζ}.

Remark 1. *Van Dyke's matching principle states that the m-term inner expansion of the n-term outer expansion equals the n-term outer expansion of the m-term inner expansion [34]. By incorporating Van Dyke's matching rule as a soft constraint into the loss function, even higher-order MAE-TransNet methods potentially could be constructed.*

3.1.2. The case of multiple boundary layers

Since the matched asymptotic expansion approach is amenable to singular perturbation problems with multiple boundary layers, the proposed MAE-TransNet method can be easily generalized to solve such problems. The sole modification involves reformulating (8) into the following equation:

$$u^c(x) = u^o(x) + \sum_{k=1}^{K} (u_k^i(x) - (u^o)_k^{in}(x)), \quad x \in \Omega, \tag{13}$$

where $K \geq 2$ represents the number of boundary layers, u_k^i denotes the inner solution of the k-th boundary layer, and $(u^o)_k^{in}$ signifies the inner limit of u^o at the k-th boundary layer. For illustration, let us consider the following 1D singular perturbation problem in the domain $\Omega = (0, 1)$ [44]:

$$\begin{cases} \varepsilon^2 \dfrac{\mathrm{d}^2 u(x)}{\mathrm{d}x^2} + \varepsilon x \dfrac{\mathrm{d}u(x)}{\mathrm{d}x} - u(x) = -e^x, \quad x \in (0, 1), \\ u(0) = 2, \quad u(1) = 1. \end{cases} \tag{14}$$

This problem involves two boundary layers, one is at the point $x = 0$ and the other is at $x = 1$ with the same thickness ε. As shown in Figure 5, the left plot presents the reference solutions u_{ref} of the problem (14) for different values of ε, while the right plot presents the outer, inner, and composite solutions for the problem (14) under $\varepsilon = 1 \times 10^{-3}$.

As $\varepsilon \to 0$, the problem (14) reduces to the following outer solution problem:

$$-u^o(x) = -e^x, \quad x \in (0, 1), \tag{15}$$

and this directly gives the outer solution $u^o(x) = e^x$, which cannot satisfy either boundary conditions $u(0) = 2$ and $u(1) = 1$. To establish the inner solution $u_1^i(x)$ at $x = 0$, the scaling

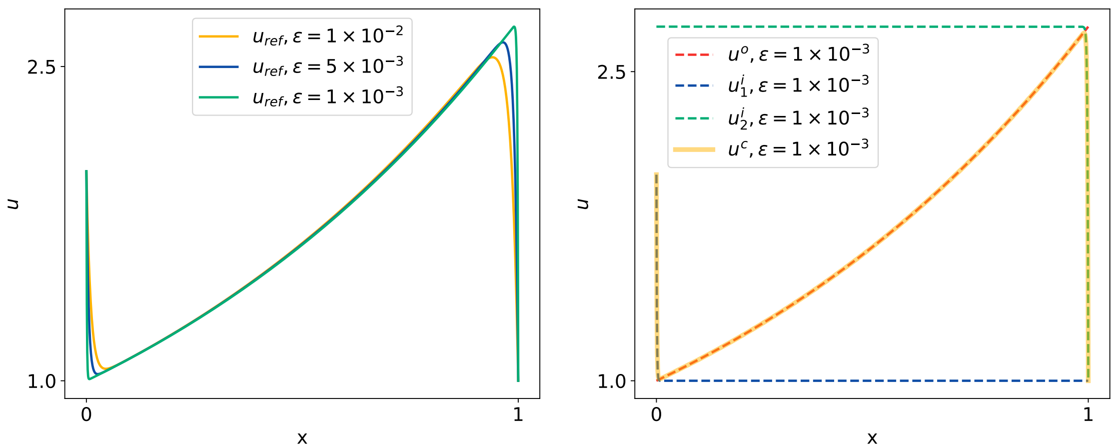

Figure 5: Left: The reference solutions of the 1D singular perturbation problem (14) with different values of ε. Right: The outer, inner and composite solutions for the problem (14) under $\varepsilon = 1 \times 10^{-3}$.

transformation $\zeta = \frac{x}{\delta_1(\varepsilon)} := \frac{x}{\varepsilon}$, $\varepsilon \to 0$ and the matching principle are used to obtain the following boundary value problem:

$$\begin{cases} \dfrac{\mathrm{d}^2 \bar{u}_1^i(\zeta)}{\mathrm{d}\zeta^2} - \bar{u}_1^i(\zeta) = -1, \quad \zeta \in (0, \dfrac{1}{\varepsilon}), \\ \bar{u}_1^i(0) = 2, \quad \bar{u}_1^i(\dfrac{1}{\varepsilon}) = u^o(0). \end{cases} \tag{16}$$

Similarly, for the inner solution $u_2^i(x)$ at $x = 1$, $\zeta = \frac{1-x}{\delta_2(\varepsilon)} := \frac{1-x}{\varepsilon}$, $\varepsilon \to 0$ and the matching principle are used to obtain the following boundary value problem:

$$\begin{cases} \dfrac{\mathrm{d}^2 \bar{u}_2^i(\zeta)}{\mathrm{d}\zeta^2} - \dfrac{\mathrm{d}\bar{u}_2^i(\zeta)}{\mathrm{d}\zeta} - \bar{u}_2^i(\zeta) = -e, \quad \zeta \in (0, \dfrac{1}{\varepsilon}), \\ \bar{u}_2^i(0) = 1, \quad \bar{u}_2^i(\dfrac{1}{\varepsilon}) = u^o(1). \end{cases} \tag{17}$$

Note the computational domain for $\bar{u}_1^i(\zeta)$ and $\bar{u}_2^i(\zeta)$ is $\Omega_1^\zeta = \Omega_2^\zeta = (0, \frac{1}{\varepsilon})$. According to (13) with $K = 2$, the composite solution of problem (14) is then given by

$$u^c = u^o + u_1^i - (u^o)_1^{in} + u_2^i - (u^o)_2^{in}.$$

As shown in Figure 6, in the analysis module, $u^o(x)$ is directly obtained for $x \in \Omega$ and the scaling transformations are introduced to magnify the two boundary layers. In the computation module, the network inner solutions $(\bar{u}_1^i)_{\mathrm{NN}}(\zeta)$ from (16) and $(\bar{u}_2^i)_{\mathrm{NN}}(\zeta)$ from (17) are computed in the domain Ω^ζ. In the composite module, $u_{\mathrm{NN}}^c(x)$ is finally determined through (13).

Finally, we summarize in Algorithm 2 the implementation details of the proposed MAE-TransNet for 1D singular perturbation problems with $K \geq 1$ boundary layers.

3.2. *Multi-dimensional singular perturbation problems with boundary layers*

In this subsection, we extend the proposed MAE-TransNet method from one-dimensional cases to multi-dimensional cases. For simplicity, we illustrate the MAE-TransNet method for multi-dimensional cases through 2D problems.

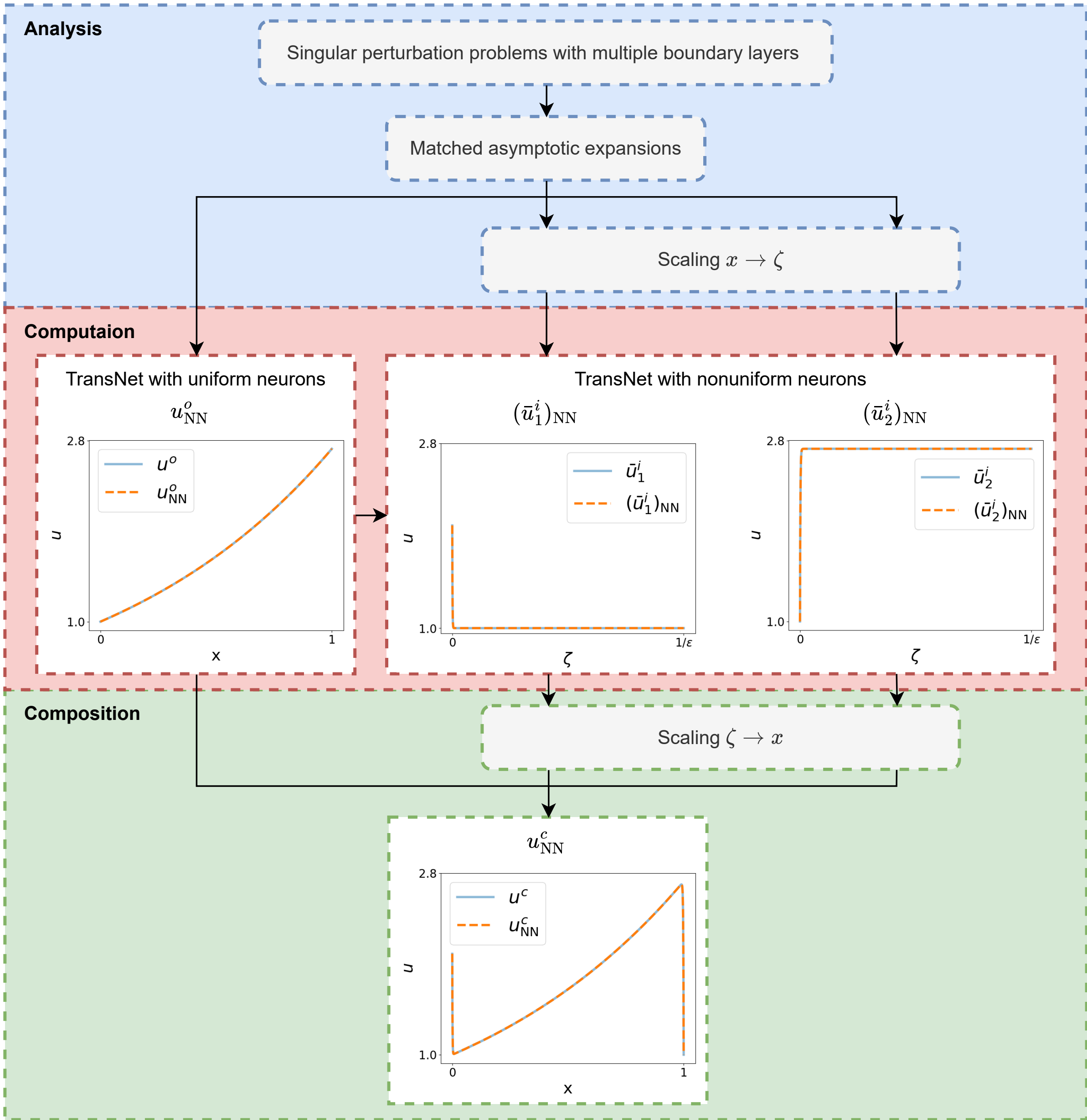

Figure 6: A schematic diagram of MAE-TransNet for 1D singular perturbation problems with $K = 2$ boundary layers.

3.2.1. The case of single boundary layer case

Let us consider the 2D Couette flow problem (advection-diffusion transport) in the domain $\Omega = (0,1)^2$ [27]:

$$\begin{cases} 10y\dfrac{\partial u(x,y)}{\partial x} - \varepsilon\left(\dfrac{\partial^2 u(x,y)}{\partial x^2} + \dfrac{\partial^2 u(x,y)}{\partial y^2}\right) = 0, \quad (x,y) \in (0,1)^2, \\ u(x=0,y) = 0, \quad \dfrac{\partial u}{\partial y}(x, y=0) = -10, \quad \dfrac{\partial u}{\partial y}(x=1,y) = \dfrac{\partial u}{\partial x}(x, y=1) = 0, \end{cases} \tag{18}$$

Algorithm 2: (MAE-TransNet method for 1D singular perturbation problems)

Input: The number of boundary layers K, the shape parameters $\{\gamma^o, \gamma_1^i, \cdots, \gamma_K^i\}$ and the number of hidden-layer neurons $\{M^o, M_1^i, \cdots, M_K^i\}$ for outer solution and inner solutions respectively, the scaling transformation factor $\{\delta_1(\varepsilon), \cdots, \delta_K(\varepsilon)\}$ for inner solutions.

Output: The MAE-TransNet composite solution u_{NN}^c.

Form the outer solution problem defined over Ω and compute the network outer solution u_{NN}^o by feeding γ^o and M^o into Algorithm 1.

for $j = 1 : K$ **do**

  Find the scaled domain Ω_j^ζ with the scaling factor $1/\delta_j(\varepsilon)$ and correspondingly form the j-inner solution problem defined over Ω_j^ζ.

  Construct the set of hidden-layer neurons $\left\{\sigma\left(\gamma_j^i\left(a_m^\top \zeta + r_m\right)\right)\right\}_{m=1}^{M_j^i}$ which are uniformly distributed only within the scaled j-boundary layer region of Ω_j^ζ.

  Generate collocation points in Ω_j^ζ and on $\partial\Omega_j^\zeta$ by uniform sampling.

  Optimize the parameters $\boldsymbol{\alpha}_j^i$ by minimizing the loss function (12) associated with the j-inner solution problem over the collocation points using a least squares solver to obtain the network j-inner solution $(\bar{u}_j^i)_{\mathrm{NN}}$.

end

Compute and return u_{NN}^c based on (13).

which is a typical 2D singular perturbation problem with single boundary layer at $y = 0$. The reference solution under $\varepsilon = 1 \times 10^{-4}$ is shown in Figure 7.

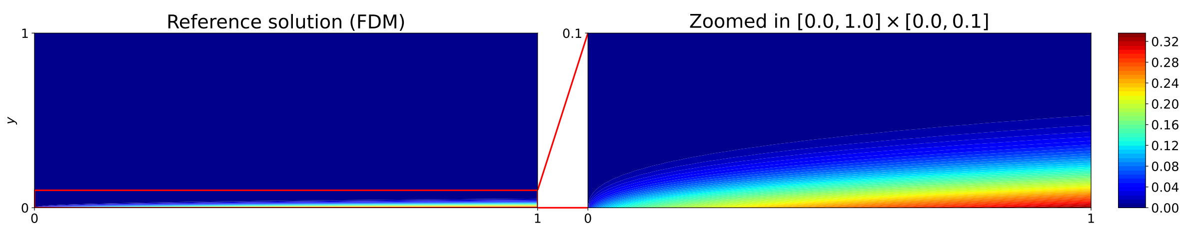

Figure 7: The reference solution of the 2D Couette flow problem (18) under $\varepsilon = 1 \times 10^{-4}$. Left: Over the whole domain Ω. Right: Zoomed in $[0, 1] \times [0, 0.1]$.

As $\varepsilon \to 0$, the problem (14) reduces to the following outer solution problem:

$$\frac{\partial u(x, y)}{\partial x} = 0, \quad (x, y) \in (0, 1)^2, \tag{19}$$

and with the boundary condition $u(x = 0, y) = 0$, this directly gives the outer solution $u^o(x, y) = 0$. To establish the inner solution u^i, the scaling transformation $\eta = \frac{y}{\delta(\varepsilon)} := \frac{y}{\sqrt{\varepsilon}}$ (i.e., asymptotic expansion in y) and the matching principle are used to obtain the following boundary value problem:

$$\begin{cases} 10\sqrt{\varepsilon}\eta \dfrac{\partial \bar{u}^i(x,\eta)}{\partial x} = \dfrac{\partial^2 \bar{u}^i(x,\eta)}{\partial \eta^2}, \quad (x,\eta) \in (0,1)\times(0,\dfrac{1}{\sqrt{\varepsilon}}), \\ \bar{u}^i(x=0,\eta) = 0, \quad \dfrac{\partial \bar{u}^i}{\partial \eta}(x,\eta=0) = -10\sqrt{\varepsilon}, \quad \bar{u}^i(x,\eta=\dfrac{1}{\sqrt{\varepsilon}}) = 0, \end{cases} \tag{20}$$

where $\bar{u}^i(x,\eta) = u^i(x,\sqrt{\varepsilon}\eta) = u^i(x,y)$. Subsequently, the composite solution u^c_{NN} is constructed through (8). The implementation details can be similarly referred to Algorithm 2 with $K = 1$ in Subsection 3.1.2, thus are skipped here.

3.2.2. The case of coupled boundary layers

Due to the absence of matched asymptotic expansion theory for coupled boundary layer problems, the direct deployment of MAE-TransNet could result in significant errors in the coupled region. To address this issue, an auxiliary TransNet in the coupled region is combined with MAE-TransNet over the whole domain, thereby improving its capability to handle multi-dimensional coupled boundary layer problems.

To illustrate how MAE-TransNet can be employed in this case, let us consider the following 2D coupled boundary layer problem [45] in the domain $\Omega = (0,1)^2$:

$$\begin{cases} -\varepsilon\left(\dfrac{\partial^2 u(x,y)}{\partial x^2} + \dfrac{\partial^2 u(x,y)}{\partial y^2}\right) - (x+2)\dfrac{\partial u(x,y)}{\partial x} \\ \qquad\qquad - \left(y^3+3\right)\dfrac{\partial u(x,y)}{\partial y} + u(x,y) = f(x,y), \quad (x,y)\in(0,1)^2, \\ u(x=0,y)=0, \quad u(x=1,y)=0, \quad u(x,y=0)=0, \quad u(x,y=1)=0, \end{cases} \tag{21}$$

where $f(x,y)$ is chosen appropriately so that the exact solution is

$$u(x,y) = \cos\left(\frac{\pi x}{2}\right)\left(1-e^{-2x/\varepsilon}\right)\left(1-y^3\right)\left(1-e^{-3y/\varepsilon}\right).$$

The exact solution under $\varepsilon = 2^{-6}$ is shown in Figure 8. Figure 9 presents our proposed computational framework, in which a MAE-TransNet is implemented for finding the full-domain solution in Ω, and an extra auxiliary TransNet is used for correction in the coupled region $\Omega^{ii} = (0,A)^2$ with $\varepsilon \le A \ll 1$.

Over the whole domain, as $\varepsilon \to 0$, the problem (21) is simplified and the outer solution u^o satisfies the following boundary value problem:

$$\begin{cases} -(x+2)\dfrac{\partial u^o(x,y)}{\partial x} - \left(y^3+3\right)\dfrac{\partial u^o(x,y)}{\partial y} + u^o(x,y) = f^o(x,y), \quad (x,y)\in(0,1)^2, \\ u^o(x=1,y)=0, \quad u^o(x,y=1)=0, \end{cases} \tag{22}$$

where

$$f^o(x,y) = \left(1-y^3\right)\cos\frac{\pi x}{2} + (x+2)\frac{\pi}{2}\left(1-y^3\right)\sin\frac{\pi x}{2} + \left(y^3+3\right)3y^2\cos\frac{\pi x}{2}.$$

In order to capture the two inner solutions along the x-axis (i.e., u^i_1) and the y-axis (i.e., u^i_2) respectively, the scaling transformation $\zeta = \frac{x}{\delta_1(\varepsilon)} := \frac{x}{\varepsilon}$ (i.e., asymptotic expansion in x) and $\eta =$

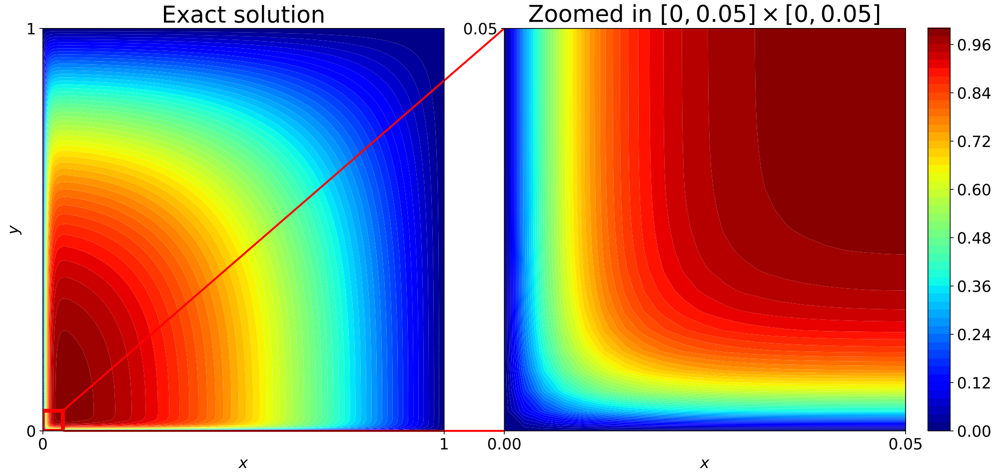

Figure 8: The exact solution of the 2D singular perturbation problem (21) with coupled boundary layers under $\varepsilon = 2^{-6}$. Left: Over the whole domain Ω. Right: Zoomed in $[0, 0.05] \times [0, 0.05]$.

$\frac{y}{\delta_2(\varepsilon)} := \frac{y}{\varepsilon}$ (i.e, asymptotic expansion in y), and the matching principle are used to obtain the following two boundary value problems:

$$\begin{cases} -\dfrac{\partial^2 \bar{u}_1^i(\zeta, y)}{\partial \zeta^2} - 2\dfrac{\partial \bar{u}_1^i(\zeta, y)}{\partial \zeta} = 0, \quad (\zeta, y) \in (0, \dfrac{1}{\varepsilon}) \times (0, 1), \\ \bar{u}_1^i(\zeta = 0, y) = 0, \quad \bar{u}_1^i(\zeta = \dfrac{1}{\varepsilon}, y) = u^o(x = 0, y), \quad \bar{u}_1^i(\zeta, y = 1) = 0, \end{cases} \tag{23}$$

for solving $\bar{u}_1^i(\zeta, y)$ in $\Omega_1^\zeta = (0, \frac{1}{\varepsilon}) \times (0, 1)$, and

$$\begin{cases} -\dfrac{\partial^2 \bar{u}_2^i(x, \eta)}{\partial \eta^2} - 3\dfrac{\partial \bar{u}_2^i(x, \eta)}{\partial \eta} = 0, \quad (x, \eta) \in (0, 1) \times (0, \dfrac{1}{\varepsilon}), \\ \bar{u}_2^i(x, \eta = 0) = 0, \quad \bar{u}_2^i(x, \eta = \dfrac{1}{\varepsilon}) = u^o(x, y = 0), \quad \bar{u}_2^i(x = 1, \eta) = 0. \end{cases} \tag{24}$$

for solving $\bar{u}_2^i(x, \eta)$ in $\Omega_2^\eta = (0, 1) \times (0, \frac{1}{\varepsilon})$. The MAE-TransNet method described in Algorithm 2 with $K = 2$ in the Subsection 3.1.2 can be similarly employed to obtain u_{NN}^o from (22), $(\bar{u}_1^i)_{\mathrm{NN}}$ from (23), $(\bar{u}_2^i)_{\mathrm{NN}}$ from (24), and u_{NN}^c from (13).

In the coupled region $\Omega^{ii} = (0, A)^2$, the two scaling transformations mentioned above are utilized simultaneously, as special solution in $\Omega^{\zeta,\eta} = (0, \frac{A}{\varepsilon})^2$ is denoted as $\tilde{u}^{ii}(\zeta, \eta) = u^{ii}(x, y)$ which satisfies the following boundary value problem:

$$\begin{cases} -\left(\dfrac{\partial^2 \tilde{u}^{ii}(\zeta, \eta)}{\partial \zeta^2} + \dfrac{\partial^2 \tilde{u}^{ii}(\zeta, \eta)}{\partial \eta^2}\right) - 2\dfrac{\partial \tilde{u}^{ii}(\zeta, \eta)}{\partial \zeta} - 3\dfrac{\partial \tilde{u}^{ii}(\zeta, \eta)}{\partial \eta} = 0, \quad (\zeta, \eta) \in (0, \dfrac{A}{\varepsilon})^2, \\ \tilde{u}^{ii}(\zeta = 0, \eta) = 0, \quad \tilde{u}^{ii}(\zeta = \dfrac{A}{\varepsilon}, \eta) = u^c(x = A, \varepsilon\eta), \\ \tilde{u}^{ii}(\zeta, \eta = 0) = 0, \quad \tilde{u}^{ii}(\zeta, \eta = \dfrac{A}{\varepsilon}) = u^c(\varepsilon\zeta, y = A), \end{cases} \tag{25}$$

where the top and right boundary conditions are supplied by u_{NN}^c found in the previous step. Subsequently, a TransNet with uniformly distributed hidden-layer neurons is employed to solve the problem (25) for the network solution in the coupled region, denoted as $\tilde{u}_{\mathrm{NN}}^{ii}$.

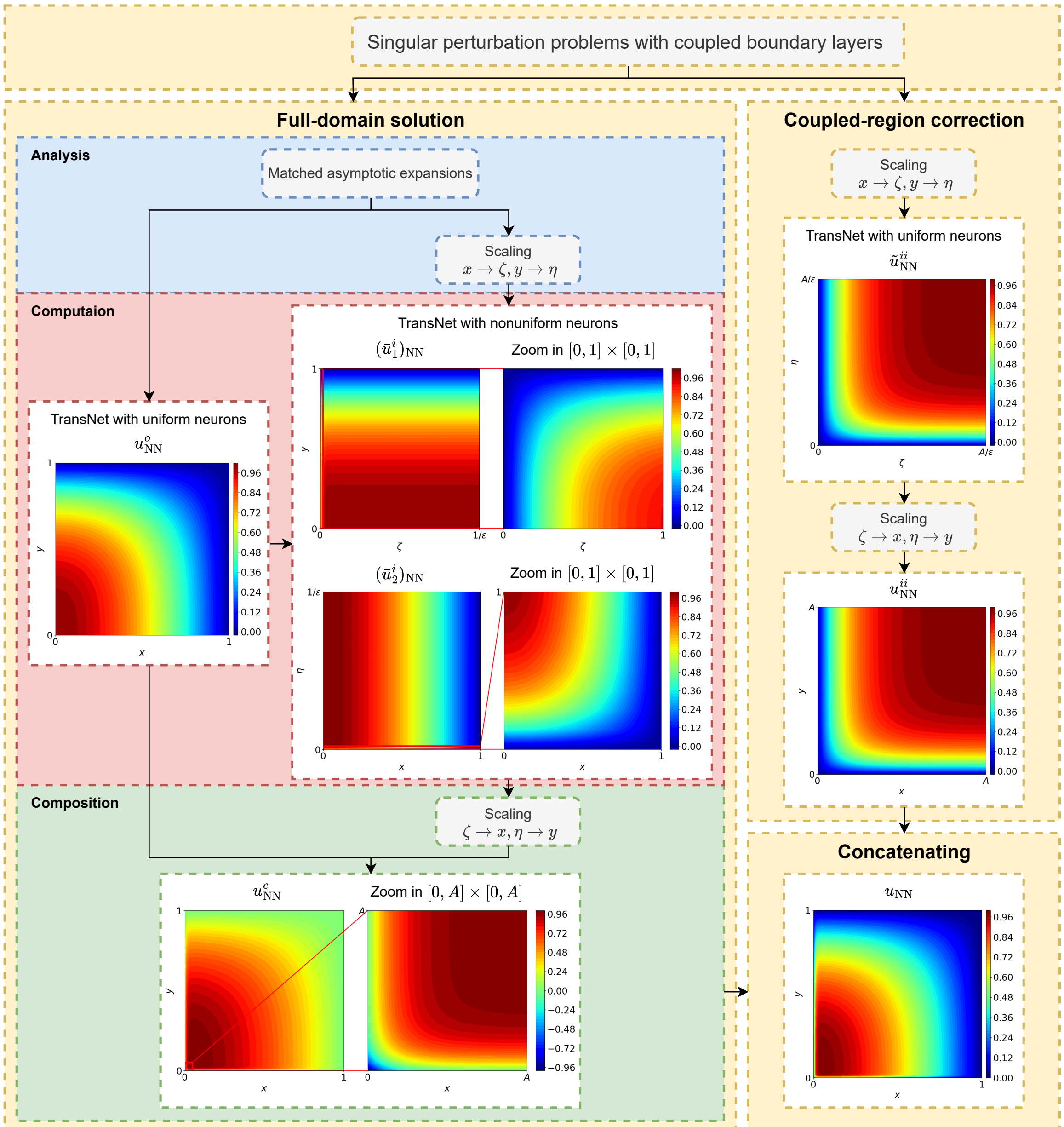

Figure 9: A schematic diagram of MAE-TransNet for 2D singular perturbation problems with coupled boundary layers. During the concatenating stage, u_{NN} is synthesized through u^c_{NN} in $[0,1]^2 \setminus [0,A]^2$ and u^{ii}_{NN} in $[0,A]^2$.

Finally, by concatenating u^c_{NN} in $\Omega \setminus \Omega^{ii}$ and u^{ii}_{NN} in Ω^{ii}, a consistently valid solution u_{NN} across the entire computational domain is obtained. Algorithm 3 presents the implementation details of the proposed MAE-TransNet for multi-dimensional singular perturbation problems with coupled boundary layers.

Algorithm 3: (MAE-TransNet method for multi-dimensional singular perturbation problems with coupled boundary layers)

Input: The number of boundary layers K, the shape parameters $\{\gamma^o, \gamma_1^i, \gamma_2^i, \ldots, \gamma_K^i, \gamma^{ii}\}$ and the number of hidden-layer neurons $\{M^o, M_1^i, M_2^i, \ldots, M_K^i, M^{ii}\}$ for outer solution, inner solution and solution in the coupled region respectively, the scaling transformation factor $\{\delta_1(\varepsilon), \delta_2(\varepsilon), \ldots, \delta_K(\varepsilon)\}$ for inner solutions.

Output: The MAE-TransNet solution u_{NN}.

1 Form the outer solution problem defined over Ω and the inner solution problems defined over $\{\Omega_1^\zeta, \Omega_2^\zeta, \cdots \Omega_K^\zeta\}$ with the scaling factors $\{1/\delta_1(\varepsilon), 1/\delta_2(\varepsilon), \ldots, 1/\delta_K(\varepsilon)\}$ respectively, then compute the network composite solution u_{NN}^c by feeding $\{\gamma^o, \gamma_1^i, \gamma_2^i, \ldots, \gamma_K^i\}$ and $\{M^o, M_1^i, M_2^i, \ldots, M_K^i\}$ into Algorithm 2.

2 Find the coupled region $\Omega^{\zeta,\eta}$ with the scaling factors $1/\delta_m(\varepsilon)$ and $1/\delta_n(\varepsilon)$ (the coupled region Ω^{ii} is assumed to arise from the interaction between the m-th and n-th boundary layers), then correspondingly form the problem defined over $\Omega^{\zeta,\eta}$.

3 Compute the network solution $\tilde{u}^{ii}$ over the coupled region $\Omega^{\zeta,\eta}$ by feeding γ^{ii} and M^{ii} into Algorithm 1.

4 Return u_{NN} by concatenating u_{NN}^c in $\Omega \setminus \Omega^{ii}$ and u_{NN}^{ii} in Ω^{ii}.

4. Numerical experiments

In this section, we conduct numerical experiments for the proposed MAE-TransNet method on a series of singular perturbation problems in various dimensions, including linear problems with single or multiple boundary layers (test cases 1, 2, and 3), a nonlinear problem (test case 4), the 2D Couette flow problem (test case 5), a 2D coupled boundary layers problem (test case 6), and the 3D Burgers vortex problem (test case 7).

The collocation points (i.e., the training points) and the test points are sampled uniformly in the computational domain (including interior domain and boundaries), and the number of test points is set to 2^d times as large as that of collocation points, where d is the problem dimension. For the singular perturbation problem with a specific ε, the number of collocation points is maintained consistent across the computation of the outer, inner and coupled-region solutions, if needed. As in [39], the discrete L_2 error and the discrete L_∞ error are used to evaluate the network solution u_{NN} obtained by different neural network method

$$\|u - u_{\mathrm{NN}}\|_2 = \sqrt{\sum_{k=1}^{K}(u(\boldsymbol{x}^k) - u_{\mathrm{NN}}(\boldsymbol{x}^k))^2}, \quad \|u - u_{\mathrm{NN}}\|_\infty = \max\left\{|u(\boldsymbol{x}^k) - u_{\mathrm{NN}}(\boldsymbol{x}^k)|\right\}_{k=1}^{K},$$

where $\boldsymbol{x}^k$ is the k-th test point and K is the total number of test points. Our code is implemented using PyTorch on a workstation with an NVIDIA GeForce RTX 4090.

4.1. Test case 1: 1D linear problem with a single boundary layer

We consider the 1D advection-diffusion-reaction problem in the domain $\Omega = (0, 1)$ [27, 35]:

$$\begin{cases} \varepsilon \dfrac{\mathrm{d}^2 u(x)}{\mathrm{d}x^2} + (1+\varepsilon)\dfrac{\mathrm{d}u(x)}{\mathrm{d}x} + u(x) = 0, \quad x \in (0,1), \\ u(0) = 0, \quad u(1) = 1. \end{cases} \tag{26}$$

It is well known that one boundary layer occurs at $x = 0$ with the thickness ε for this problem. In fact the exact solution is given by $u(x) = (e^{-x} - e^{-x/\varepsilon})/(e^{-1} - e^{-1/\varepsilon})$.

As $\varepsilon \to 0$, the outer solution u^o satisfies the following boundary value problem in the domain $\Omega = (0, 1)$:

$$\begin{cases} \dfrac{\mathrm{d}u^o(x)}{\mathrm{d}x} + u^o(x) = 0, \quad x \in (0,1), \\ u^o(1) = 1. \end{cases} \tag{27}$$

To establish the inner solution u^i at $x = 0$, the scaling transformation $\zeta = \frac{x}{\delta(\varepsilon)} := \frac{x}{\varepsilon}$ is introduced to magnify the boundary layer and the second boundary condition is obtained through the matching principle, then as $\varepsilon \to 0$, the problem (26) reduces to the following boundary value problem in the domain $\Omega^\zeta = (0, \frac{1}{\varepsilon})$:

$$\begin{cases} \dfrac{\mathrm{d}^2\bar{u}^i(\zeta)}{\mathrm{d}\zeta^2} + \dfrac{\mathrm{d}\bar{u}^i(\zeta)}{\mathrm{d}\zeta} = 0, \quad \zeta \in (0, \dfrac{1}{\varepsilon}), \\ \bar{u}^i(0) = 0, \quad \bar{u}^i(\dfrac{1}{\varepsilon}) = u^o(0). \end{cases} \tag{28}$$

It is easy to verify $u^o(x) = e^{1-x}$ and $\bar{u}^i(\zeta) = e - e^{1-\zeta}$, Finally, the composite solution to the problem (26) $u^c(x) = e^{1-x} - e^{1-x/\varepsilon}$ is obtained from (8).

The thickness parameter ε is first set to 5×10^{-4}. For our MAE-TransNet method, the shape parameter for the outer solution is set to $\gamma^o = 1$, and for the inner solution, it is set to $\gamma^i = 0.5$. The numbers of hidden-layer neurons for the outer solution and the inner solution are set to $M^o = M^i = 10$. The number of collocation points in Ω and Ω^ζ is uniformly set to $N_\Omega = N_{\Omega^\zeta} = 1999$ and that on $\partial\Omega$ and $\partial\Omega^\zeta$ is uniformly set to $N_{\partial\Omega} = N_{\partial\Omega^\zeta} = 2$, then the total number of collocation points is denoted as $N = N_\Omega + N_{\partial\Omega} = N_{\Omega^\zeta} + N_{\partial\Omega^\zeta} = 2001$. The left column of Figure 10 presents the comparisons of the exact solutions u^o, u^i and u and corresponding MAE-TransNet solutions u^o_{NN}, u^i_{NN} and u^c_{NN}, while the right column shows corresponding pointwise absolute errors. Using MAE-TransNet with only 10 neurons for each of two boundary value problems, the produced numerical results exhibit errors with magnitudes 10^{-7} in u^o_{NN}, 10^{-3} in $\bar{u}^i_{\mathrm{NN}}$, and 10^{-3} in u^c_{NN}. Figure 11 compares the performance of MAE-TransNet and TransNet methods with different number of neurons under identical collocation and test points. The shape parameters for MAE-TransNet are mentioned above, and those for TransNet are determined via the golden section search. It is observed that the MAE-TransNet solution u^c_{NN} achieve the best accuracy with just a total of $M^i + M^o = 20$ neurons, and the errors $\|u^c_{\mathrm{NN}} - u\|_2$ and $\|u^c_{\mathrm{NN}} - u\|_\infty$ show no further improvements along the increase of neurons. This is mainly because that the MAE error $u^c - u$ already dominates the network approximation error $u^c_{\mathrm{NN}} - u^c$ in this case. In addition, we easily find that the solutions exhibits the MAE-TransNet method significantly and consistently outperforms the original TransNet method.

Table 1 reports the performance comparisons of the four neural network methods (PINN, BL-PINN, TransNet and MAE-TransNet) using the same collocation and test points under different values of ε (5×10^{-3}, 5×10^{-4} and 5×10^{-5} respectively). For PINN, the network consists of 5 hidden layers with 120 neurons per layer, and the learning rate is set to 1×10^{-4} with 2000 epochs. For BL-PINN, all parameters follow the code in [27]. For instance, each network corresponds to the inner solution and outer solution has 5 hidden layers with 60 neurons per layer, and the learning rate is set to 1×10^{-4} with 2000 epochs. The shape parameters for TransNet and MAE-TransNet are mentioned above. It is observed that: first, PINN and TransNet fails to solve this singular perturbation problem; second, as BL-PINN does, the solution accuracy of MAE-TransNet gets

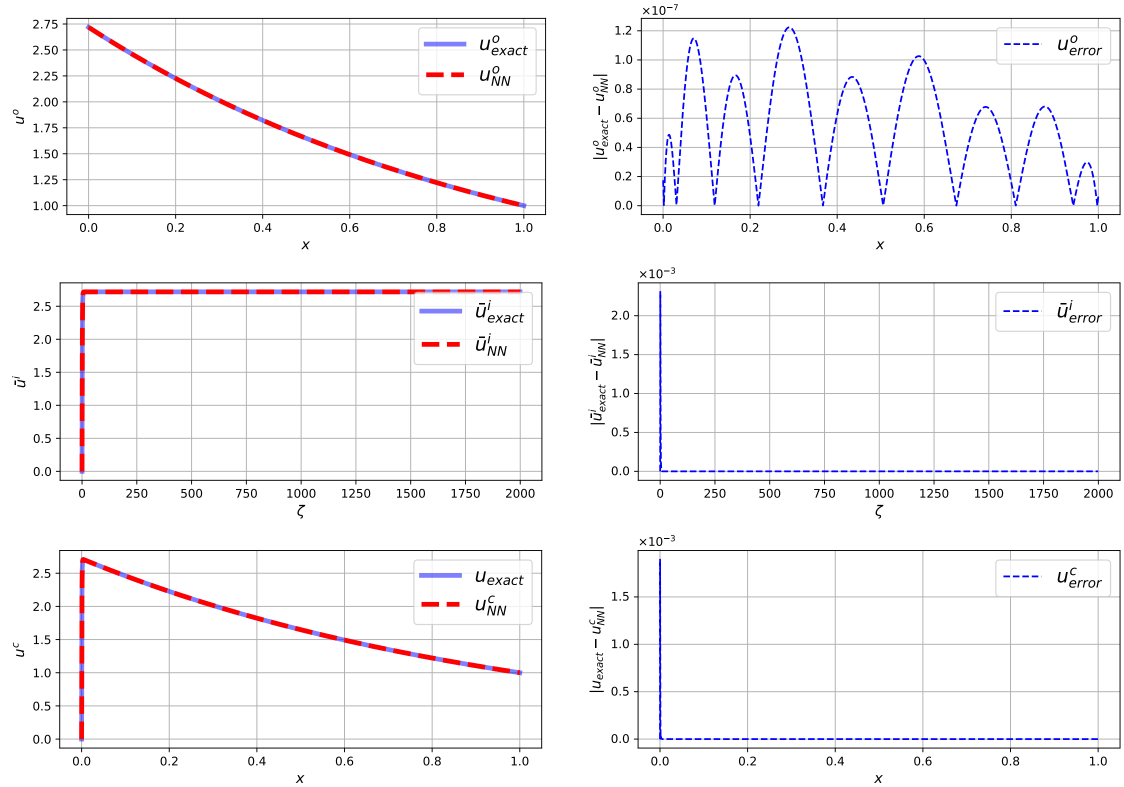

Figure 10: Comparisons between the exact solutions u^o, u^i and u and corresponding MAE-TransNet solutions $u^o_{\rm NN}$, $u^i_{\rm NN}$ and $u^c_{\rm NN}$ (from top to bottom) for the test case 1 with $\varepsilon = 5 \times 10^{-4}$ in Subsection 4.1.

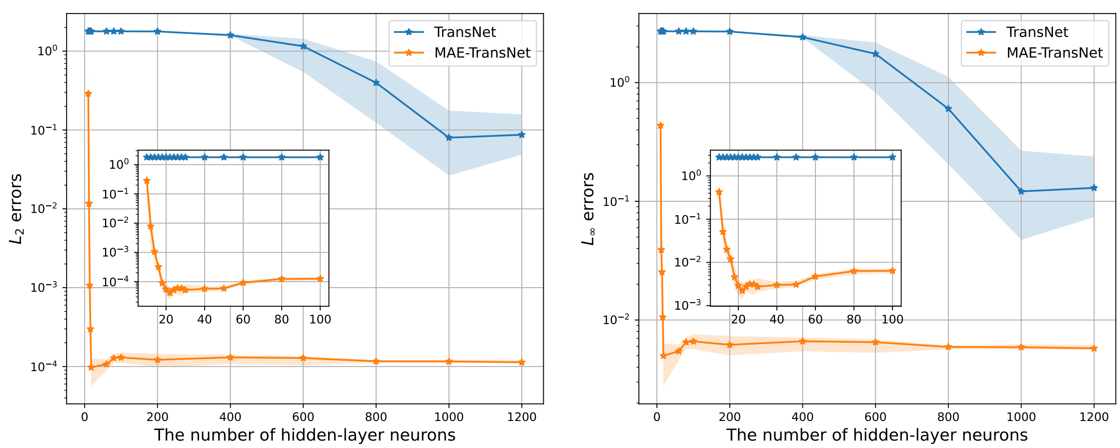

Figure 11: Comparison of the performance between MAE-TransNet solution $u^c_{\rm NN}$ and TransNet solution $u_{\rm NN}$ for the test case 1 with $\varepsilon = 5 \times 10^{-4}$ in Subsection 4.1. Left: L_2 errors $\|u^c_{\rm NN} - u\|_2$ and $\|u_{\rm NN} - u\|_2$. Right: L_∞ errors $\|u^c_{\rm NN} - u\|_\infty$ and $\|u_{\rm NN} - u\|_\infty$. The confidence band represents the range between the 30-th and 70-th percentiles across 50 repeated runs.

improved when ε decreases, which is consistent with the theory of matched asymptotic expansions; third, compared to BL-PINN, our MAE-TransNet requires significantly fewer neurons and less running times while achieves more accurate solutions.

Table 1: Performance comparison of the four neural network methods (PINN, BL-PINN, TransNet and MAE-TransNet) for the test case 1 in Subsection 4.1.

Method	ε	Points	Neurons	Time(s)	L_2 Error	L_∞ Error
PINN	5×10^{-3}	201	600	16.9	1.49e+0	2.38e+0
	5×10^{-4}	2,001	600	18.9	1.53e+0	2.37e+0
	5×10^{-5}	20,001	600	19.7	1.55e+0	2.35e+0
BL-PINN	5×10^{-3}	201	600	34.4	4.80e−3	4.92e−2
	5×10^{-4}	2,001	600	38.5	2.27e−3	6.88e−3
	5×10^{-5}	20,001	600	40.2	1.90e−3	5.88e−3
TransNet	5×10^{-3}	201	20	0.0059	1.79e+0	2.65e+0
	5×10^{-4}	2,001	20	0.0116	1.78e+0	2.70e+0
	5×10^{-5}	20,001	20	0.0671	1.78e+0	2.70e+0
MAE-TransNet	5×10^{-3}	201	20	0.0044	2.77e−5	2.41e−3
	5×10^{-4}	2,001	20	0.0078	2.39e−5	1.94e−3
	5×10^{-5}	20,001	20	0.0241	1.06e−5	1.65e−3

4.2. Test case 2: 1D linear problem with two boundary layers of the same thickness

We consider the linear boundary value problem (14) presented in Subsection 3.1.2. The shape parameters of our MAE-TransNet for the two inner solutions are set to $\gamma_1^i = \gamma_2^i = 0.25$, and the numbers of hidden-layer neurons are set to $M_1^i = M_2^i = 10$. A classic finite difference method (FDM) with a sampling interval length of 5×10^{-5} is used to generate the high-quality reference solution u_{ref}. The left column of Figure 12 presents the comparisons of the reference solution u_{ref} and corresponding MAE-TransNet solution u^c_{NN} with different values of ε, while the right column shows corresponding pointwise absolute errors. Our MAE-TransNet method achieves high accuracy with different values of ε while using the same hidden layer parameters, highlighting its transferability. Table 2 reports the performance comparisons of the two neural network methods (TransNet and MAE-TransNet) using the same collocation and test points under different values of ε (1×10^{-2}, 5×10^{-3} and 1×10^{-3} respectively). It is observed that TransNet again fails to accurately solve the singular perturbation problem with two boundary layers of the same thickness and as ε decreases, the error of MAE-TransNet solution also decreases, demonstrating strong agreement with the predictions of matched asymptotic expansion theory.

Table 2: Performance comparison of the two neural network methods (TransNet and MAE-TransNet) for the test case 2 in Subsection 4.2.

Method	ε	Points	Neurons	L_2 Error	L_∞ Error
TransNet	1×10^{-2}	2,001	20	1.44e−1	1.62e+0
	5×10^{-3}	2,001	20	5.25e−1	1.59e+0
	1×10^{-3}	2,001	20	5.72e−1	1.74e+0
MAE-TransNet	1×10^{-2}	2,001	20	1.27e−2	2.74e−2
	5×10^{-3}	2,001	20	6.36e−3	1.40e−2
	1×10^{-3}	2,001	20	1.19e−3	3.13e−3

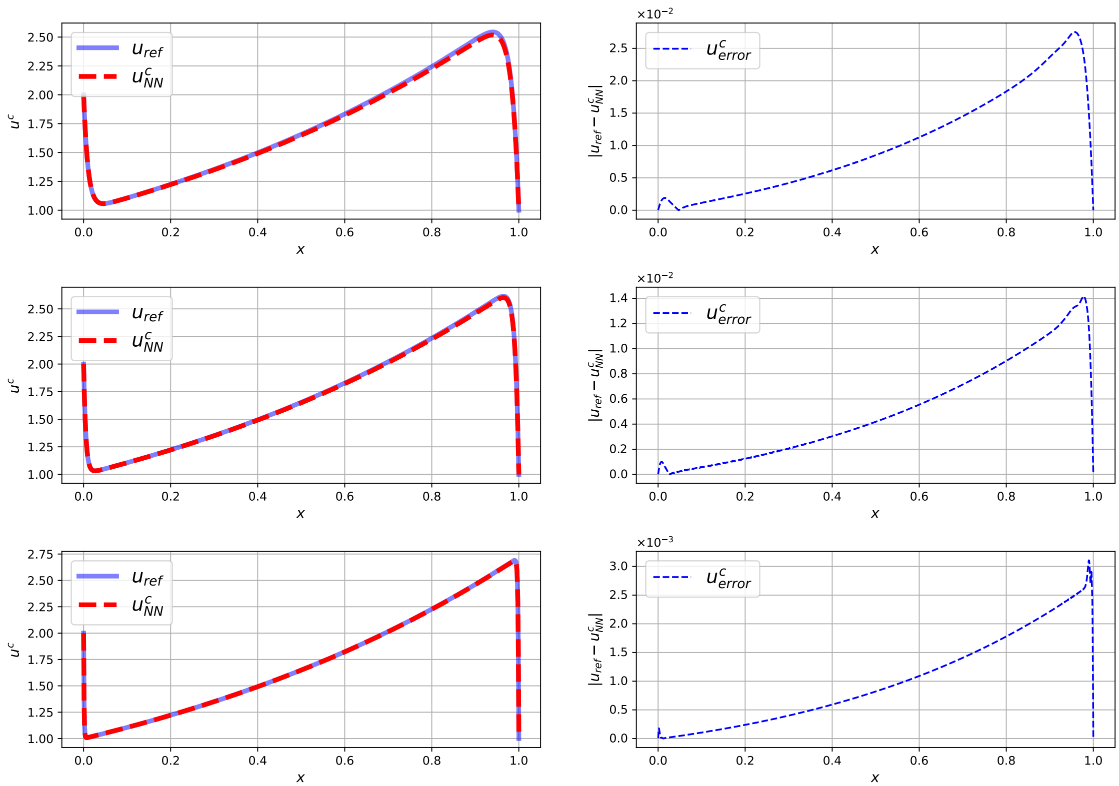

Figure 12: Comparison between the reference solution u_{ref} and corresponding MAE-TransNet solution u^c_{NN} for the test case 2 with $\varepsilon = 1 \times 10^{-2}$, $\varepsilon = 5 \times 10^{-3}$ and $\varepsilon = 1 \times 10^{-3}$ (from top to bottom) in Subsection 4.2.

4.3. Test case 3: 1D linear problem with two boundary layers of different thicknesses

We consider the following linear problem in the domain $\Omega = (0, 1)$ [35]:

$$\begin{cases} \varepsilon \dfrac{\mathrm{d}^2 u(x)}{\mathrm{d}x^2} - x^2 \dfrac{\mathrm{d}u(x)}{\mathrm{d}x} - u(x) = 0, \quad x \in (0, 1), \\ u(0) = 1, \quad u(1) = 1. \end{cases} \tag{29}$$

This problem involves two boundary layers, one is at the point $x = 0$ with the thickness $\sqrt{\varepsilon}$ and the other is at $x = 1$ with the thickness ε.

As $\varepsilon \to 0$, the problem (29) reduces to the following outer solution problem:

$$-x^2 \frac{\mathrm{d}u^o(x)}{\mathrm{d}x} - u^o(x) = 0, \quad x \in (0, 1), \tag{30}$$

which gives the outer solution $u^o(x) = C_0 e^{1/x}$. Moreover, as $x \to 0^+$, $u^o(x)$ becomes infinite unless $C_0 = 0$, i.e., $u^o(x) = 0$. For the inner solution $u^i_1(x)$ at $x = 0$, the scaling transformation $\zeta = \frac{x}{\delta_1(\varepsilon)} := \frac{x}{\sqrt{\varepsilon}}$ and the matching principle are used to obtain the following boundary value problem:

$$\begin{cases} \dfrac{\mathrm{d}^2 \bar{u}^i_1(\zeta)}{\mathrm{d}\zeta^2} - \bar{u}^i_1(\zeta) = 0, \quad \zeta \in (0, \dfrac{1}{\sqrt{\varepsilon}}), \\ \bar{u}^i_1(0) = 1, \quad \bar{u}^i_1(\dfrac{1}{\sqrt{\varepsilon}}) = u^o(0). \end{cases} \tag{31}$$

Similarly, for the inner solution $u_2^i(x)$ at $x=1$, $\zeta=\frac{1-x}{\delta_2(\varepsilon)}:=\frac{1-x}{\varepsilon}$, $\varepsilon\to 0$ and the matching principle are used to obtain the following problem:

$$\begin{cases}\dfrac{\mathrm{d}^2\bar{u}_2^i(\zeta)}{\mathrm{d}\zeta^2}+\dfrac{\mathrm{d}\bar{u}_2^i(\zeta)}{\mathrm{d}\zeta}=0, \quad \zeta\in(0,\dfrac{1}{\varepsilon}),\\ \bar{u}_2^i(0)=1, \quad \bar{u}_2^i(\dfrac{1}{\varepsilon})=u^o(1).\end{cases} \tag{32}$$

Then, the composite solution of problem (29) is obtained from (13) with $K=2$.

The shape parameters for the two inner solutions are set to $\gamma_1^i=\gamma_2^i=0.25$, and the numbers of hidden-layer neurons are set to $M_1^i=M_2^i=10$. A FDM with a sampling interval length of 5×10^{-5} is again implemented to provide the reference solution u_{ref}. The left column of Figure 13 presents the comparisons of the reference solution u_{ref} and corresponding MAE-TransNet solution u_{NN}^c with different values of ε, while the right column shows corresponding pointwise absolute errors. MAE-TransNet effectively captures boundary layers of varying thicknesses under different values of ε, while maintaining the same hidden layer parameters, demonstrating the transferability of MAE-TransNet. Table 3 reports the performance comparisons of the two neural network methods (TransNet and MAE-TransNet) using the same collocation and test points under different values of ε (1×10^{-2}, 5×10^{-3} and 1×10^{-3} respectively).

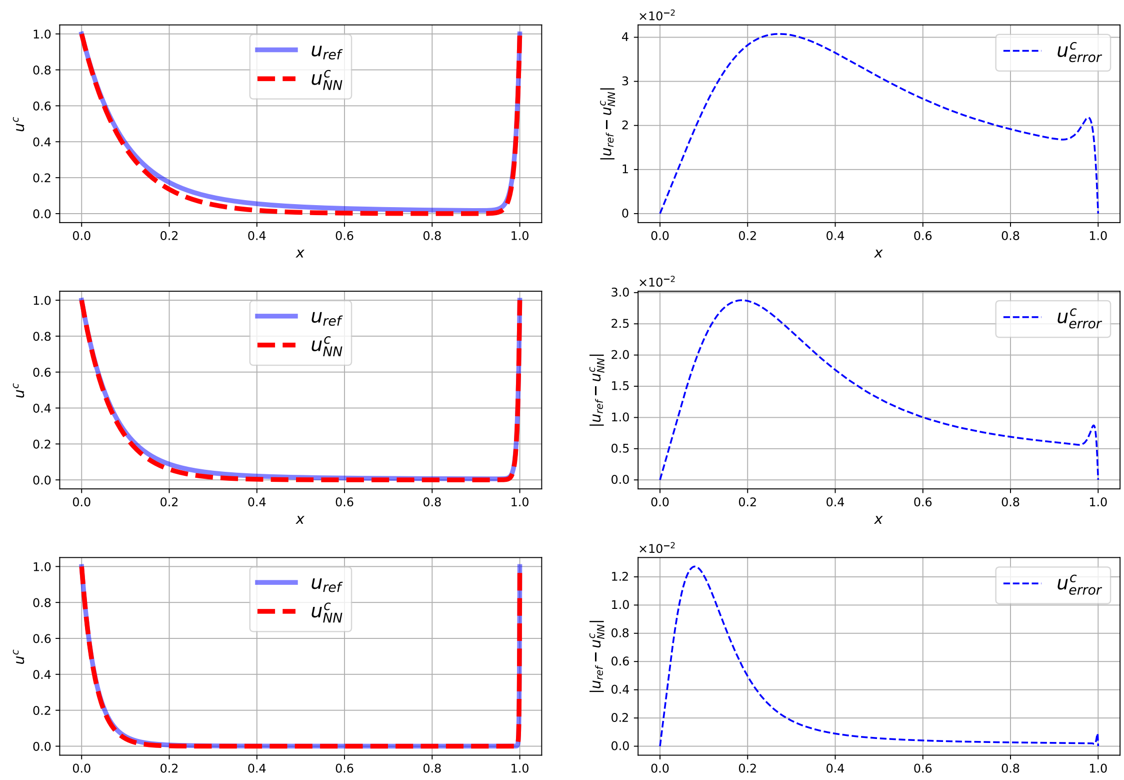

Figure 13: Comparison between the reference solution u_{ref} and corresponding MAE-TransNet solution u_{NN}^c for the test case 3 with $\varepsilon=1\times10^{-2}$, $\varepsilon=5\times10^{-3}$ and $\varepsilon=1\times10^{-3}$ (from top to bottom) in Subsection 4.3.

Table 3: Performance comparison of the two neural network methods (TransNet and MAE-TransNet) for the test case 3 in Subsection 4.3.

Method	ε	Points	Neurons	L_2 Error	L_∞ Error
TransNet	1×10^{-2}	2,001	20	2.42e−1	9.98e−1
	5×10^{-3}	2,001	20	1.74e−1	9.68e−1
	1×10^{-3}	2,001	20	1.24e−1	9.37e−1
MAE-TransNet	1×10^{-2}	2,001	20	2.88e−2	4.15e−2
	5×10^{-3}	2,001	20	1.62e−2	2.87e−2
	1×10^{-3}	2,001	20	4.32e−3	1.27e−2

4.4. Test case 4: 1D nonlinear problem with a single boundary layer

We consider the following nonlinear advection-diffusion-reaction problem in the domain $\Omega = (0,1)$ [27, 35]:

$$\begin{cases} \varepsilon\dfrac{\mathrm{d}^2u(x)}{\mathrm{d}x^2} + 2\dfrac{\mathrm{d}u(x)}{\mathrm{d}x} + e^{u(x)} = 0, \quad x\in(0,1), \\ u(0)=0, \quad u(1)=0. \end{cases} \tag{33}$$

It is well known that one boundary layer occurs at $x=0$ with the thickness ε.

As $\varepsilon\to 0$, the problem (33) reduces to the following outer solution problem:

$$\begin{cases} 2\dfrac{\mathrm{d}u^o(x)}{\mathrm{d}x} + e^{u^o(x)} = 0, \quad x\in(0,1), \\ u^o(1)=0. \end{cases} \tag{34}$$

For the inner solution $u^i(x)$ at $x=0$, the scaling transformation $\zeta = \frac{x}{\delta(\varepsilon)} := \frac{x}{\varepsilon}$ and the matching principle are used to obtain the following boundary value problem:

$$\begin{cases} \dfrac{\mathrm{d}^2\bar{u}^i(\zeta)}{\mathrm{d}\zeta^2} + 2\dfrac{\mathrm{d}\bar{u}^i(\zeta)}{\mathrm{d}\zeta} = 0, \quad \zeta\in(0,\dfrac{1}{\varepsilon}), \\ \bar{u}^i(0)=0, \quad \bar{u}^i(\dfrac{1}{\varepsilon}) = u^o(0). \end{cases} \tag{35}$$

Finally, the composite solution of problem (33) is determined through (8).

The shape parameters for the outer and inner solutions are set to $\gamma^o=\gamma^i=1$, and the numbers of hidden-layer neurons are set to $M^o=M^i=10$. The reference solution u_{ref} is obtained from 20 Picard iterations, in each iteration the FDM with a sampling interval length of 5×10^{-4} is used for solution of the resulting linear system. The left column of Figure 14 presents the comparisons of the reference solution u_{ref} and corresponding MAE-TransNet solution u^c_{NN} with different values of ε, while the right column shows corresponding pointwise absolute errors. For the nonlinear singular perturbation problem with different values of ε, MAE-TransNet can still achieve high accuracy while using the same hidden layer parameters. Table 4 reports the performance comparisons of the two neural network methods (TransNet and MAE-TransNet) using the same collocation and test points under different values of ε (5×10^{-2}, 1×10^{-2} and 5×10^{-3} respectively). We observed that transNet fails to accurately solve the nonlinear singular perturbation problem, and as ε decreases, the error of MAE-TransNet solution decreases as well, remaining consistent with the matched asymptotic expansions theory.

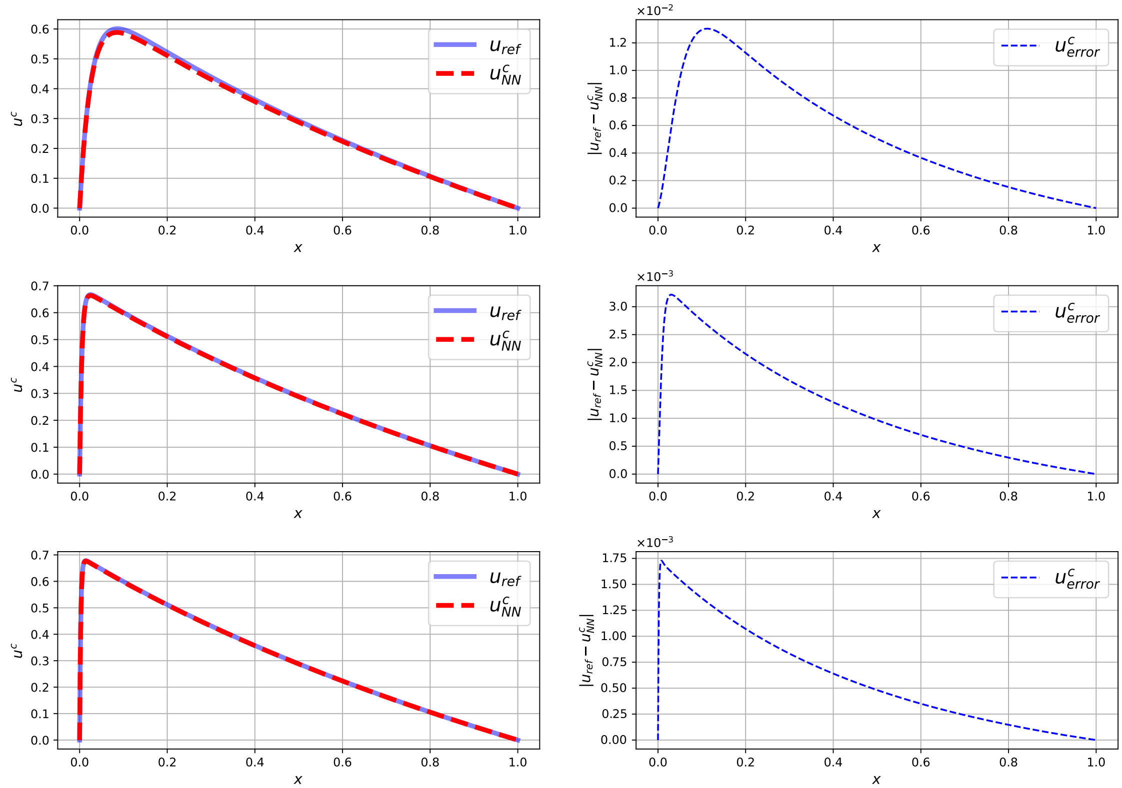

Figure 14: Comparison between the reference solution u_{ref} and corresponding MAE-TransNet solution u^c_{NN} for the test case 4 with $\varepsilon = 5 \times 10^{-2}$, $\varepsilon = 1 \times 10^{-2}$ and $\varepsilon = 5 \times 10^{-3}$ (from top to bottom) in Subsection 4.4.

Table 4: Performance comparison of the two neural network methods (TransNet and MAE-TransNet) for the test case 4 in Subsection 4.4.

Method	ε	Points	Neurons	L_2 Error	L_∞ Error
TransNet	5×10^{-2}	1,001	20	2.59e−1	6.00e−1
	1×10^{-2}	1,001	20	2.75e−1	6.66e−1
	5×10^{-3}	1,001	20	2.77e−1	6.78e−1
MAE-TransNet	5×10^{-2}	1,001	20	6.83e−2	1.30e−2
	1×10^{-2}	1,001	20	1.51e−3	3.29e−3
	5×10^{-3}	1,001	20	7.67e−4	1.73e−3

4.5. Test case 5: 2D Couette flow problem

We consider the 2D advection-diffusion transport in Couette flow problem (18), as detailed in Subsection 3.2.1 and compare the performance of our MAE-TransNet with BL-PINN. Note that the outer solution $u^o(x, y) = 0$ is known for this example, so we don't need to solve it. For BL-PINN, all parameters are adopted from the code in [27]. For MAE-TransNet, the shape parameter is set to $\gamma^i = 0.2$ and the number of hidden-layer neurons is set to $M^i = 128$ to solve for the inner solution. The number of collocation points in Ω and Ω^ζ is uniformly set to $N_\Omega = N_{\Omega\zeta} = 199 \times 199$ and that on $\partial\Omega$ and $\partial\Omega^\zeta$ is uniformly set to $N_{\partial\Omega} = N_{\partial\Omega\zeta} = 4000$, then the total number of collocation points is set to $N = N_\Omega + N_{\partial\Omega} = N_{\Omega\zeta} + N_{\partial\Omega\zeta} = 43601$. The reference solution u_{ref} is obtained by FDM with a sampling interval length of 1.25×10^{-3}. Table 5 reports the performance comparisons of the two neural network methods (BL-PINN and MAE-TransNet) using the same collocation and test points under different values of ε (1×10^{-2}, 1×10^{-3} and 1×10^{-4} respectively). We find that the solution accuracy of MAE-TransNet still gets improved when ε decreases, which is consistent with the theory of matched asymptotic expansions. Compared to BL-PINN, our MAE-TransNet requires much fewer neurons and less running times while achieves even better accuracy.

Table 6 reports the performance of MAE-TransNet with different number of hidden-layer neurons in the case of $\varepsilon = 1 \times 10^{-4}$. It is easy to see that MAE-TransNet gets improved accuracy when the number of neurons increases. The results of BL-PINN with 896 neurons and MAE-TransNet with 128 neurons in the case of $\varepsilon = 1 \times 10^{-4}$ are presented in Figure 15 and Figure 16. We observe that both the MAE-TransNet and BL-PINN solutions closely match the reference solution, and the MAE-TransNet solution converges noticeably closer to the reference solution on the bottom wall within the boundary layer (i.e., $y = 0$).

Table 5: Performance comparison of the two neural network methods (BL-PINN and MAE-TransNet) for the test case 5 in Subsection 4.5.

Method	ε	Points	Neurons	Time(s)	L_2 Error	L_∞ Error
BL-PINN	1×10^{-2}	43,601	896	20,228	1.42e−1	6.86e−1
	1×10^{-3}	43,601	896	20,236	1.10e−2	9.18e−2
	1×10^{-4}	43,601	896	20,200	1.34e−3	3.79e−2
MAE-TransNet	1×10^{-2}	43,601	128	0.717	9.07e−3	2.54e−1
	1×10^{-3}	43,601	128	0.721	3.23e−3	8.40e−2
	1×10^{-4}	43,601	128	0.713	1.18e−3	2.04e−2

Table 6: Performance of MAE-TransNet with different numbers of hidden-layer neurons for the test case 5 with $\varepsilon = 1 \times 10^{-4}$ in Subsection 4.5.

Method	ε	Points	Neurons	Time(s)	L_2 Error	L_∞ Error
MAE-TransNet	1×10^{-4}	43,601	32	0.156	3.89e−3	5.31e−2
		43,601	64	0.318	1.78e−3	3.05e−2
		43,601	128	0.713	1.18e−3	2.04e−2

4.6. Test case 6: 2D coupled boundary layers problem

We consider the 2D singular perturbation convection-diffusion problem (21), as detailed in Subsection 3.2.2. Taking $\varepsilon = 2^{-6}$ as an example, the parameter associated with the coupled region is set to $A = 0.05$ (i.e., the coupled region $\Omega^{ii} = [0, 0.05]\times[0, 0.05]$). For our MAE-TransNet method, the shape parameters for the outer, the two inner and the coupled-region solutions are set to $\gamma^o = 1$, $\gamma_1^i = \gamma_2^i = 0.2$ and $\gamma^{ii} = 1$. The numbers of hidden-layer neurons for the outer, the two inner and the coupled-region solutions are set to $M^o = M_1^i = M_2^i = M^{ii} = 200$. The total number of collocation points in Ω, Ω_1^ζ, Ω_2^η and $\Omega^{\zeta,\eta}$ is uniformly set to $N = 399\times399+16000 = 175201$. Figure 17 compares the MAE-TransNet solution and the exact solution with $\varepsilon = 2^{-6}$. It is observed that the two solutions exhibit almost no differences either across the entire domain Ω or in the coupled region Ω^{ii}.

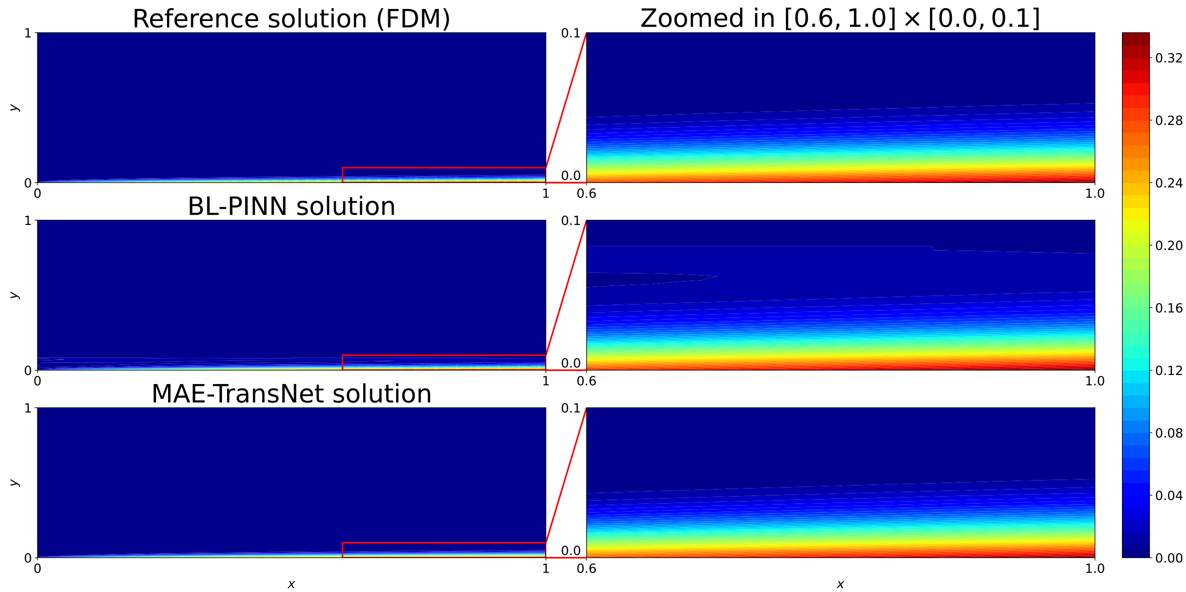

Figure 15: Comparison of the performance between MAE-TransNet and BL-PINN for the test case 5 with $\varepsilon = 1 \times 10^{-4}$ in Subsection 4.5. Left: Over the whole domain Ω. Right: Zoomed in $[0.6, 1] \times [0, 0.1]$.

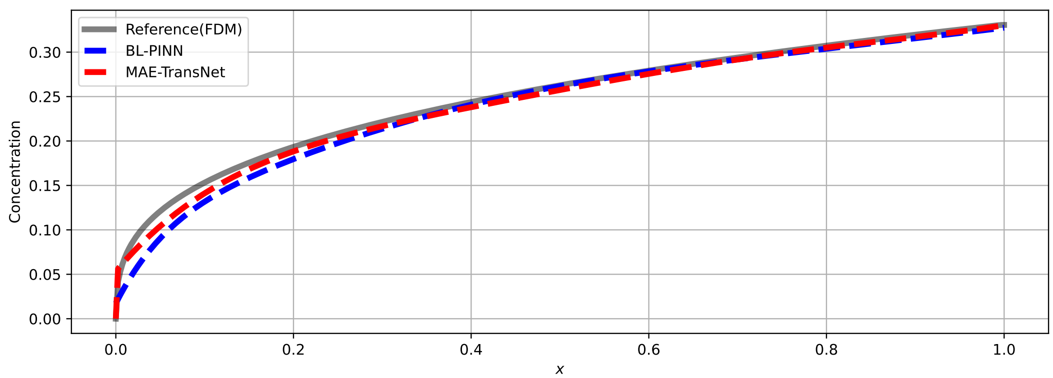

Figure 16: Comparison of the performance at the bottom between MAE-TransNet and BL-PINN for the test case 5 with $\varepsilon = 1 \times 10^{-4}$ in Subsection 4.5.

The left plot of Figure 18 presents the comparison of the exact solution and corresponding MAE-TransNet solution in the coupled region Ω^{ii}, while the right plot shows corresponding pointwise absolute errors. The produced numerical results exhibit errors with magnitudes $O(10^{-3})$ for the MAE-TransNet solution, demonstrating its capability to effectively resolve mutual interactions between boundary layers.

4.7. Test case 7: 3D Burgers vortex problem

The Burgers vortex describes a stationary, self-similar flow based on the balance among radial flow, axial stretching and viscous diffusion. We consider axisymmetric transport in the 3D Burgers vortex problem as discussed in [27, 46]. In cylindrical coordinates (r, θ, x), the velocity field can be written as:

$$v_r = -\frac{a}{2}r, \quad v_\theta = \frac{\Gamma_0}{2\pi r}\left(1 - e^{-br^2}\right), \quad v_x = ax,$$

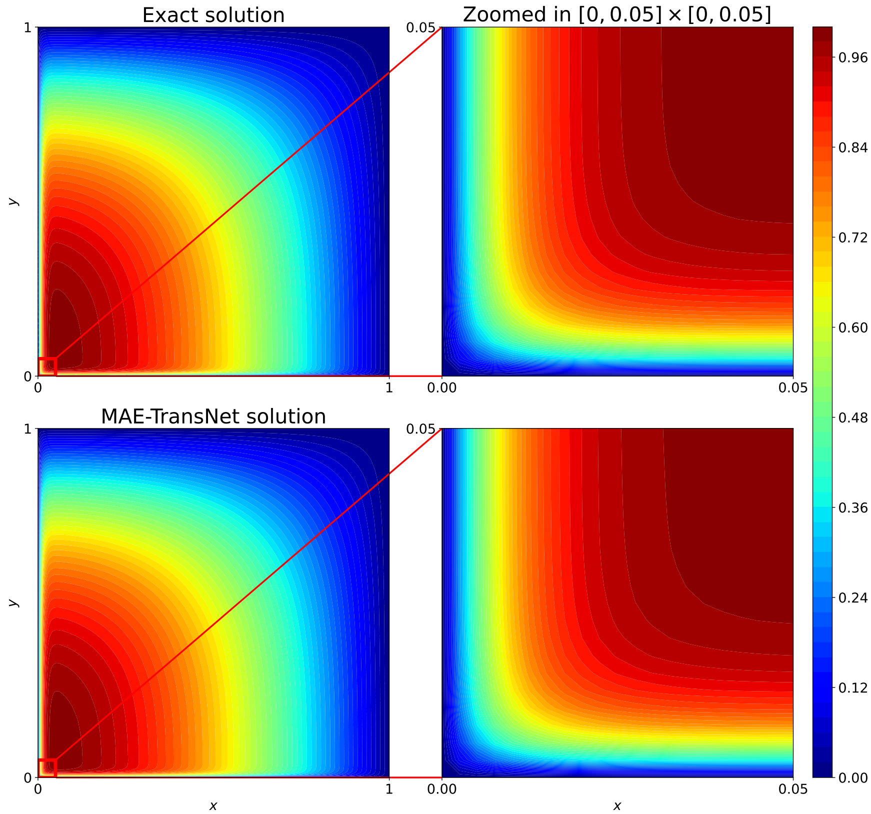

Figure 17: Comparison of the exact solution and the MAE-TransNet solution for the test case 6 with $\varepsilon = 2^{-6}$ in Subsection 4.6. Left: over the whole domain Ω. Right: zoomed in the coupled region $\Omega^{ii} = [0, 0.05] \times [0, 0.05]$.

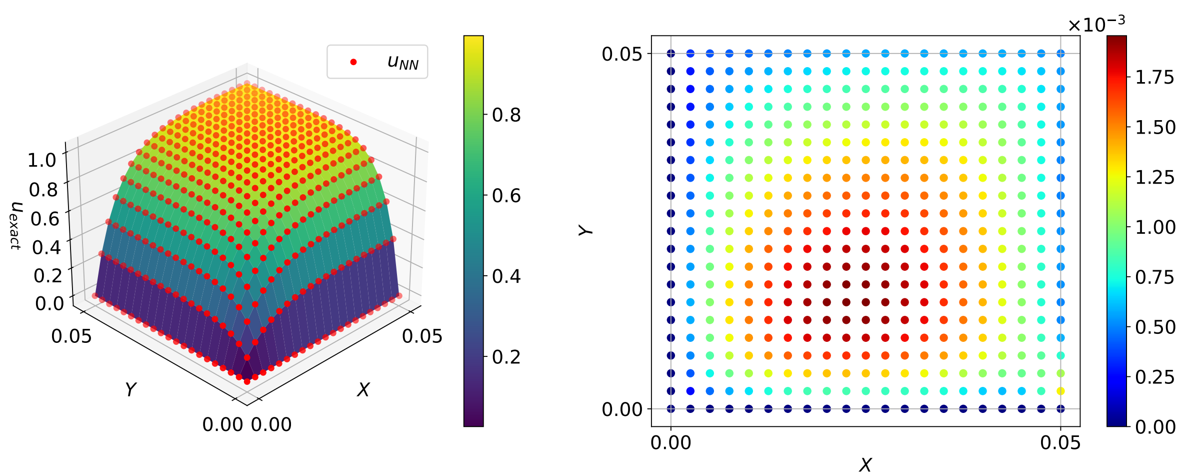

Figure 18: Comparison of the exact solution and the MAE-TransNet solution in the coupled region $\Omega^{ii} = [0, 0.05] \times [0, 0.05]$ for the test case 6 with $\varepsilon = 2^{-6}$ in Subsection 4.6. Left: the surface represents the exact solution, while the red points denote the results from MAE-TransNet. Right: pointwise absolute error.

where $a = 0.2$, $\Gamma_0 = 2\pi$ and $b = 1$. The diffusion coefficient is set to $\varepsilon = 1 \times 10^{-4}$ and the cylindrical domain has a radius of $r = 0.5$ and a height of $x = 0.3$. Due to its axisymmetric nature, the original 3D problem can be simplified into a 2D problem in the domain $\Omega = (0.002, 0.5) \times (0, 0.3)$

via cylindrical coordinate parametrization:

$$
\begin{cases}
\varepsilon\dfrac{\partial^2 c(r,x)}{\partial r^2}+\dfrac{\varepsilon}{r}\dfrac{\partial c(r,x)}{\partial r}+\varepsilon\dfrac{\partial^2 c(r,x)}{\partial x^2} \\
\qquad\qquad = v_r\dfrac{\partial c(r,x)}{\partial r}+v_x\dfrac{\partial c(r,x)}{\partial x}, \quad (r,x)\in(0.002,0.5)\times(0,0.3), \\
\dfrac{\partial c}{\partial r}(r=0.002,x)=0, \quad c(r=0.5,x)=0, \\
\dfrac{\partial c}{\partial x}(r,x=0)=-5, \quad \dfrac{\partial c}{\partial x}(r,x=0.3)=0.
\end{cases}
\tag{36}
$$

Notably, when the boundary layer is sufficiently thick such that $\Omega^{ii}=\Omega$, the solution can be exclusively computed within the coupled region. The scaling transformation $\zeta=\frac{r}{\delta_1(\varepsilon)}:=\frac{r}{\sqrt{\varepsilon}}$, $\eta=\frac{x}{\delta_2(\varepsilon)}:=\frac{x}{\sqrt{\varepsilon}}$ are used to obtain the following boundary value problem:

$$
\begin{cases}
\dfrac{\partial^2\tilde{c}^{ii}(\zeta,\eta)}{\partial\zeta^2}+\dfrac{1}{\zeta}\dfrac{\partial\tilde{c}^{ii}(\zeta,\eta)}{\partial\zeta}+\dfrac{\partial^2\tilde{c}^{ii}(\zeta,\eta)}{\partial\eta^2} \\
\qquad = -0.1\zeta\dfrac{\partial\tilde{c}^{ii}(\zeta,\eta)}{\partial\zeta}+0.2\eta\dfrac{\partial\tilde{c}^{ii}(\zeta,\eta)}{\partial\eta}, \quad (\zeta,\eta)\in(\dfrac{0.002}{\sqrt{\varepsilon}},\dfrac{0.5}{\sqrt{\varepsilon}})\times(0,\dfrac{0.3}{\sqrt{\varepsilon}}), \\
\dfrac{\partial\tilde{c}^{ii}}{\partial\zeta}(\zeta=\dfrac{0.002}{\sqrt{\varepsilon}},\eta)=0, \quad \tilde{c}^{ii}(\zeta=\dfrac{0.5}{\sqrt{\varepsilon}},\eta)=0, \\
\dfrac{\partial\tilde{c}^{ii}}{\partial\eta}(\zeta,\eta=0)=-5\sqrt{\varepsilon}, \quad \dfrac{\partial\tilde{c}^{ii}}{\partial\eta}(\zeta,\eta=\dfrac{0.3}{\sqrt{\varepsilon}})=0,
\end{cases}
\tag{37}
$$

where the boundary conditions are supplied by the problem (36). Subsequently, a TransNet with uniformly distributed hidden-layer neurons is employed to solve the problem (37). The shape parameter is set to $\gamma^{ii}=0.1$ and the number of hidden-layer neurons is set to $M^{ii}=500$. The total number of collocation points in $\Omega^{\zeta,\eta}$ is set to $N=497\times299+1600=150203$. The reference solution u_{ref} is obtained from the FDM with a sampling interval length of 1×10^{-3}. Figure 19 presents the half section views and the cross-sections in cylindrical coordinates of MAE-TransNet solution and the reference solution. It is observed that the two solutions exhibit visually no differences, demonstrating that MAE-TransNet effectively captures the global features of the Burgers vortex. Figure 20 compares the MAE-TransNet solution and the reference solution at the bottom (i.e., $z=0$), and it shows that MAE-TransNet also effectively captures the intricate details in the boundary layers.

5. Conclusions

In this work, we present the matched asymptotic expansions-based transferable neural networks for solving singular perturbation problems characterized by solutions exhibiting sharp gradients within narrow boundary layers. Using matched asymptotic expansions, the original problem is decomposed into inner and outer boundary value problems. The outer solution, valid outside the boundary layer, is approximated by the TransNet with uniform hidden-layer neurons, while the inner solution, valid within the boundary layer, is approximated by the improved TransNet with nonuniform hidden-layer neurons. A matching term is then applied to combine these solutions into a uniformly valid solution across the computational domain. To address the challenging multi-dimensional coupled boundary layer system in the absence of theoretical guidance,

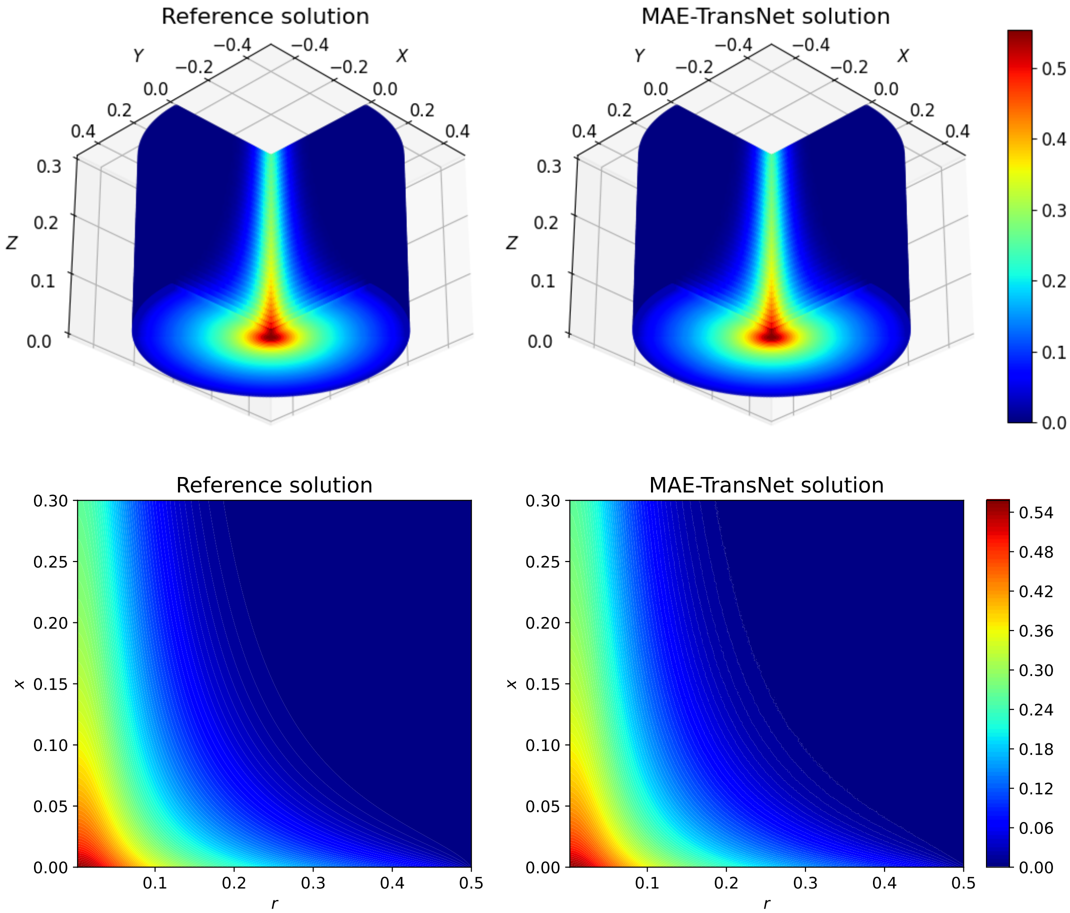

Figure 19: Comparison of the reference solution and the MAE-TransNet solution for the test case 7 with $\varepsilon = 1 \times 10^{-4}$ in Subsection 4.7. Top: the half section views. Bottom: the cross-section in cylindrical coordinates.

MAE-TransNet computes solutions over the entire domain, followed by an auxiliary TransNet that effectively corrects the solution in the coupled region. Numerical experiments confirm that our MAE-TransNet method inherits both the accuracy of matched asymptotic expansions and the efficiency of TransNets, thereby significantly improving computational precision and efficiency compared to other neural network methods, such as BL-PINN in [27]. Additionally, the experiments demonstrate that the same hidden-layer parameters can be employed to boundary layer problems with different value of ε, further highlighting its transferability and substantially reducing the cost of parameter tuning. The MAE-TransNet method provides a promising tool for solving singular perturbation problems with boundary layers. There are several potential related topics that will be studied in the future work. Although various numerical results illustrate that the proposed method converges well and has a good accuracy, exploring rigorously theoretical convergence analysis and error estimates is still an interesting topic. It is also worthy of further investigation into more challenging singular perturbation problems involving moving thin layers [25, 47] or turning points [48], as well as coordinate perturbation problems discussed in [3].

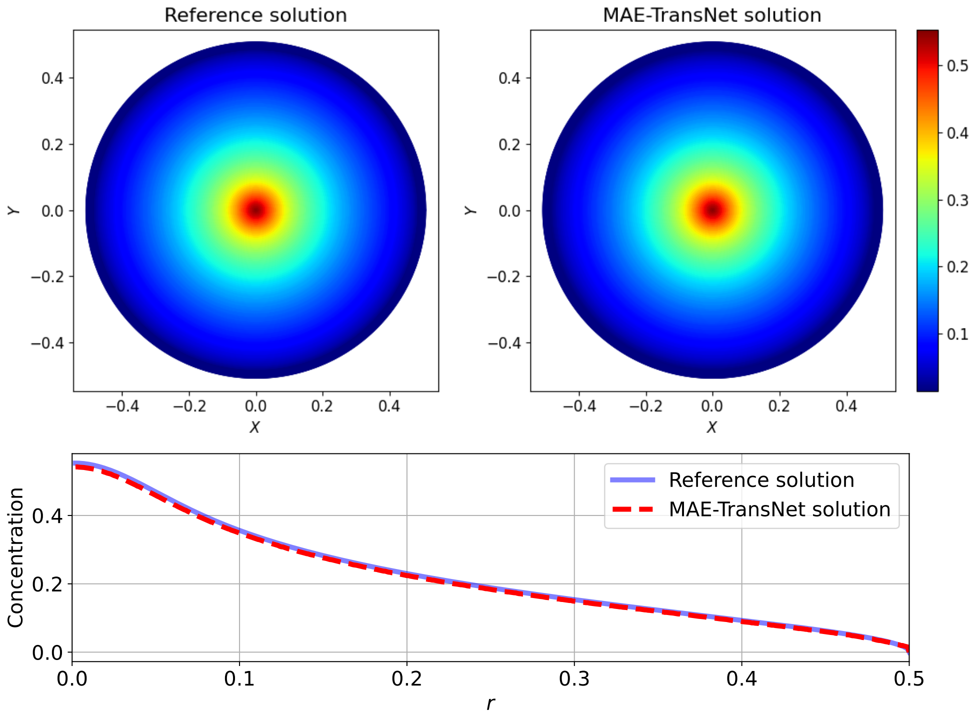

Figure 20: Comparison of the reference solution and the MAE-TransNet solution at the bottom (i.e., $z = 0$) for the test case 7 with $\varepsilon = 1 \times 10^{-4}$ in Subsection 4.7. Top: the section views. Bottom: in cylindrical coordinates.

6. Acknowledgement

L. Zhu's work was partially supported by National Natural Science Foundation of China under grant number 11871091.

References

[1] T. Coupez, E. Hachem, Solution of high-reynolds incompressible flow with stabilized finite element and adaptive anisotropic meshing, Computer Methods in Applied Mechanics and Engineering 267 (2013) 65–85.

[2] H. Riou, P. Ladevèze, A new numerical strategy for the resolution of high-péclet advection–diffusion problems, Computer Methods in Applied Mechanics and Engineering 241-244 (2012) 302–310.

[3] M. J. Ward, M.-C. Kropinski, Asymptotic methods for pde problems in fluid mechanics and related systems with strong localized perturbations in two-dimensional domains, in: Asymptotic methods in fluid mechanics: survey and recent advances, Springer, 2010, pp. 23–70.

[4] X. Li, N. Polydorides, Fast heat transfer simulation for laser powder bed fusion, Computer Methods in Applied Mechanics and Engineering 412 (2023) 116107.

[5] H.-G. Roos, Robust numerical methods for singularly perturbed differential equations, Springer, 2008.

[6] C. Clavero, J. Jorge, F. Lisbona, A uniformly convergent scheme on a nonuniform mesh for convection–diffusion parabolic problems, Journal of Computational and Applied Mathematics 154 (2) (2003) 415–429.

[7] Y. Ge, F. Cao, Multigrid method based on the transformation-free hoc scheme on nonuniform grids for 2d convection diffusion problems, Journal of Computational Physics 230 (10) (2011) 4051–4070.

[8] J. Du, E. Chung, An adaptive staggered discontinuous galerkin method for the steady state convection–diffusion equation, Journal of Scientific Computing 77 (3) (2018) 1490–1518.

[9] V. Kumar, B. Srinivasan, An adaptive mesh strategy for singularly perturbed convection diffusion problems, Applied Mathematical Modelling 39 (7) (2015) 2081–2091.

[10] K. Kumar, P. C. Podila, A new stable finite difference scheme and its error analysis for two-dimensional singularly perturbed convection–diffusion equations, Numerical Methods for Partial Differential Equations 38 (5) (2022) 1215–1231.

[11] P.-W. Hsieh, S.-Y. Yang, C.-S. You, A robust finite difference scheme for strongly coupled systems of singularly perturbed convection-diffusion equations, Numerical Methods for Partial Differential Equations 34 (1) (2018) 121–144.

[12] M. K. Singh, S. Natesan, A parameter-uniform hybrid finite difference scheme for singularly perturbed system of parabolic convection-diffusion problems, International Journal of Computer Mathematics 97 (4) (2020) 875–905.

[13] S. Toprakseven, N. Srinivasan, An efficient weak galerkin fem for third-order singularly perturbed convection-diffusion differential equations on layer-adapted meshes, Applied Numerical Mathematics 204 (2024) 130–146.

[14] J. Zhang, Y. Lv, High-order finite element method on a bakhvalov-type mesh for a singularly perturbed convection–diffusion problem with two parameters, Applied Mathematics and Computation 397 (2021) 125953.

[15] R. Lin, X. Ye, S. Zhang, P. Zhu, A weak galerkin finite element method for singularly perturbed convection-diffusion–reaction problems, SIAM Journal on Numerical Analysis 56 (3) (2018) 1482–1497.

[16] Z. Gharibi, M. Dehghan, Convergence analysis of weak galerkin flux-based mixed finite element method for solving singularly perturbed convection-diffusion-reaction problem, Applied Numerical Mathematics 163 (2021) 303–316.

[17] J. Zhang, M. Stynes, Supercloseness of continuous interior penalty method for convection–diffusion problems with characteristic layers, Computer Methods in Applied Mechanics and Engineering 319 (2017) 549–566.

[18] F. Cao, F. Gao, X. Guo, D. Yuan, Physics-informed neural networks with parameter asymptotic strategy for learning singularly perturbed convection-dominated problem, Computers & Mathematics with Applications 150 (2023) 229–242.

[19] B. Yu, et al., The deep ritz method: a deep learning-based numerical algorithm for solving variational problems, Communications in Mathematics and Statistics 6 (1) (2018) 1–12.

[20] J. Sirignano, K. Spiliopoulos, Dgm: A deep learning algorithm for solving partial differential equations, Journal of computational physics 375 (2018) 1339–1364.

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

[22] L. Lu, X. Meng, Z. Mao, G. E. Karniadakis, Deepxde: A deep learning library for solving differential equations, SIAM Review 63 (1) (2021) 208–228.

[23] F. Cao, F. Gao, D. Yuan, J. Liu, Multistep asymptotic pre-training strategy based on pinns for solving steep boundary singular perturbation problems, Computer Methods in Applied Mechanics and Engineering 431 (2024) 117222.

[24] J. Huang, R. Qiu, J. Wang, Y. Wang, Multi-scale physics-informed neural networks for solving high reynolds number boundary layer flows based on matched asymptotic expansions, Theoretical and Applied Mechanics Letters 14 (2) (2024) 100496.

[25] S. Wang, P. Zhao, Q. Ma, T. Song, General-kindred physics-informed neural network to the solutions of singularly perturbed differential equations, Physics of Fluids 36 (11) (2024).

[26] Z. Aldirany, R. Cottereau, M. Laforest, S. Prudhomme, Multi-level neural networks for accurate solutions of boundary-value problems, Computer Methods in Applied Mechanics and Engineering 419 (2024) 116666.

[27] A. Arzani, K. W. Cassel, R. M. D'Souza, Theory-guided physics-informed neural networks for boundary layer problems with singular perturbation, Journal of Computational Physics 473 (2023) 111768.

[28] S. Dong, Z. Li, Local extreme learning machines and domain decomposition for solving linear and nonlinear partial differential equations, Computer Methods in Applied Mechanics and Engineering 387 (2021) 114129.

[29] J. Chen, X. Chi, Z. Yang, et al., Bridging traditional and machine learning-based algorithms for solving pdes: the random feature method, J Mach Learn 1 (2022) 268–98.

[30] S. Dong, J. Yang, On computing the hyperparameter of extreme learning machines: Algorithm and application to computational pdes, and comparison with classical and high-order finite elements, Journal of Computational Physics 463 (2022) 111290.

[31] J. Chen, Y. Luo, et al., The random feature method for time-dependent problems, arXiv preprint arXiv:2304.06913 (2023).

[32] Z. Cai, J. Chen, M. Liu, X. Liu, Deep least-squares methods: An unsupervised learning-based numerical method for solving elliptic pdes, Journal of Computational Physics 420 (2020) 109707.

[33] F. Calabrò, G. Fabiani, C. Siettos, Extreme learning machine collocation for the numerical solution of elliptic pdes with sharp gradients, Computer Methods in Applied Mechanics and Engineering 387 (2021) 114188.

[34] A. H. Nayfeh, Perturbation methods, John Wiley & Sons, 2024.

[35] C. M. Bender, S. A. Orszag, Advanced mathematical methods for scientists and engineers I: Asymptotic methods and perturbation theory, Springer Science & Business Media, 2013.

[36] W. A. Clark, M. W. Gomes, A. Rodriguez-Gonzalez, L. C. Stein, S. H. Strogatz, Surprises in a classic boundary-layer problem, SIAM Review 65 (1) (2023) 291–315.

[37] F. Hoppensteadt, Analysis of some problems having matched asymptotic expansion solutions, SIAM Review 17 (1) (1975) 123–135.

[38] L. M. Faria, R. R. Rosales, Equation level matching: An extension of the method of matched asymptotic expansion for problems of wave propagation, Studies in Applied Mathematics 139 (2) (2017) 265–287.

[39] Z. Zhang, F. Bao, L. Ju, G. Zhang, Transferable neural networks for partial differential equations, Journal of Scientific Computing 99 (1) (2024) 2.

[40] T. Tang, B. Wang, J. Yang, Asymptotic analysis on the sharp interface limit of the time-fractional cahn–hilliard equation, SIAM Journal on Applied Mathematics 82 (3) (2022) 773–792.

[41] P. C. Bressloff, Asymptotic analysis of target fluxes in the three-dimensional narrow capture problem, Multiscale Modeling & Simulation 19 (2) (2021) 612–632.

[42] T. Lu, L. Ju, L. Zhu, A multiple transferable neural network method with domain decomposition for elliptic interface problems, Journal of Computational Physics (2025) 113902.

[43] K. Pieper, Z. Zhang, G. Zhang, Nonuniform random feature models using derivative information, arXiv preprint arXiv:2410.02132 (2024).

[44] M. H. Holmes, Introduction to the foundations of applied mathematics, Vol. 56, Springer, 2009.

[45] P. C. Podila, V. Sundrani, H. Ramos, J. Vigo-Aguiar, Wavelet-based approximation for two-dimensional singularly perturbed elliptic problems, Journal of Computational and Applied Mathematics (2024) 116069.

[46] R. L. Panton, Incompressible flow, John Wiley & Sons, 2024.

[47] S. Wu, B. Peng, Z. Tian, Exponential compact adi method for a coupled system of convection-diffusion equations arising from the 2d unsteady magnetohydrodynamic (mhd) flows, Applied Numerical Mathematics 146 (2019) 89–122.

[48] K. K. Sharma, P. Rai, K. C. Patidar, A review on singularly perturbed differential equations with turning points and interior layers, Applied Mathematics and Computation 219 (22) (2013) 10575–10609.