SEMI-ANALYTIC PINN METHODS FOR SINGULARLY PERTURBED BOUNDARY VALUE PROBLEMS

GUNG-MIN GIE, YOUNGJOON HONG, AND CHANG-YEOL JUNG

Abstract. We propose a new semi-analytic physics informed neural network (PINN) to solve singularly perturbed boundary value problems. The PINN is a scientific machine learning framework that offers a promising perspective for finding numerical solutions to partial differential equations. The PINNs have shown impressive performance in solving various differential equations including time-dependent and multi-dimensional equations involved in a complex geometry of the domain. However, when considering stiff differential equations, neural networks in general fail to capture the sharp transition of solutions, due to the spectral bias. To resolve this issue, here we develop the semi-analytic PINN methods, enriched by using the so-called corrector functions obtained from the boundary layer analysis. Our new enriched PINNs accurately predict numerical solutions to the singular perturbation problems. Numerical experiments include various types of singularly perturbed linear and nonlinear differential equations.

Contents

1. Introduction 1
1.1. Elliptic and hyperbolic equations 5
2. Enriched Neural Network: linear singularly perturbed boundary value problems 5
2.1. Convection-diffusion equations 5
2.2. Reaction-diffusion equations 9
3. Enriched Neural Network: nonlinear singularly perturbed boundary value problems 11
3.1. Convection-diffusion equations with a non-linear reaction 11
3.2. Stationary Burgers’ equation 13
4. Conclusion 17
Acknowledgments 17
References 17

1. Introduction

Neural networks have been widely studied and used for approximating solutions to differential equations; see, e.g., [1, 4, 5, 7, 20, 21, 28, 29, 30, 35]. In this research direction, many (unsupervised) neural networks without training datasets have been successfully developed, e.g., physics informed neural networks (PINNs) [2, 19, 21, 13], deep Ritz method.

Date: August 22, 2022.
(DRM) [39], and deep Galerkin method (DGM) [33] where the loss function is defined by using a certain residual from the differential equation under consideration. Especially the PINNs use collocation points in the space-time domain as the training data set, and hence the PINNs are suitable for solving time dependent, multi-dimensional equations involved in a complex geometry of the domain, [19, 24, 22, 31, 38, 25, 18, 36, 37].

Compared to many other types of neural networks, some advantages of using the PINNs in the study of differential equations include, but not limited to, first the fact that they are unsupervised learning processes and hence the exact solution of a model differential equation is not a-priori required for the learning process. The exact solution is usually used only when we measure the error between the exact solution and the approximate solution obtained by the PINNs, and hence the PINNs work as like the traditional numerical methods for differential equations, e.g., finite difference, finite elements, and so on. Another big advantage of using PINNs (over the traditional numerical methods) is their flexibility applied to many different types of differential equations mainly because the model equation under consideration is used only when the loss is computed. However, being unsupervised learning process, a repeat of learning is required for different data, i.e., external force, or initial data, and each learning takes a significant amount of time, compared to the relatively short time cost of most traditional numerical methods.

In this article, we approximate the solutions of various 1D boundary value problems, especially when the highest order derivative appearing in each equation is multiplied by a small parameter $\varepsilon > 0$. More precisely, we consider the 1D elliptic differential equations in the form,

$$F(x; u, u_x, \varepsilon u_{xx}) = f(x), \quad 0 < x < 1,$$

supplemented with a Dirichlet boundary condition,

$$u = 0, \quad x = 0, 1.$$

A singularly perturbed boundary value problem, such as our problem (1.1) - (1.2), is well-known to generate a thin layer near the boundary (called the boundary layer), in which a sharp transition of the solution occurs. A large literature has been developed on the mathematical theory of singular perturbations and boundary layers; see, e.g., [11, 14, 26, 32]. Concerning numerical approximation of the singular perturbation problem, a very large computational error is created near the boundary, due to the stiffness of solution inside the boundary layer. Thus, to achieve a sufficiently accurate approximation of the solution near the boundary, a massive mesh refinement is usually required, near the boundary, for the most classical numerical schemes. Instead of introducing massive mesh refinements, new semi-analytic methods have been proposed, see, e.g., [6, 8, 9, 12, 15, 16]. The main component of this semi-analytic method is enriching the basis of traditional numerical methods, e.g. finite elements, finite volumes and so on, by adding a global basis function, called the corrector, which describes the singular behavior of the solution inside boundary layers. Such semi-analytic methods have proven to be highly efficient without any help of mesh refinement near the boundary.
Our main goal is to construct a semi-analytic physics-informed neural networks (PINNs), enriched by using the so-called corrector functions; see Section 2 and 3 below. Toward this end, we first briefly recall the PINNs (well-developed in earlier works; see, e.g., [24]) for our model equation in (1.1) - (1.2):

For an $L$-layer Neural Network (NN) (or $L - 1$-hidden layer NN), the $\ell$-th layer with $N_\ell$ neurons is denoted by $\mathcal{N}^\ell$, $0 \leq \ell \leq L$. Writing the weight matrix and the bias vector at each $\ell$-th layer as $W^\ell \in \mathbb{R}^{N_{\ell} \times N_{\ell-1}}$, $1 \leq \ell \leq L$, $b^\ell \in \mathbb{R}^{N_{\ell}}$, $1 \leq \ell \leq L$, and $b^L \in \mathbb{R}$ (with $N_0 = N_L = 1$ for our 1D problem (1.1) - (1.2)), we use the feed-forward neural network (FNN) with an activation function $\sigma$, and we recursively define

$$
\begin{align*}
\text{input layer} : & \quad \mathcal{N}^0(x) = x \in \mathbb{R}, \\
\text{hidden layers} : & \quad \mathcal{N}^\ell(x) = \sigma(W^\ell \mathcal{N}^{\ell-1}(x) + b^\ell) \in \mathbb{R}^{N_\ell}, \quad 1 \leq \ell \leq L - 1, \\
\text{output layer} : & \quad \mathcal{N}^L(x) = W^L \mathcal{N}^{L-1}(x) + b^L \in \mathbb{R}.
\end{align*}
$$

Using the NN above, we construct an approximation $\hat{u} = \hat{u}(x; \theta)$ where the parameters $\theta = \{W^\ell, b^\ell\}_{1 \leq \ell \leq L}$ is the set of the weight matrices and bias vectors in the neural network. In the next step, the NN approximation $\hat{u}$ is restricted to satisfy the constraint imposed by the PDE and boundary conditions (1.1) - (1.2). For this purpose, we prepare the training set $\mathcal{T} \subset [0, 1]$ which consists of two sets of scattered points, $\mathcal{T}_I \subset (0, 1)$ and $\mathcal{T}_B \subset \{0, 1\}$; and define the corresponding loss function, using the weighted $L^2$ norm, by

$$
\mathcal{L}(\theta, \mathcal{T}) = w_I \mathcal{L}_I(\theta; \mathcal{T}_I) + w_B \mathcal{L}_B(\theta; \mathcal{T}_B),
$$

where

$$
\mathcal{L}_I(\theta, \mathcal{T}_I) = \frac{1}{|\mathcal{T}_I|} \sum_{x \in \mathcal{T}_I} \| F(x; \hat{u}, \hat{u}_x, \varepsilon \hat{u}_{xx}) - f \|_{L^2(0,1)},
$$

and $w_I$ and $w_F$ are the certain weight parameters. The loss in (1.4) indeed involves the derivatives of $\hat{u}$ and it is handled via the so-called automatic differentiation (AD). In the last step, the procedure of searching for a good $\theta$ by minimizing the loss $\mathcal{L}(\theta, \mathcal{T})$ is called “training” where we usually use a gradient-based optimizer such as gradient descent, Adam, or L-BFGS.

Our approach to construct a two-layer NN solving (1.1) - (1.2) is closely related to the PINNs above, but it is a bit different as explained below; see an earlier work [23] where the NN closely related to ours is introduced:

To obtain an approximate solution $\overline{u}$ to (1.1) - (1.2), we employ a simple 2-layer NN (denoted by $\hat{u}$) multiplied by $x(x - 1)$ to enforce the boundary condition (1.2), in the form:

$$
\overline{u}(x; \theta) = x(x - 1) \hat{u}(x; \theta),
$$

where $\hat{u}$ is defined by the 2-layer NN,

$$
\begin{align*}
\text{input layer} : & \quad \mathcal{N}^0(x) = x \in \mathbb{R}, \\
\text{hidden layer} : & \quad \mathcal{N}^1(x) = \sigma(W^1 x + b^1) \in \mathbb{R}^{N_1}, \\
\text{output layer} : & \quad \mathcal{N}^2(x) = W^2 \mathcal{N}^1(x) \in \mathbb{R},
\end{align*}
$$

and it is handled via the so-called automatic differentiation (AD). In the last step, the procedure of searching for a good $\theta$ by minimizing the loss $\mathcal{L}(\theta, \mathcal{T})$ is called “training” where we usually use a gradient-based optimizer such as gradient descent, Adam, or L-BFGS.
i.e.,
\[
\hat{u}(x; \theta) = W^2 \sigma(W^1 x + b^1),
\]
(1.8)
for the parameters \( \theta = \{W^1 = (W^1_1 \cdots W^1_{N_1})^T, W^2 = (W^2_1 \cdots W^2_{N_1}), b^1 = (b^1_1 \cdots b^1_{N_1})^T \} \in \mathbb{R}^{N_1 \times 1} \times \mathbb{R}^{1 \times N_1} \times \mathbb{R}^{N_1} \). Here we choose the activation function as the logistic sigmoid,
\[
\sigma(x) = \frac{1}{1 + e^{-x}}.
\]
(1.9)

Because the boundary condition (1.2) is already embedded in the approximate solution (1.6), we define the loss function as
\[
L(\theta, T) = \frac{1}{|T|} \sum_{x \in T} \| F(x; \hat{u}, \hat{u}_x, \varepsilon \hat{u}_{xx}) - f \|^2_{L^2(0,1)},
\]
(1.10)
where the training set \( T \) is chosen as a set of scattered points in \((0, 1)\).

One big difference between the usual PINNs and our 2 layer modified PINNs is on the computing the derivatives of the NN approximation in the loss function. Using a fact on the sigmoid function that
\[
\sigma'(x) = \sigma(x)(1 - \sigma(x)),
\]
\[
\sigma''(x) = \sigma(x)(1 - \sigma(x))(1 - 2\sigma(x)),
\]
(1.11)
we compute
\[
\frac{d}{dx} \hat{u}(x; \theta) = \sum_{j=1}^{N_1} W^2_j W^1_j \sigma'(W^1_j x + b^1_j)
\]
\[
= \sum_{j=1}^{N_1} W^2_j W^1_j \sigma(W^1_j x + b^1_j)(1 - \sigma(W^1_j x + b^1_j)),
\]
\[
\frac{d^2}{dx^2} \hat{u}(x; \theta) = \sum_{j=1}^{N_1} W^2_j (W^1_j)^2 \sigma(W^1_j x + b^1_j)(1 - \sigma(W^1_j x + b^1_j))(1 - 2\sigma(W^1_j x + b^1_j)).
\]
(1.12)

Thanks to the simple structure (1.6) of our 2 layer modified PINNs, we use the derivatives of \( \hat{u}(x; \theta) \) above, and compute explicitly the loss function (1.10) without using the automatic differentiation (AD); hence our new 2 layer modified PINNs do not rely on the AD and their computational errors are independent of the AD. In addition, see the sections below where we use (1.6) (or the proper modifications of (1.6) by using the so-called boundary layer correctors) to define the loss function of each example we consider in this article.

In general, when the target function contains high-frequency components, i.e., when \( \varepsilon > 0 \) is small in our model problem (1.1), the PINN algorithms often fail to converge to the desirable solutions, because the so-called spectral bias phenomenon [3, 27]. Since the general learning process of neural networks rely on the smooth prior, the spectral bias leads to a failure to capture sharp transitions accurately or singular behaviors of the target solution function. More precisely, while the neural networks tend to learn low frequency components, it requires much time to fit high frequency components. Hence, without care, neural networks cannot fit the sharp transition caused by the boundary layer. In our
algorithm, we split our test function into two parts, slow and fast gradient components, and learn effectively the two components simultaneously. This approach is motivated from an enriched basis method in numerical PDEs [8, 9, 15, 16].

In Section 1.1 below, we first verify that the 2 layer modified PINNs work well for the (regular) problem (1.1) - (1.2) when the parameter is not small, e.g., $\varepsilon = 1$. Then, in Section 2 and 3, as the main work of our article, we consider different types of linear and non-linear singular perturbation problems in the form of (1.1) - (1.2) when the parameter $\varepsilon$ is small. As we will see below, the 2 layer modified PINNs (as well as the usual PINNs) do not capture well the singular behavior of perturbation problems, caused by the so-called boundary layers, and hence they fail to produce an accurate approximate solution for each singular perturbation. To overcome this difficulty of boundary layers, we first perform the boundary layer analysis of each singular perturbation problem, and find the so-called corrector function that exhibits the singular behavior of the problem inside the boundary layer. Then we construct our new semi-analytic Neural Networks, enriched by embedding the corrector function in the structure of 2 layer modified PINNs. Numerical simulations for each example below confirm that our new enriched 2 layer modified PINNs captures naturally the singular behavior of boundary layers and produce accurate approximations of singularly perturbed boundary value problems considered in this article.

1.1. **Elliptic and hyperbolic equations.** Our first task is to numerically confirm that the modified PINN (1.6) works as good as the usual PINN (1.3) for a certain class of boundary value problems. To this end, we introduce the following boundary value problem,

$$
\begin{align*}
- au_{xx} - bu_x + cu &= f(x), & 0 < x < 1, \\
u &= 0, & x = 0, 1.
\end{align*}
$$

(1.13)

By setting the coefficients $(a, b, c)$ as $(0, -1, 0)$, $(1, 1, 0)$, and $(1, 0, 1)$, we consider (1.13) as the hyperbolic, convection-diffusion, and reaction-diffusion equation.

We apply our 2-layer modified PINN with the sigmoid function, defined in (1.6), to approximate solutions to the boundary value problem (1.13). The computation results below in Fig. 1.1 show that our 2 layer modified PINNs works well and produce accurate approximations for the solutions to (1.13); see [23, 18, 24, 28] as well for the comparable computational results of the usual PINNs.

### 2. Enriched Neural Network: Linear Singularly Perturbed Boundary Value Problems

In this section, we study a linear singularly perturbed boundary value problem. We first consider the convection-diffusion equations where the boundary layer occurs near the outflow boundary only. The reaction-diffusion equations are then investigated where the boundary layer takes place at each boundary.

2.1. **Convection-diffusion equations.** As a first project, we consider the following singularly perturbed convection-diffusion equation,

$$
\begin{align*}
- \varepsilon u_{xx} - u_x &= f(x), & 0 < x < 1, \\
\varepsilon u_x &= 0, & x = 0, 1.
\end{align*}
$$

(2.1)
Our main object here is to construct the semi-analytic PINN for the singular perturbation problem above, and compare its performance with the PINN:

In order to find the so-called corrector, which is suitable to enrich the 2-layer PINNs for the problem (2.1) above, we first notice, by formally replacing $\varepsilon$ by 0 in (2.1), that the limit problem of (2.1) at the vanishing diffusivity at $\varepsilon = 0$ is

$$
\begin{align*}
- u^0_x &= f(x), \quad 0 < x < 1, \\
 u^0 &= 0, \quad x = 1.
\end{align*}
$$

(2.2)

Here we impose the so-called *inflow* boundary condition at $x = 1$ for $u^0$ (see, e.g., [11] for more information), and hence we find the formal limit $u^0$ in the form,

$$
u^0(x) = \int_0^1 f \, dx.
$$

(2.3)

Now, by performing the matching asymptotics for (2.1), we observe that the boundary layer of size $\varepsilon$ occurs near the *outflow* boundary at $x = 0$, and find that the asymptotic equation (with respect to the small $\varepsilon$) for the corrector $\varphi$, which approximate the difference $u^\varepsilon - u^0$, is given in the form,

$$
\begin{align*}
-\varepsilon \varphi_{xx} - \varphi_x &= 0, \quad 0 < x < 1, \\
 \varphi &= -u^0, \quad x = 0.
\end{align*}
$$

(2.4)

It is well-known, see, e.g., [11], that the corrector $\varphi$ is given in the form

$$
\varphi(x) = -u^0(0) e^{-x/\varepsilon} + e.s.t.,
$$

(2.5)

where the *e.s.t.* denotes an exponentially small term with respect to the small perturbation parameter $\varepsilon$.

Figure 1.1. Comparison on the exact solution of (1.13) and our neural network approximation of (1.13) using the 2-layer modified PINN. For numerical experiments, we set $f = 1$ in (1.13)

(a) 1st order equation with $a = 0, c = 0, b = -1$

(b) Convection-diffusion eqn with $a = 1, b = 1, c = 0$

(c) Reaction-diffusion eqn with $a = 1, b = 0, c = 1$
Employing the simple energy estimates on the difference \( u^\varepsilon - (u^0 + \varphi) \), and then using the smallness of the corrector \( \varphi \), we then further notice that
\[
\|u^\varepsilon - (u^0 + \varphi)\|_{L^2((0,1))} \leq \kappa \varepsilon, \\
\|u^\varepsilon - u^0\|_{L^2((0,1))} \leq \kappa \varepsilon^{\frac{1}{2}},
\]
for a constant \( \kappa > 0 \) independent of \( \varepsilon \). The convergence results above denote first that the diffusive solution \( u^\varepsilon \) converges, as \( \varepsilon \) tends to 0, to the limit solution \( u^0 \) as fast as \( \varepsilon \).

Second, we also infer from the convergence results that the corrector \( \varphi \) exhibits the singular behavior of \( u^\varepsilon \) at a small diffusivity \( \varepsilon \), i.e., the diffusive solution \( u^\varepsilon \) is decomposed into the sum of fast (decaying part) \( \varphi \), and the slow part \( u^0 \).

Inspired by the analysis above, we modify the 2-layer PINN by using the profile of the corrector \( \varphi \), and define our new semi-analytic 2-layer PINN for the problem (2.1) in the form,
\[
\tilde{u}(x; \theta) = (x - 1)(\hat{u}(x; \theta) - \hat{u}(0; \theta) e^{-x/\varepsilon}),
\]
where \( \hat{u} \) is exactly the same as in (1.8), which was used for the modified 2-layer PINN approximation \( \overline{u} \) in (1.6). Here we observe that the boundary value of \( \tilde{u} \) at \( x = 0 \) is ensured to be zero, thanks to the factor \( -\hat{u}(0; \theta) \) multiplied to the exponentially decaying boundary layer function; see and compare the difference of (2.7) and (1.6) with or without the factor \( x \).

The loss for this problem (2.1) is defined by
\[
\mathcal{L}(\theta, \mathcal{T}) = \frac{1}{|\mathcal{T}|} \sum_{x \in \mathcal{T}} \| -\varepsilon \tilde{u}_{xx} - \tilde{u}_x - f \|_{L^2((0,1))}^2,
\]
where the training set \( \mathcal{T} \) is chosen as a set of scattered points in \((0, 1)\). The derivatives of our enriched 2 layer approximation \( \tilde{u} \) are given by
\[
\frac{d}{dx} \tilde{u} = (\hat{u}(x; \theta) - \hat{u}(0; \theta) e^{-x/\varepsilon}) + (x - 1)\left(\hat{u}_x(x; \theta) + \frac{1}{\varepsilon} \hat{u}(0; \theta) e^{-x/\varepsilon}\right), \\
\frac{d^2}{dx^2} \tilde{u} = 2\left(\hat{u}_x(x; \theta) + \frac{1}{\varepsilon} \hat{u}(0; \theta) e^{-x/\varepsilon}\right) + (x - 1)\left(\hat{u}_{xx}(x; \theta) - \frac{1}{\varepsilon^2} \hat{u}(0; \theta) e^{-x/\varepsilon}\right).
\]

Using the fact that the exponentially decaying function \( e^{-x/\varepsilon} \) satisfies the model equation (2.1), we find that, for the loss function in (2.8),
\[
-\varepsilon \tilde{u}_{xx} - \tilde{u}_x - f = -\varepsilon(x - 1)\tilde{u}_{xx}(x; \theta) - (x - 1 + 2\varepsilon)\tilde{u}_x(x; \theta) - \hat{u}(x; \theta) - \hat{u}(0; \theta) e^{-x/\varepsilon} - f,
\]
where the derivatives of \( \hat{u} \) are given in (1.12). One important remark here is that all the terms in (2.10) stay bounded as \( \varepsilon \) vanishes, and hence our enriched 2 layer PINN produces an accurate approximation \( \tilde{u} \) for (2.1), independent of the small parameter \( \varepsilon \).

Now, we compare the performance of the usual 2-layer (modified) PINN approximation \( \overline{u} \) in (1.6) and that of our new semi-analytic 2-layer PINN approximation \( \tilde{u} \): The Fig. 2.1 shows that the usual PINN, constructed in Section 1.1 fails to approximate the solution of the singularly perturbation (2.2). The numerical results below in Fig. 2.2...
and Table 1 confirm that the semi-analytic enriched 2-layer PINN performs much better than the usual PINN method, thanks to the corrector function embedded in the scheme. Note that our semi-analytic PINN enriched with corrector produces stable and accurate approximate solutions, independent of the small parameter $\varepsilon$, as shown in the Fig. 2.3.

![Convection-Diffusion equation](image)

**Figure 2.1.** The exact solution and the approximation solution of the convection-diffusion equation (2.1) with $\varepsilon = 10^{-4}$ are displayed. Here, the (plain) modified 2-layer PINN (1.6) is used for the predicted solution. The predicted solution is not close to the corresponding exact solution.
2.2. Reaction-diffusion equations. We construct in this Section the semi-analytic PINN for the following singularly perturbed reaction-diffusion equation:

\[
\begin{aligned}
-\varepsilon u_{xx} + u^\varepsilon &= f(x), & 0 < x < 1, \\
 u^\varepsilon &= 0, & x = 0, 1.
\end{aligned}
\]
The corresponding limit, as \( \varepsilon \to 0 \), of \( u^\varepsilon \) (solution of (2.11)) is obtained by formally replacing \( \varepsilon \) by 0 in (2.11):

\[
u^0 = f.
\]

(2.12)

Performing the matching asymptotics, we find that the size of boundary layer for (2.11) is of order \( \varepsilon^{1/2} \), and it appears near both ends of the domain, i.e., near \( x = 0 \) and \( x = 1 \). Writing the asymptotic equation of the difference \( u^\varepsilon - u^0 \), we find the equation for the corrector \( \varphi \sim u^\varepsilon - u^0 \) as

\[
\begin{cases}
-\varepsilon \varphi_{xx} + \varphi = 0, & 0 < x < 1, \\
\varphi = -u^0, & x = 0, 1.
\end{cases}
\]

(2.13)

By solving (2.13), we find that

\[
\varphi(x) = -u^0(0) e^{-x/\sqrt{\varepsilon}} - u^0(1) e^{-(1-x)/\sqrt{\varepsilon}} + \text{e.s.t.},
\]

(2.14)
i.e., the fast decaying part \( \varphi \) of \( u^\varepsilon \) is the sum of two exponentially decaying functions from each part of the boundary points \( x = 0 \) and \( x = 1 \), scaled by the stretched variables \( x/\sqrt{\varepsilon} \) and \( (1-x)/\sqrt{\varepsilon} \) up to an exponentially small term.

Employing the simple energy estimates on the difference \( u^\varepsilon - (u^0 + \varphi) \), and then using the smallness of the corrector \( \varphi \), we then further notice that

\[
\| u^\varepsilon - (u^0 + \varphi) \|_{L^2((0,1))} \leq \kappa \varepsilon,
\]

\[
\| u^\varepsilon - u^0 \|_{L^2((0,1))} \leq \kappa \varepsilon^{\frac{1}{4}},
\]

(2.15)
for a constant \( \kappa > 0 \) independent of \( \varepsilon \). Hence we notice here that the corrector \( \varphi \) exhibits the singular behavior of \( u^\varepsilon \) at a small diffusivity \( \varepsilon \), i.e., the diffusive solution \( u^\varepsilon \) is decomposed into the sum of fast (decaying part) \( \varphi \), and the slow part \( u^0 \).

Based on the boundary layer analysis above, using the corrector \( \varphi \), we construct the new semi-analytic enriched 2-layer PINN for the problem (2.11) in the form,

\[
\tilde{u}(x; \theta) = \hat{u}(x; \theta) - \hat{u}(0; \theta) e^{-x/\sqrt{\varepsilon}} - \hat{u}(1; \theta) e^{-(1-x)/\sqrt{\varepsilon}},
\]

(2.16)
where \( \hat{u} \) is exactly the same as in (1.8). Because the effect of the exponentially decaying function \( e^{-x/\sqrt{\varepsilon}} \) (or \( e^{-(1-x)/\sqrt{\varepsilon}} \)) on the boundary point at \( x = 1 \) (or \( x = 0 \)) is exponentially small with respect to the small \( \varepsilon \), the enriched PINN approximation \( \tilde{u} \) attains the zero boundary value at \( x = 0, 1 \), up to an exponentially small (computationally negligible) error.

The loss for this problem (2.11) is defined by

\[
\mathcal{L}(\theta, \mathcal{T}) = \frac{1}{|\mathcal{T}|} \sum_{x \in \mathcal{T}} \| -\varepsilon \tilde{u}_{xx} - \tilde{u} - f \|^2_{L^2((0,1))},
\]

(2.17)
where the training set \( \mathcal{T} \) is chosen as a set of scattered points in \((0,1)\), and

\[
-\varepsilon \tilde{u}_{xx} - \tilde{u} - f = -\varepsilon (x-1) \hat{u}_{xx}(x; \theta) - \hat{u}(x; \theta) - f,
\]

(2.18)
with \( \hat{u}_{xx} \) in (1.12). In derivation of (2.18), we used the fact that the exponentially decaying functions \( e^{-x/\sqrt{\varepsilon}} \) and \( e^{-(1-x)/\sqrt{\varepsilon}} \) satisfy the equation (2.11).
Now, we notice from Fig. 2.4 and Table 1 below that our semi-analytic 2-layer PINN, enriched with the corrector, approximates well the solution of (2.11):

\[
\text{Figure 2.4. The exact sol. and the approx. sol. of the reaction-diffusion equation (2.11) with } f = 1 \text{ and } \varepsilon = 10^{-4} \text{ are displayed. The semi-analytic 2-layer PINN (2.16) is used for the predicted sol. The predicted sol. is close to the corresponding exact sol. with relative } L^2 \text{ error: } 1.245 \times 10^{-4}.\]

3. ENRICHED NEURAL NETWORK: NONLINEAR SINGULARLY PERTURBED BOUNDARY VALUE PROBLEMS

In this section, we propose the 2-layer modified PINN for the nonlinear singularly perturbed boundary value problems. We first consider the convection-diffusion equations with a non-linear reaction. The shape of the boundary layer profile is similar to that in the linear case. We then examine the stationary Burgers’ equation with a small viscosity parameter. Since the analysis and computation of singularly perturbed Burgers’ equations are not straightforward, our neural network requires careful numerical treatment.

3.1. CONVECTION-DIFFUSION EQUATIONS WITH A NON-LINEAR REACTION. In this section, we apply our methodology of semi-analytic enriched PINNs to a certain non-linear equation for which we can determine the profile of boundary layer. Note that the corresponding boundary layer analysis is not straightforward at all; see [17].

We consider the singularly perturbed convection-diffusion equation with a non-linear reaction term:

\[
\begin{aligned}
\varepsilon u_{xx}^\varepsilon - u_x^\varepsilon + (u^\varepsilon)^3 &= f(x), & 0 < x < 1, \\
 u^\varepsilon &= 0, & x = 0, 1. \\
\end{aligned}
\]  

(3.1)

The corresponding limit problem at \( \varepsilon = 0 \) is given by

\[
\begin{aligned}
-u_x^0 + (u^0)^3 &= f(x), & 0 < x < 1, \\
 u^0 &= 0, & x = 1. \\
\end{aligned}
\]  

(3.2)
The well-posedness and the regularity of (3.1) and (3.2) are well-studied and here we omit any further discussion on those issues; see, e.g., [17] for the detailed information.

Although it is a non-linear problem, the boundary layer associated with (3.1) is linear. In fact, by performing the matching asymptotics, one can verify that the boundary layer of size $\varepsilon$ occurs near the outflow boundary at $x = 0$, just like the linear convection-diffusion problem (2.1). Moreover we find that the asymptotic equation (with respect to the small $\varepsilon$) for the corrector $\varphi$, which approximate the difference $u^\varepsilon - u^0$, is given in the form,

$$
\begin{cases}
-\varepsilon \varphi_{xx} - \varphi_x = 0, & 0 < x < 1, \\
\varphi = -u^0, & x = 0,
\end{cases}
$$

and hence the corrector $\varphi$ is explicitly written in the form,

$$
\varphi(x) = -u^0(0) e^{-x/\varepsilon} + \text{e.s.t.}
$$

Concerning the detailed boundary layer analysis as well as the convergence results of $u^\varepsilon$ to $u^0$, see, e.g., [17].

By enriching the 2-layer PINNs with the profile of the corrector $\varphi$ above, we define the new semi-analytic 2-layer PINN for the problem (3.1) in the form,

$$
\tilde{u}(x; \theta) = (x - 1) (\hat{u}(x; \theta) - \hat{u}(0; \theta) e^{-x/\varepsilon}),
$$

with $\hat{u}$ as in (1.8).

The loss for this problem is defined by

$$
L(\theta, \mathcal{T}) = \frac{1}{|\mathcal{T}|} \sum_{x \in \mathcal{T}} \| - \varepsilon \tilde{u}_{xx} - \tilde{u}_x + \tilde{u}^3 - f \|_{L^2(0, 1)}^2,
$$

where the training set $\mathcal{T}$ is chosen as a set of scattered points in $(0, 1)$. The derivatives of our enriched 2 layer approximation $\tilde{u}$ are exactly the same as in (2.9), and thus, using the fact that the exponentially decaying function $e^{-x/\varepsilon}$ satisfies the equation (3.3), we find for the loss function (3.6) that

$$
-\varepsilon \tilde{u}_{xx} - \tilde{u}_x + \tilde{u}^3 - f \\
= -\varepsilon(x - 1) \tilde{u}_{xx}(x; \theta) - (x - 1 + 2\varepsilon) \tilde{u}_x(x; \theta) - \hat{u}(x; \theta) - \hat{u}(0; \theta) e^{-x/\varepsilon} + (x - 1)^3 (\hat{u}(x; \theta) - \hat{u}(0; \theta) e^{-x/\varepsilon})^3 - f.
$$

Here the derivatives of $\hat{u}$ are given in (1.12). Note that all the terms in (3.7) stay bounded as $\varepsilon$ vanishes, and hence our semi-analytic enriched 2 layer PINN produces an accurate approximation $\tilde{u}$ for (3.1), independent of the small parameter $\varepsilon$.

We observe from Fig. 3.1 and Table 1 below that our semi-analytic 2-layer PINN, enriched with the corrector, approximates well the solution of (3.1):
FIGURE 3.1. The exact sol. and the approx. sol. of (3.1) with $f = 1$ and $\varepsilon = 10^{-4}$ are displayed. Here the semi-analytic 2-layer PINN (3.5) is used for the predicted sol. The predicted sol. is close to the corresponding exact sol. with relative $L^2$ error: $5.474 \times 10^{-2}$.

3.2. Stationary Burgers’ equation. In this section, we consider the stationary 1D Burgers’ equation in a bounded interval $(0, 1)$ as

$$
\begin{align*}
-\varepsilon u_{xx}^\varepsilon + u_x^\varepsilon u_x^\varepsilon &= f(x), \quad 0 < x < 1, \\
        u_x^\varepsilon &= -\alpha, \quad x = 0, \\
        u_x^\varepsilon &= -\beta, \quad x = 1.
\end{align*}
$$

Here $0 < \varepsilon \ll 1$ is a small viscosity parameter, $f$ is a smooth data, independent of $\varepsilon$, and $\alpha$, $\beta$ are positive constants. We set the boundary values of $u^\varepsilon$ at $x = 0, 1$ as negative numbers so that $u^\varepsilon < 0$ for all $0 \leq x \leq 1$. Hence, consequently, the convection $u_x^\varepsilon u_x^\varepsilon$ occurs always in one direction from right to left and the boundary layer occurs near only the out-flow boundary at $x = 0$.

For the sake of convenience on our computations below, we set $\alpha = 1$ and $\beta = 1$. We also assume that the data $f$ satisfies the following condition so that the limit solution $u^0$ is well-defined and explicitly written as in (3.11) below:

$$
\int_1^x f(x') \, dx' > -\frac{1}{2}, \quad \text{for any } 0 \leq x \leq 1;
$$

see, e.g., [6, 10] for more information and full boundary layer analysis for this version of stationary 1D Burgers’ equation.

The corresponding limit (inviscid) problem is obtained by setting $\varepsilon = 0$ in (3.8) and imposing the in-flow boundary condition at $x = 1$:

$$
\begin{align*}
    u^0 u_x^0 &= f(x), \quad 0 < x < 1, \\
    u^0(1) &= -1.
\end{align*}
$$
The formal limit $u^0$, a solution of (3.10), is given in the form,

$$u^0(x) = -\left(2 \int_1^x f(x') \, dx' + 1\right)^{\frac{1}{2}},$$

and hence we infer from (3.9) that

$$u^0 < 0 \quad \text{for any } 0 \leq x \leq 1. \quad (3.12)$$

Performing the matching asymptotics, we find that the size of boundary layer for (3.8) is of order $\varepsilon$, and it appears near the out-flow boundary at $x = 0$. Writing the asymptotic equation of the difference $u^\varepsilon - u^0$, we find the following (non-linear) asymptotic equation for $\varphi \sim u^\varepsilon - u^0$ as

$$
\begin{cases}
-\varepsilon \varphi_{xx} + (u^0(0) + \varphi) \varphi_x = 0, & 0 < x < 1, \\
\varphi = -1 - u^0, & x = 0, \\
\varphi \to 0 \quad \text{as } x \to 1.
\end{cases}
$$

(3.13)

By integrating (3.13) from $x' = 1$ to $x' = x$, we write the first order equation,

$$
\begin{cases}
-\varepsilon \varphi_x + u^0(0) \varphi + \frac{1}{2} \varphi^2 = 0, & 0 < x < 1, \\
\varphi = -1 - u^0, & x = 0.
\end{cases}
$$

(3.14)

Then, solving the equation above, we find

$$\varphi(x) = \frac{2u^0(0) \left(1 + u^0(0)\right) e^{u^0(0)x/\varepsilon}}{1 - u^0(0) - (1 + u^0(0)) e^{u^0(0)x/\varepsilon}}. \quad (3.15)$$

The fast decaying part of $u^\varepsilon$ (near the out-flow boundary at $x = 0$) is hence described by the corrector $\varphi$ above. Moreover, by performing the boundary layer analysis as in, e.g., [6, 10], one can verify that

$$\|u^\varepsilon - (u^0 + \varphi)\|_{L^2(0,1)} \leq \kappa \varepsilon, \quad (3.16)$$

for a constant $\kappa > 0$ independent of $\varepsilon$, and hence obtain the vanishing viscosity limit as well:

$$\|u^\varepsilon - u^0\|_{L^2(0,1)} \leq \kappa \varepsilon^{\frac{1}{2}}. \quad (3.17)$$

Now, thanks to the asymptotic analysis above, we construct below the semi-analytic 2-layer PINNs enriched by the profile of the corrector $\varphi$:

We first normalize the boundary value of $\varphi$ in (3.15), and introduce the normalized corrector $\tilde{\varphi}$, which describe the boundary layer profile for (3.8), in the form,

$$\tilde{\varphi}(x) = \frac{\varphi(x)}{\varphi(0)}. \quad (3.18)$$

Then, we define our new semi-analytic enriched 2-layer PINN for the problem (3.8) as

$$\tilde{u}(x; \theta) = (x - 1) \hat{u}(x; \theta) + \tilde{\varphi}(x) \hat{u}(0; \theta) - g(x)$$

(by setting $\tilde{\varphi}^*(x) := \tilde{\varphi}(x) \hat{u}(0; \theta)$)

$$= (x - 1) \hat{u}(x; \theta) + \tilde{\varphi}^*(x) - g(x), \quad (3.19)$$
where \( \hat{u} \) is exactly the same as in (1.8), which was used for the usual 2-layer PINNs approximation \( \varpi \) in (1.6), and \( g(x) \) is a simple boundary lifting function,

\[
g(x) = (\beta - \alpha)x + \alpha = 1, \quad \text{(by using } \alpha = \beta = 1)\; (3.20)
\]

note that the choice of this lifting \( g = 1 \) is for our convenience, but any other lifting, which gives the value 1 at \( x = 0, 1 \), produces the same computational results as those we obtain in this article.

The loss for this problem is defined by

\[
\mathcal{L}(\theta, \mathcal{T}) = \frac{1}{|\mathcal{T}|} \sum_{x \in \mathcal{T}} \| -\varepsilon \tilde{u}_{xx} + \tilde{u}_{x} - f \|^2_{L^2(0,1)},\tag{3.21}
\]

where the training set \( \mathcal{T} \) is chosen as a set of scattered points in \( (0,1) \). The derivatives of our enriched 2 layer approximation \( \tilde{u} \) are given by

\[
\frac{d}{dx} \tilde{u} = \hat{u}(x; \theta) + (x-1)\hat{u}_x(x; \theta) + \tilde{\varphi}_x^* (x)\tag{3.22}
\]

\[
\frac{d^2}{dx^2} \tilde{u} = 2\hat{u}_x(x; \theta) + (x-1)\hat{u}_{xx}(x; \theta) + \tilde{\varphi}_{xx}^* (x).\tag{3.23}
\]

We recall from (3.16) that the asymptotic expansion of \( u^\varepsilon \) is well defined in the sense that

\[
u^\varepsilon = u^0 + \varphi + O(\varepsilon).\tag{3.24}
\]

Then, because the corrector \( \varphi \) satisfies the equation (3.13)_1, and because we use \( \tilde{u} \) to approximate \( u^\varepsilon \), i.e., \( u^\varepsilon \simeq \tilde{u} \), we observe that the corrector \( \varphi \) satisfies

\[
-\varepsilon \varphi_{xx} + (\tilde{u} - (u^0 - u^0(0))) \varphi_x = O(\varepsilon),\tag{3.25}
\]

which is equivalent, in terms of the normalized corrector, to

\[
-\varepsilon \tilde{\varphi}_{xx}^* + \tilde{u}\tilde{\varphi}_x^* - (u^0 - u^0(0))\tilde{\varphi}_x^* = O(\varepsilon).\tag{3.26}
\]

Finally, using (3.22) and the fact that the normalized corrector \( \tilde{\varphi} \) satisfies the equation (3.25) above, we find for the loss function (3.21) that

\[
-\varepsilon \tilde{u}_{xx} + \tilde{u}_{x} - f = -2\varepsilon \hat{u}_x - \varepsilon(x-1)\hat{u}_{xx} + ((x-1)\hat{u})_x((x-1)\hat{u} - \tilde{\varphi}^* - g) + (u^0 - u^0(0))\tilde{\varphi}_x^*,\tag{3.27}
\]

where the derivatives of \( \hat{u} \) are given in (1.12). Because \( u^0 - u^0(0) \simeq O(\varepsilon) \) and \( \tilde{\varphi}_x \simeq \varepsilon^{-1}e^{-x/\varepsilon} \) from (3.15), we notice that all the terms in (3.27) stay bounded as \( \varepsilon \) vanishes, and hence our semi analytic 2 layer PINN produces an accurate approximation \( \tilde{u} \) for (3.8), independent of the small parameter \( \varepsilon \).

To measure performance of the semi-analytic 2-layer PINN approximation, we manufacture an exact solution to the Burgers’ equations. For this, we use a numerical solution with a large number of discretization point since the exact solution is not available in general. More precisely, we implement the Burgers’ equations, (3.8), with \( f = -1 \) using the spectral element method with the number of elements \( M = 2048 \). Since the collocation points (input points) of the neural network is much less than \( M = 2048 \), we exploit spline interpolation when comparing the manufactured solution with the predicted solution. Since a lot of collocation points are used in the spline interpolation, the numerical error from the
interpolation is much less than the approximation error from the semi-analytic enriched 2-layer PINN approximation.

We observe from Fig. 3.2 and 3.3, and Table 1 below that our semi-analytic 2-layer PINN, enriched with the corrector, approximates well the solution of (3.8). Our semi-analytic PINN produces stable and accurate approximate solutions, independent of the small parameter $\varepsilon$, as shown in the Fig. 3.3.

![Burgers' equation](image1)

**Figure 3.2.** The exact sol. and the approx. sol. of (3.8) with $f = -1$ and $\varepsilon = 10^{-4}$ are displayed. Here the semi analytic 2-layer PINN (3.19) is used for the predicted sol. The predicted sol. is close to the corresponding exact sol. with relative $L^2$ error: $2.599 \times 10^{-3}$.

![Boundary layer transition](image2)

**Figure 3.3.** Boundary layer transition for (3.8) with respect to different viscosity parameters, $\varepsilon$, is displayed. As $\varepsilon$ decreases, thickness of the boundary layer becomes smaller.
Table 1. Relative $L^2$ errors. Here, CD, LRD, NCD, and BE stand for, in turns, the convection-diffusion, linear reaction-diffusion, nonlinear convection-diffusion equations, and Burgers’ equation respectively. For the CD equation, ECD and CCD means respectively the semi-analytic enriched PINN and the 2 layer PINN without using the corrector. We set $N$ to be the number of data points. In this table, the maximum number of iterations is 50,000, but early stopping is used to avoid over-fitting.

| N  | ECD ($\varepsilon = 10^{-4}$) | CCD ($\varepsilon = 10^{-4}$) | LRD ($\varepsilon = 10^{-8}$) | NCD ($\varepsilon = 10^{-4}$) | BE ($\varepsilon = 10^{-4}$) |
|----|-------------------------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|
| 50 | $7.295 \times 10^{-3}$        | $9.977 \times 10^{-1}$        | $1.245 \times 10^{-4}$        | $5.474 \times 10^{-2}$        | $2.599 \times 10^{-3}$        |
| 100| $3.724 \times 10^{-3}$        | $9.931 \times 10^{-1}$        | $5.382 \times 10^{-4}$        | $4.885 \times 10^{-3}$        | $1.982 \times 10^{-3}$        |
| 200| $2.148 \times 10^{-3}$        | $9.813 \times 10^{-1}$        | $4.245 \times 10^{-4}$        | $5.735 \times 10^{-3}$        | $1.474 \times 10^{-3}$        |
| 400| $9.310 \times 10^{-4}$        | $9.550 \times 10^{-1}$        | $5.317 \times 10^{-4}$        | $8.912 \times 10^{-4}$        | $9.422 \times 10^{-4}$        |
[3] Yuan Cao, Zhiying Fang, Yue Wu, Ding-Xuan Zhou, and Quanquan Gu. Towards Understanding the Spectral Bias of Deep Learning. Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, 8 (2021).

[4] Z. Chen, D. Xiu. On generalized residual network for deep learning of unknown dynamical systems. Journal of Computational Physics, 438 (2021).

[5] Z. Chen, V. Churchill, K. Wu, D. Xiu. Deep neural network modeling of unknown partial differential equations in nodal space. Journal of Computational Physics 449 (2022).

[6] Junho Choi, Chang-Yeol Jung, and Hoyeon Lee. On boundary layers for the Burgers equations in a bounded domain. Commun. Nonlinear Sci. Numer. Simul., 67:637–657, 2019.

[7] W. E and B. Yu. The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems. Communications in Mathematics and Statistics, 6 (2018), pp. 1–12.

[8] G.-M. Gie, C.-Y. Jung, and H. Lee. Enriched Finite Volume approximations of the plane-parallel flow at a small viscosity. Journal of Scientific Computing, 84, 7 (2020).

[9] G.-M. Gie, C.-Y. Jung, and H. Lee. Semi-analytic time differencing methods for singularly perturbed initial value problems. Numerical Methods for Partial Differential Equations, 38, 5, 1367 - 139, 2022.

[10] G.-M. Gie, C.-Y. Jung, and H. Lee. Semi-analytic shooting methods for Burgers’ equation. Accepted in Journal of Computational and Applied Mathematics.

[11] G.-M. Gie, M. Hamouda, C.-Y. Jung, and R. Temam, Singular perturbations and boundary layers, volume 200 of Applied Mathematical Sciences. Springer Nature Switzerland AG, 2018. https://doi.org/10.1007/978-3-030-00638-9

[12] H. Han and R. B. Kellogg, A method of enriched subspaces for the numerical solution of a parabolic singular perturbation problem. In: Computational and Asymptotic Methods for Boundary and Interior Layers, Dublin, pp.46-52 (1982).

[13] J. Han, Y. Lee. Hierarchical learning to solve partial differential equations using physics-informed neural networks. arXiv preprint, arXiv:2112.01254 (2021).

[14] M. H. Holmes, Introduction to perturbation methods, Springer, New York, 1995.

[15] Youngjoon Hong, Chang-Yeol Jung, and Jacques Laminie. Singularly perturbed reaction-diffusion equations in a circle with numerical applications. Int. J. Comput. Math., 90(11):2308–2325, 2013.

[16] Youngjoon Hong, Chang-Yeol Jung, and Roger Temam. On the numerical approximations of stiff convection-diffusion equations in a circle. Numer. Math., 127(2):291–313, 2014.

[17] Chang-Yeol Jung and Du Pham. Singular perturbation of semi-linear reaction-convection equations in a channel and numerical applications. Advances in Differential Equations. Vol.12, no. 3, 2007.

[18] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning. Nature Reviews Physics, 3 (6) (2021) 422–440.

[19] E. Kharazmi, Z. Zhang, G. E. Karniadakis. Variational physics-informed neural networks for solving partial differential equations. arXiv preprint, arXiv:1912.00873 (2019).

[20] E. Kharazmi, Z. Zhang, and G. E. Karniadakis. hp-VPINNs: Variational physics-informed neural networks with domain decomposition. Comput. Methods in Appl. Mech. Eng., 374 (2021).

[21] S. Kollmannsberger, D. D’Angella, M. Jokeit, L. Herrmann. Deep Learning in Computational Mechanics. Studies in Computational Intelligence, Springer International Publishing, Cham (2021). https://doi.org/10.1007/978-3-030-76587-3

[22] L. Lu, M. Dao, P. Kumar, U. Ramamurty, G. E. Karniadakis, S. Suresh. Extraction of mechanical properties of materials through deep learning from instrumented indentation. Proceedings of the National Academy of Sciences, 117 (13) (2020) 7052–7062.

[23] I.E. Lagaris, A. Likas, and D.I. Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. IEEE Trans Neural Netw., 1998;9(5):987-1000.

[24] 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.
[25] X. Meng, Z. Li, D. Zhang, G. E. Karniadakis. Ppinn: Parareal physics-informed neural network for time-dependent pdes. *Computer Methods in Applied Mechanics and Engineering*, 370 (2020).

[26] R. E. O’Malley, *Singularly perturbed linear two-point boundary value problems*. SIAM Rev. 50 (2008), no. 3, pp 459-482.

[27] Nasim Rahaman, Aristeide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the Spectral Bias of Neural Networks. *Proceedings of the 36th International Conference on Machine Learning*, 97 (2019).

[28] M. Raissi and G. E. Karniadakis. Hidden physics models: machine learning of nonlinear partial differential equations. *J. Comput. Phys.*, 357 (2018), pp. 125–141.

[29] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.*, 378 (2019), pp. 686–707.

[30] S. H. Rudy, S. L. Brunton, J. L. Proctor, J. N. Kutz. Data-driven discovery of partial differential equations. *Science advances*, 3 (4) (2017).

[31] K. Shukla, P. C. Di Leoni, J. Blackshire, D. Sparkman, G. E. Karniadakis. Physics-informed neural network for ultrasound nondestructive quantification of surface breaking cracks. *Journal of Nondestructive Evaluation*, 39 (3) (2020) 1–20.

[32] S. Shih and R. B. Kellogg, *Asymptotic analysis of a singular perturbation problem*. SIAM J. Math. Anal. 18 (1987), pp. 1467-1511.

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

[34] Mario De Florio, Enrico Schiassi, and Roberto Furfaro. Physics-informed neural networks and functional interpolation for stiff chemical kinetics. *Chaos* 32, 063107 (2022).

[35] T. Qin, Z. Chen, J. D. Jakeman, D. Xiu, Data-driven learning of nonautonomous systems. *SIAM Journal on Scientific Computing*, 43 (3) (2021).

[36] S. Wang, S. Sankaran, P. Perdikaris. Respecting causality is all you need for training physics-informed neural networks. *arXiv preprint*, arXiv:2203.07404 (2022).

[37] R. Xu, D. Zhang, M. Rong, N. Wang. Weak form theory-guided neural network (tgmn-wf) for deep learning of subsurface single-and two-phase flow. *Journal of Computational Physics*, 436 (2021).

[38] L. Yang, X. Meng, G. E. Karniadakis. B-pinns: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data. *Journal of Computational Physics*, 425 (2021).

[39] 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 Department of Mathematics, University of Louisville, Louisville, KY 40292

2 Department of Mathematics, Sungkyunkwan University

3 Department of Mathematical Sciences, Ulsan National Institute of Science and Technology, Ulsan 44919, Korea

Email address: gungmin.gie@louisville.edu
Email address: hongyj@skku.edu
Email address: cjung@unist.ac.kr