Deep learning methods for singular variational problems with Lavrentiev phenomena

Dong Wang[a,b], Xianmin Xu[c], Boyi Zou[a]

[a] School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, Guangdong 518172, China
[b] Shenzhen International Center for Industrial and Applied Mathematics, Shenzhen Research Institute of Big Data, Guangdong 518172, China
[c] LSEC, ICMSEC, NCMIS, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China

Abstract

Deep learning has gained significant development in the field of scientific computing, especially in its application to solve problems related to differential operators using deep neural networks. However, the utilization of neural networks to solve problems involving singularities still faces challenges. In this paper, we discussed the failure of deep learning methods for the singular variational problems exhibiting the Lavrentiev phenomenon. For such problems, we show the standard deep Ritz method and some variants fail to detect the singular minimizers. We then introduce a guiding term that renders the neural network to explore solutions as desired during training. Numerical experiments demonstrate that the method achieves much better approximations than the previous methods. Furthermore, we apply the same algorithm to solve problems with regular solutions to show the robustness of the proposed method.

Keywords: Deep neural networks; regularity; Lavrentiev phenomenon;

1. Introduction

Artificial intelligence for science has been receiving increasing attention in recent years, especially in the utilization of deep neural networks (DNNs) to solve problems related to differential operators. Numerous works have gained significant attention, such as physics-informed neural network (PINN) [1], deep Galerkin method (DGM) [2], deep Ritz method (DRM) [3], weak adversarial network (WAN) [4, 5, 6, 7], tensor neural network (TNN) [8, 9], and many others [10, 11, 12, 13, 14, 15, 16, 17, 18, 19]. For a more comprehensive review, we refer to [20, 21, 22, 23] and references therein.

It is known that deep learning methods have advantages in dealing with high dimensional problems which can not be solved efficiently by traditional numerical methods due to the curse of dimensionality. In addition to high dimensional problems, standard methods are inefficient for problems with singularity. Therefore, it would be interesting to see whether the deep learning methods have advantages in solving problems with singularity. This is the main problem we study in this work. To be more specific, we consider problems exhibiting the Lavrentiev phenomenon (which will be specified in Section 2.1 and 3). These problems pose a challenge when attempting to solve them using traditional numerical methods such as finite difference methods and finite

Email addresses: `wangdong@cuhk.edu.cn` (Dong Wang), `xmxu@lsec.cc.ac.cn.` (Xianmin Xu), `boyizou@link.cuhk.edu.cn` (Boyi Zou)

element methods [24]. However, in the context of deep learning, to the best of our knowledge, there has been no exploration of utilizing deep learning based methods to solve problems with the Lavrentiev phenomenon. The most relevant problems are solving PDEs in regions with complex domain or corner singularities, or situations where the solutions of evolving PDEs become singular over time. There is an extensive range of related methods, such as adaptive weight or sampling methods [25, 26, 27, 28, 29], adaptive loss function methods [18], and many others [30, 31, 32].

When we use deep learning methods to solve problems exhibiting the Lavrentiev phenomenon, they struggle to obtain the global minimizer, which behaves similarly to traditional numerical methods. We have also tried some alternative approaches commonly used to deal with such problems but resulted in little improvement (as will be seen in Section 3). The failure of these deep learning methods indicates that the essence of problems exhibiting the Lavrentiev phenomenon differs from other extensively studied cases. Although the universal approximation property of DNNs to continuous functions is well established [33], a gap exists between what can be approximated and what is prone to be approximated, and this gap is the intrinsic difficulty of problems exhibiting the Lavrentiev phenomenon.

To mitigate the gap that exists between function spaces and to obtain better approximations of problems exhibiting the Lavrentiev phenomenon, we propose a novel method in the framework of deep learning. Specifically, we incorporate a guiding term and a scheduling technique into the standard deep learning process. The proposed method effectively mitigates the Lavrentiev phenomenon and achieves promising approximations for such problems. Additionally, we can apply the same method to handle problems without the Lavrentiev phenomenon, yielding consistent results. This once again demonstrates the capabilities of the proposed method. Furthermore, we compare the proposed method with some other commonly used deep learning methods to highlight the differences and capabilities it possesses.

The rest of this paper is organized as follows. In Section 2.1, we briefly introduce the Lavrentiev phenomenon and the difficulties associated with it. Some standard techniques used for solving differential operator related problems with DNNs are reviewed in Section 2.2. The failures of conventional numerical methods and deep learning methods are discussed in Section 3. Section 4.1 presents the framework of the proposed method, and Section 4.2 provides a brief review of some related methods. Numeric examples and comparisons are provided in Section 5. Finally, we draw conclusions and discussions in Section 6.

2. Preliminary

2.1. The Lavrentiev phenomenon

The Lavrentiev phenomenon is a fascinating property observed in certain functionals within the calculus of variations [34]. Suppose $\mathcal{A}$ is an admissible function space and $I(u)$ is a functional defined on $\mathcal{A}$. The Lavrentiev phenomenon refers to the property that

$$\inf_{u\in\mathcal{A}} I(u) < \inf_{u\in\mathcal{A}_1} I(u), \tag{1}$$

where $\mathcal{A}_1$ is a dense subspace of $\mathcal{A}$. The first example was found by Lavrentiev [35] in 1927. Since then, many researchers have considered this problem from different points of view. For a more comprehensive review, we refer to [36, 37] and references therein. Despite its theoretical significance, the Lavrentiev phenomenon causes a major obstacle to numerical approximation to the minimization problems.

To make the above definition clear, we consider the following Manià [38] example,

$$\begin{aligned}
&\min_{u}\ I_1(u) = \int_0^1 (u^3(x)-x)^2(u'(x))^6 dx \\
&s.t.\ u(0)=0,\ u(1)=1.
\end{aligned} \tag{2}$$

It is easy to observe that the analytical solution of this problem is $u_* = x^{\frac{1}{3}}$ which is in $W_D^{1,1}([0,1])$ but not in $W_D^{1,\infty}([0,1])$. Here $W_D^{1,p} := \{u \in W^{1,p} : u(0)=0, u(1)=1\}$. This problem was proposed by Manià [38] in 1934 and exhibits the Lavrentiev phenomenon, i.e.,

$$\inf_{u\in W_D^{1,1}([0,1])} I_1(u) < \inf_{u\in W_D^{1,\infty}([0,1])} I_1(u). \tag{3}$$

The proof can be found in [39]. Due to the existence of the Lavrentiev phenomena, the standard numerical methods such as the conforming finite element method fail to approximate the global minimizer of I_1 in $W_D^{1,1}$. Actually, one can only approximate the minimizer in $W_D^{1,\infty}$ by the standard method since the finite element space is a subspace of $W_D^{1,\infty}$.

To solve variational problems with Lavrentiev phenomena numerically, some specific numerical techniques have been developed(c.f. [40, 41, 42, 24, 43]). For example, an element removal method was proposed to deal with this problem in [41]. However, it needs to estimate the position of the singular point and remove the neighboring elements of it in the subsequent computational process. The auxiliary variable technique was used in [42] to decouple u and u', which does not require prior information about the singularity position. The main disadvantage is the introduction of additional unknowns in the decoupled problem. For more related methods, we refer to [37, 39, 36] and references therein.

2.2. Deep Ritz method

Using deep neural networks to solve problems related to differential operators has gained significant attention in recent years [3, 1, 44]. Since the problems we considered are of variational form, the deep Ritz method (DRM) [3] is the most natural choice and will be described below.

The general form of problems interested in [3] is as follows,

$$\min_{u\in\mathcal{H}} I_h(u) := \int_\Omega L(u)dx, \tag{4}$$

where $\mathcal{H}$ is the set of admissible functions represented by a neural network. For example, a ResNet with N_d hidden layer and width N_w is constructed to represent the admissible function space H as follows:

$$\begin{aligned}
&u(x_0;\theta) = \mathbf{W}_{out}x_{N_d+1} + \mathbf{b}_{out}, \\
&x_{k+1} = f_k(x_k) = \Phi_k(\mathbf{W}_k x_k + \mathbf{b}_k) + x_k, \quad k = N_d, ..., 1, \\
&x_1 = \mathbf{W}_{in}x_0 + \mathbf{b}_{in},
\end{aligned} \tag{5}$$

here $x_0 \in \Omega$ represents the input of dimension d. The weight matrices of the input layer, the k-th hidden layer, and the output layer are denoted by $\{\mathbf{W}_{in} \in \mathbb{R}^{N_w\times d}, \mathbf{W}_k \in \mathbb{R}^{N_w\times N_w}, \mathbf{W}_{out} \in \mathbb{R}^{n\times N_w}\}$, respectively. The corresponding biases are $\{\mathbf{b}_{in} \in \mathbb{R}^{N_w}, \mathbf{b}_k \in \mathbb{R}^{N_w}, \mathbf{b}_{out} \in \mathbb{R}^{n}\}$. Φ_k represents the activation function. The integration in I is computed approximately by the Monte Carlo (MC) method, and then the optimization process is performed using `Adam` [45]. The numeric results presented in this paper are obtained using this ResNet structure in combination with the activation function $\tanh^3(\cdot)$ (unless otherwise stated), which has been used and verified for its effectiveness in [7]. In many problems, one may have to deal with the Dirichlet boundary condition. This is usually done by a penalty method[3], a Nitsche type method [44] or some direct method [46] as described below.

3. Deep Ritz methods for the Lavrentiev phenomenon.

In this section, we demonstrate that the direct application of the DRM to variational problems exhibiting the Lavrentiev phenomenon fails to capture global minimizers, similar to standard numerical methods. For clarity, we use Manià's problem (2) as an example to illustrate the main challenges.

To handle the Dirichlet boundary condition, we enforce it precisely by multiplying the output of the network by a bubble function that vanishes on the boundary and then adding x, which is a natural choice satisfying the given Dirichlet boundary condition, *i.e.*, $u(x;\theta) = x(1-x)\tilde{u}(x;\theta) + x$. (We use the technique for all the numerical examples in this paper.) The optimization problem is given by

$$\min_{\tilde{u}\in\mathcal{H}} I_1(\tilde{u}) := \int_0^1 \left((x(1-x)\tilde{u}+x)^3 - x\right)^2 \left((x(1-x)\tilde{u}+x)'\right)^6 dx, \tag{6}$$

We approximate $\tilde{u}(x;\theta)$ by a ResNet as described by Equation. (5) with $N_d = 8$ hidden layers and width $N_w = 120$. We set 1000 uniformly random points in the computational domain $\Omega = [0,1]$ to compute the integration by the Monte Carlo method, unless otherwise stated. We test several implementations of the DRM and the numerical results are listed below.

Test for the standard DRM. As shown in Figure 1, under two different initializations of the network(blue curves), the corresponding training results(red curves) fail to approximate the global minimizer $u_*(x) = x^{\frac{1}{3}}$(black curves). Instead, the results are around $u(x) = x$. More specifically, Figure 1 (a) shows the training result under a random initialization, while Figure 1 (b) shows the result under an initialization close to the global minimizer $u_*(x) = x^{\frac{1}{3}}$. The latter case shows the essential difficulty of the deep learning method in dealing with the variational problem with Lavrentiev phenomena. Even if the initial state is close to the global minimizer, the training result can only approximate the minimizer in $W^{1,\infty}$.

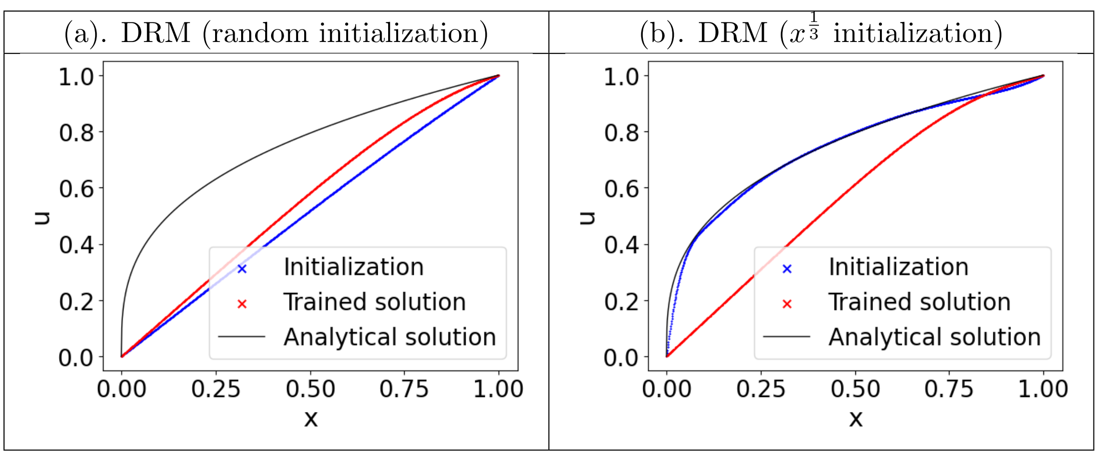

Figure 1: Training results of DRM on the Manià's example with different initializations. See Section 3.

Test for the DRM with non-uniform sampling. One naive idea to approximate well the global minimizer $u_* = x^{\frac{1}{3}}$ is to use non-uniform sampling method. Since the minimizer is singular at the left boundary $x = 0$, we add more points in its vicinity. Specifically, we add 1000 uniform distributed points within $[0, 0.25]$ in addition to the 1000 random points in the whole domain $\Omega = [0,1]$. The numerical result is shown in Figure 2 (a). We can see that adding more sampling points near 0 does not improve the results. Instead, it flattens the curve in the interval $[0, 0.25]$, resulting in a worse approximation.

Another idea to improve the DRM is to use the techniques from the numerical methods developed in the literature. For example. we can introduce an auxiliary variable as in [42] or use an element removal technique as in [41]. We will test these techniques in the following two experiments.

Test for the DRM for a model with auxiliary variable. In this test, we introduce an auxiliary variable v such that $v \approx u'$ in the auxiliary variable method, resulting in the following loss functional,

$$\tilde{I}_1(u, v) = \int_0^1 (u^3(x) - x)^2 v^6(x) dx + \lambda \int_0^1 (u'(x) - v(x))^2 dx, \tag{7}$$

where λ is the penalty weight and $u = x(1-x)\tilde{u} + x$. In the test, the unknown functions $\tilde{u}(x, \theta)$ and $v(x, \theta)$ are the output of one neural network, whose output is a 2-dimensional vector function, with the first component approximating $\tilde{u}$ and the second approximating v. Numerical experiments show that the method still fails to produce satisfactory results and leads to similar training results around $u = x$, as shown in Figure 2 (b).

Test for the DRM with element removal method. Motivated by the element removal method [41], we sample on $[x_0, 1]$ instead of on the whole domain $[0, 1]$ in the DRM, resulting in the following loss functional,

$$\hat{I}_1(\tilde{u}) = \int_{x_0}^1 \big((x(1-x)\tilde{u} + x)^3 - x\big)^2 \big((x(1-x)\tilde{u} + x)'\big)^6 dx. \tag{8}$$

Numerical experiments show that the results are better than the other techniques but still sensitive to the choice of x_0. We show one result in Figure 2 (c), where we set $x_0 = 0.01$. We can see that the numerical results agree well with the global minimizer in an interval $[0.2, 1]$ while there is a larger error in the interval $(0.01, 0.2]$.

Test for the DRM with special activation function. Finally, we test the effect of the activation function. Since the global minimizer is $u_* = x^{\frac{1}{3}}$, we consider $f(x) = x^{\frac{1}{3}}$ as an activation function. Note that $x^{\frac{1}{3}}$ is not a reasonable activation function in general, but we use it here to test if the DRM approximates well to the singular global minimizer using a singular activation function. However, as shown in Figure 2 (d), the training result of DRM with the activation function $x^{\frac{1}{3}}$ with a random initialization is close to $u = x$. This once again highlights the intrinsic difficulty of this problem.

For the failure of these commonly used deep learning methods on problems exhibiting the Lavrentiev phenomenon, there are some possible reasons as presented in [15]. Some commonly used numerical techniques in deep learning, such as initialization schemes, over-parameterized models, stochastic gradient descent (SGD), and others, result in the approximation function having a lower geometric complexity (GC) as defined in [15]

$$< u_\theta, D >_G = \frac{1}{|D|} \sum_{x \in D} \|\nabla_x u_\theta(x)\|_F^2. \tag{9}$$

Here, u_θ represents the training results of the neural network, D is a dataset in the computational domain, and $\|\cdot\|_F$ means the Frobenius norm of a matrix. This GC index is also related to the Lipschitz constant and some other regularity indexes [15]. For more details, we refer to [15, 47] and the references therein. Hence, these commonly used deep learning techniques tend to make the neural network solutions smoother, thus making it inherently difficult to use neural networks to solve such problems. As shown by the numerical experiments using the Manià's example, the deep learning methods tend to learn the minimizer $u = x$ in $W_D^{1,\infty}$ which is similar to the traditional numerical methods, although the global minimizer $u = x^{\frac{1}{3}}$ is a continuous function, which can be approximated well by neural network by the universal approximation theorem [33]. In the following, we propose a new knowledge-guided learning (KGL) method for the deep neural network to approximate the global minimizer of such problems.

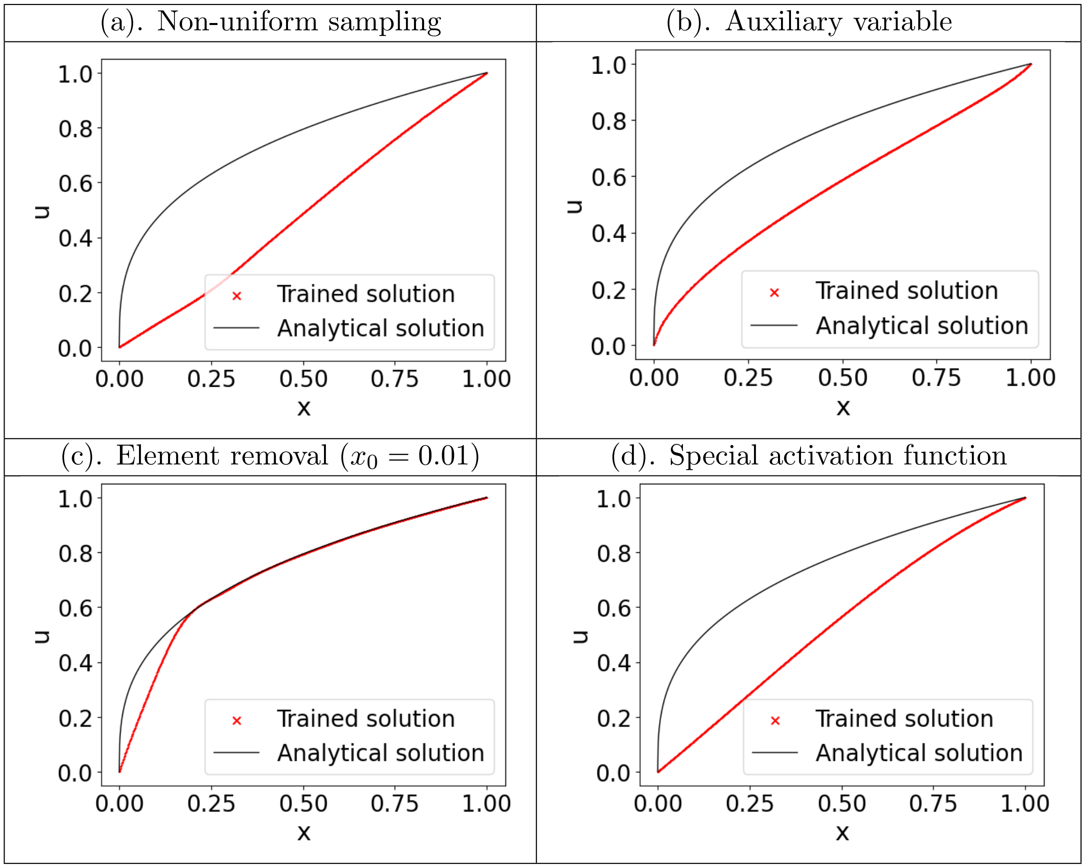

Figure 2: Training results of some adaptive deep learning methods on the Manià's example. See Section 3.

4. Knowledge-guided learning algorithm for Lavrentiev phenomenon

4.1. The algorithm

Consider a variational problem

$$\inf_{u\in\mathcal{A}} I(u) \tag{10}$$

with Lavrentiev phenomena as shown in Equation (1), with its corresponding neural network formulation

$$\min_{\theta\in\Theta} I(u(x;\theta)) \tag{11}$$

where Θ represents the parameter space of the neural network. The tests in the previous section show that direct application of the DRM can only detect the local minimizer in the dense subspace $\mathcal{A}_1$. To overcome the limitation of conventional deep learning methods, we propose the following knowledge-guided learning method to deal with problems exhibiting the Lavrentiev phenomenon.

Specifically, we integrate a guiding term G related to the decision variable into the standard process of training neural networks. The guiding term G will lead a function u in $\mathcal{A}_1$ approaches to functions in $\mathcal{A}\backslash\mathcal{A}_1$. For instance, in the Manià example (2), $\mathcal{A} = W_D^{1,1}([0,1])$ and $\mathcal{A}_1 = W_D^{1,\infty}([0,1])$, where $W_D^{1,p} := \{u \in W^{1,p} : u(0) = 0, u(1) = 1\}$. In this case, G is defined as $-\|u'\|_\infty$, and as G decreases, it encourages u to approach the boundary of the function space $W_D^{1,\infty}([0,1])$. To achieve this, we propose an iterative approach that alternates between optimizing the objective functional and updating the guiding term values toward expected directions. By incorporating this guiding term, it is expected that the training process will yield higher accuracy, particularly for problems affected by the Lavrentiev phenomenon.

The specific algorithm is given in Algorithm 1, and the guiding term will be specified in the corresponding numerical examples in Section 5. In the algorithm, N_d and N_w denote the number of hidden layers and the width of the neural network; N represents the number of training steps; N_u, N_g, τ_u and τ_g represent the scheduling numbers for the original loss functional and the guiding term and their corresponding learning rates, respectively.

Algorithm 1: Knowledge-guided learning.

Inputs:
N_d, N_w: number of hidden layers and width of the neural network representing the decision variable $u = u(x;\theta)$.
N: number of training steps.
N_u, N_g: scheduling numbers of the original functional and the guiding term updates per iteration step.
τ_u, τ_g: learning rates of the objective functional and the guiding term.
for $i = 1 : N$ **do**
    for $j = 1 : N_u$ **do**
        Update $\theta \leftarrow \theta - \tau_u \nabla_\theta I$.
    end
    for $k = 1 : N_g$ **do**
        Update $\theta \leftarrow \theta - \tau_g \nabla_\theta G$.
    end
end
Outputs: Neural network $u(x;\theta)$ with arbitrary input x and trained parameters θ.

4.2. Related methods

To the best of our knowledge, there are currently no deep learning methods that have been researched for problems exhibiting the Lavrentiev phenomenon. Therefore, we will only review some relevant methods in this section, rather than relevant literature.

The training process of the proposed knowledge-guided learning is similar to alternating direction optimization methods, with the difference that it does not involve optimizing a single objective functional under constraints with multiple variables. Instead, it focuses on separately optimizing multiple objective functionals (the original objective functional $I(u)$ and the guiding term $G(u)$), which aligns more closely with auxiliary-task learning [48] and multi-task learning [48, 49].

Auxiliary-task learning improves the performance of the main task through one or more auxiliary tasks, while multi-task learning equally considers enhancing the performance of all tasks. Specifically, in auxiliary-task learning, there is a method similar to the one proposed in this paper, called 'Hints' [48], which can be traced back to the 1990s [50, 51]. Systematic use of hints(auxiliary information) in the learning-from-examples paradigm was presented in [51], where several relevant and insightful examples were provided. It also presented some related optimization algorithms, including the weighted method and the scheduling technique. The weighted rotation scheduling, as reflected in the settings of N_u and N_g in Algorithm 1, is what we used in this paper.

The distinction between learning from hints and this work lies in the construction and demands of the auxiliary information. In [51], the requirement is to convert the hint into a non-negative quantity (or quantity bounded from below) and then minimize it to provide information regarding the main task. However, the guiding term we consider here can be unbounded from below (as demonstrated in the subsequent section, we set the guiding term as $-\|\nabla u\|_\infty$), indicating that this guiding term offers more essential information, for example, the regularity of the solution.

This is also the distinction between the proposed method and the penalty method that uses $\lambda G(u)$ as a penalty term resulting in the following loss functional,

$$Loss = I(u) + \lambda G(u). \tag{12}$$

In the next section, we will show that the proposed knowledge-guided learning can handle problems with or without the Lavrentiev phenomenon by employing the same guiding term. In other words, the proposed method does not alter the landscape of the objective functional, but rather serves as a guideline to the original problem. When the guiding term aligns well with the original objective, the results improve after guidance. However, if the guideline is inconsistent with the original objective functional, the neural network can still find the minimizer of the original problem, which is the main difference between the proposed method and the penalty method. Numerical examples in the next section show that the penalty method fails to deal with both cases.

5. Numerical results

In this section, we use the knowledge-guided learning method to deal with problems exhibiting the Lavrentiv phenomenon to demonstrate its capability. In Section 5.1, we show that the proposed method works well for the Manià's example. Section 5.2 investigates the use of knowledge-guided learning to solve problems without the Lavrentiv phenomenon, emphasizing the difference between the proposed method and the penalty method. Additionally, some 2-dimensional problems are considered in Section 5.3 and 5.4 to showcase the capability of the proposed method. The code is available at https://github.com/BoyiZou/Knowledge-Guided-learning.

5.1. Manià's example

Considering the Manià's example as given in Equation (2), we use the knowledge-guided learning method (Algorithm 1) with the guiding term $G(u) = -\|u'\|_\infty$ to solve the problem. This guiding term aims to induce a derivative blow-up in the training solution. As $G(u)$ decreases, the infinity norm of the derivative of the training results increases, allowing the training result to ideally escape the space of $W^{1,\infty}[0, 1]$. However, it is important to note that this is still a numerical approximation, and as a result, the final training results still belong to $W^{1,\infty}[0, 1]$. Nevertheless, this approach provides a more accurate and instructive approximation to Manià's example as shown in Figure 3 (a) and (b).

We compare the results of the knowledge-guided learning method and the penalty method (12) with the same network settings as used in Section 3, and set $N_u = 4$, $N_g = 1$ in knowledge-guided learning. As shown in Figure 3, the first row presents the training result of the proposed method along with the corresponding point-wise error. It is observed that with the proposed method, the neural network can effectively approximate the global minimizer beyond a small region surrounding $x = 0$. It is worth noting that the guiding term used here does not require specific information about the exact points of blow-up. We only need to provide the neural network with a guideline that there may be cases of derivative blow-up in the solution, and the network can learn to approximate the solution effectively.

For the penalty method, incorporating the penalty term $-\lambda\|u'\|_\infty$, the loss functional is defined as follows,

$$\min_u \int_0^1 \left(u^3 - x\right)^2 \left(u'\right)^6 dx - \lambda\|u'\|_\infty. \tag{13}$$

Figure 3 (c) and (d) display the results of the penalty method, which represent the best performance we obtained through parameter tuning. These two figures depict the training results at different iteration steps of the same training process. Figure 3 (c) represents the closest approximation obtained during the training process, while Figure 3 (d) illustrates the final converged result. To be more specific, we plot the changes of the loss functional and the infinity norm of the error during the training process corresponding to the penalty method in Figure 3 (e) and (f). It can be observed that during the training process, the loss functional decreases, while the infinity norm error initially decreases and then increases, which corresponds to the solution exhibiting a tendency to approximate the analytic solution in the initial stage, but failing to stabilize or converge. The result converges to a solution where the derivative at the origin becomes sufficiently large, while away from the origin, it approximates a straight line, significantly deviating from the analytic solution. This is due to the fact that the penalty term affects the landscape of the original objective functional and leads to a change in the minimizer. Moreover, the minimizer and convergence process heavily depend on the magnitude of the penalty terms and the original functional.

More importantly, the penalty method fails to train results with regular solutions, as will be demonstrated in the next section. To gain a better intuitive understanding, Figure 5 presents a schematic diagram to show the distinction between the proposed method and the penalty method for problems with and without the Lavrentiev phenomenon, which will be further explained in Section 5.2.

To better understand how the knowledge-guided learning method works, we define the local loss as $Loss_{[0.005,1]} = \int_{0.005}^1 (u^3(x) - x)^2 (u'(x))^6 dx$ to demonstrate more intrinsic results. As shown in Table 1, the local loss decreases to a small range while the global loss remains around 0.1 for the knowledge-guided learning method. However, both the global loss and local loss of the results obtained using standard DRM, are approximately 0.0666. This also illustrates why using standard DRM can only yield the result around $u(x) = x$, as the global loss is approximately 0.0666, which is less than the global loss computed by the knowledge-guided learning method.

Number of training points\\$Loss$	Global Loss, $Loss_{[0,1]}$	Local Loss, $Loss_{[0.005,1]}$
KGL: 200	0.09697798	0.01304583
KGL: 1000	0.09206629	0.00051850
KGL: 2000	0.11957884	0.00000224
DRM: 200	0.06659865	0.06669694
DRM: 1000	0.06659041	0.06670607
DRM: 2000	0.06658889	0.06670415

Table 1: Comparisons correspond to the relationship between the number of training points and the global and local $Loss$ of knowledge-guided learning and penalty method. See Section 5.1.

However, using the proposed method, we give the neural network a guideline that the result of this problem may not be in $W^{1,\infty}$, enabling it to approximate the global minimizer efficiently. More

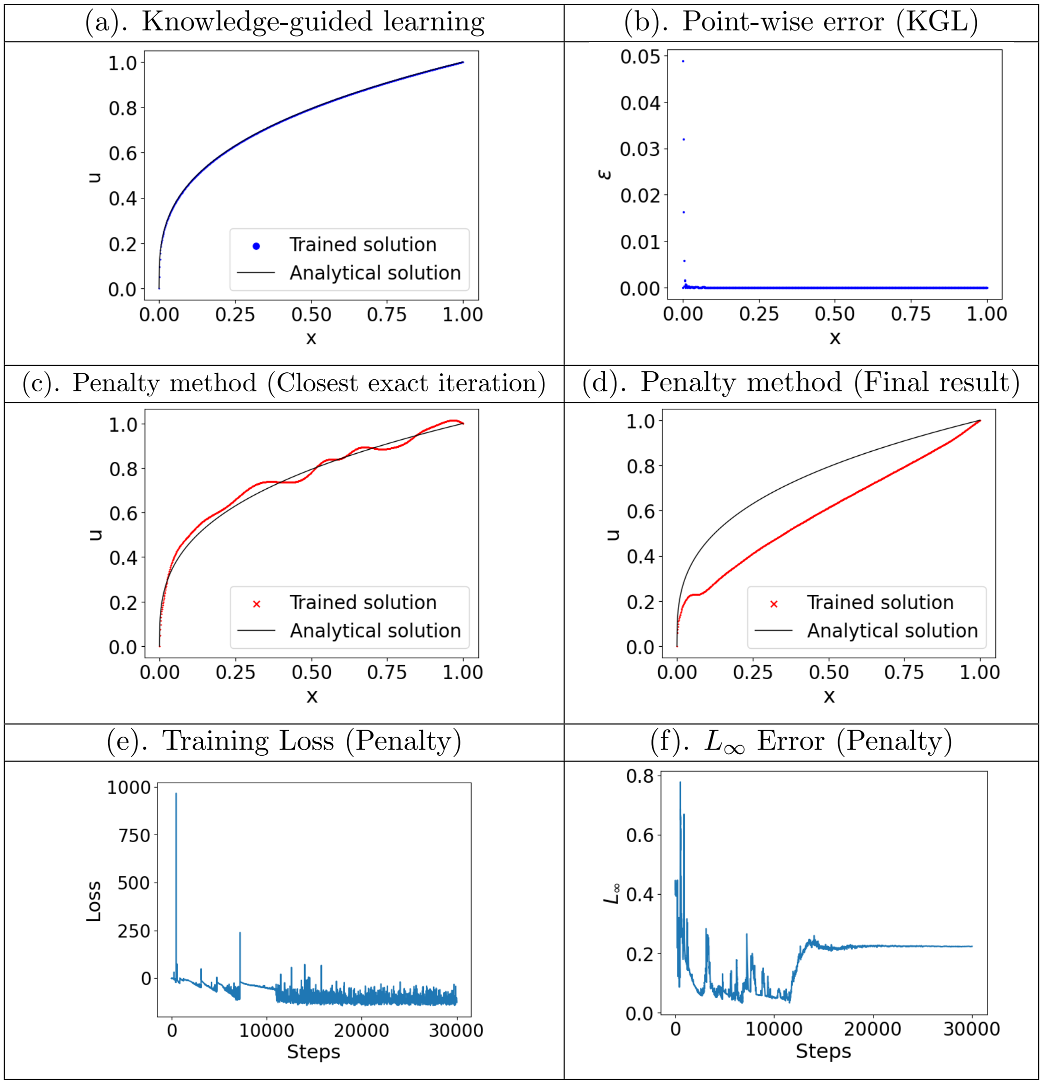

Figure 3: Comparisons of the results on Manià's example of knowledge-guided learning and penalty method. See Section 5.1.

importantly, while the guiding term is inconsistent with the original problem, the neural network can disregard this guiding term and obtain the original minimizer. This will be demonstrated in the next section.

5.2. Problems without the Lavrentiev phenomenon

In this section, we use the knowledge-guided learning method to solve a variational problem without the Lavrentiev phenomenon. Surprisingly, the proposed method still works well even when the solution is regular. This demonstrates the robustness of the method. To make the statement clear, we also compare the proposed method with the penalty method.

We consider the following variational problem with $u(x) = 2x - x^2$ as the global minimizer,

$$\begin{aligned} &\min_u \int_0^1 \frac{1}{2}|u'(x)|^2 - 2u(x)dx \\ &s.t.\ u(0) = 0,\ u(1) = 1. \end{aligned} \tag{14}$$

We solve this problem by the knowledge-guided learning method with the same guiding term $G(u) = -\|u'\|_\infty$. The guiding term is inconsistent with the analytic solution in this example. However, as shown in Figure 4 (a) and (b), the proposed method can still solve the problem correctly, albeit with more computational effort. It is noteworthy that we use the same parameters as in the previous Manià's example, demonstrating the robustness of the proposed method.

We then contrast the proposed method to the penalty method, as shown in Figure 4 (c) and (d), the results obtained by the penalty method deviate from the global minimizer due to the penalty term. To be more specific, the proposed method considers $G(u)$ merely as a guiding term rather than an exact constraint term, unlike the penalty method. Therefore, during the optimization process, the proposed method explores the approximation of the original objective functional under a guideline, whereas the penalty method modifies the landscape of the objective functional through the penalty term and approximates the minimizer of the new system.

For a more intuitive understanding of the proposed method, we illustrate the training process of knowledge-guided learning as a diagram in Figure 5 based on the numerical results obtained in Section 5.1 and Section 5.2, and compare it with the penalty method.

For Manià's example, the analytical solution $u = x^{\frac{1}{3}}$, a continuous function, falls within the approximation capabilities of DNNs. However, as shown in Section 3, deep learning based numerical methods struggle to approximate the global minimizer of this problem. Therefore, we conclude that the global minimizer lies outside the easy-to-fit space of DNNs under this loss functional, and we consider using knowledge-guided learning to improve the training results with $G(u) = -\|u'\|_\infty$. As shown in Figure 5 (a), the guiding term indicates that the derivative of the training result could have a large infinity norm, which is a difficult feature to learn under Manià's example, leading to a better approximation during the alternative optimization process. In contrast, we illustrate the training process of the penalty method, as shown in Figure 5 (c). With the penalty term $\lambda G(u) = -\lambda\|u'\|_\infty$, the easy-to-fit space is altered (or one might say that the landscape of the modified loss functional is changed). However, it still struggles to achieve a good approximation, and the approximation space after penalization, along with the corresponding training result, heavily relies on the penalty weight λ.

As for problems without the Lavrentiev phenomenon demonstrated in Section 5.2, the corresponding training diagrams are depicted in Figure 5 (b) and (d). In this regular case, the analytical solution falls in the easy-to-fit space of the original loss functional. We utilize the same guiding and penalty terms as in the previous example to showcase the differences. As shown in Figure 5 (b), the training result can still achieve a good approximation, albeit with a more oscillatory training process due to the improper guiding term. In contrast, as shown in Figure 5 (d), with the penalty method, the approximation space is modified after penalization, rendering the originally easy-to-fit analytical solution less readily trainable. Based on these observations, the proposed method can handle problems exhibiting the Lavrentiev phenomenon and may have the potential to assess the regularity of some unknown systems.

5.3. Two-dimensional extension of Manià's example.

In this section, we consider the following variational problem in the unit square $\Omega = [0, 1]^2$, which can be viewed as a constant extension along the y-axis of Manià's example,

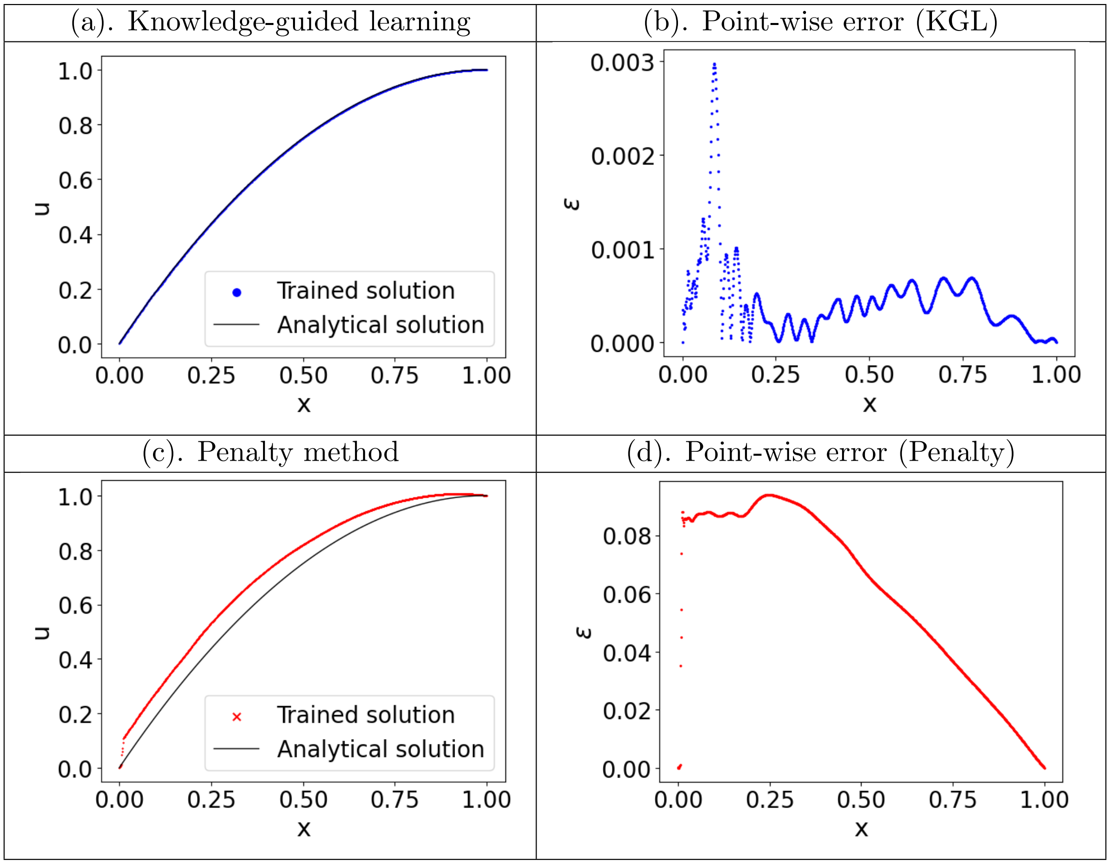

Figure 4: Comparisons of the results on problem (14) of knowledge-guided learning and penalty method. See Section 5.2.

$$
\begin{aligned}
&\min_u \int_0^1 \int_0^1 (u^3(x) - x)^2 (u_x(x))^6 dx dy \\
&s.t.\ u(0, y) = 0,\ u(1, y) = 1.
\end{aligned}
\tag{15}
$$

The main difference between problems (2) and (15) is that the singularity here is a line segment $\{(x, y)|x = 0, y \in [0, 1]\}$ instead of one single point. For this example, we use $G(u) = -\|\nabla u\|_\infty$ as the guiding term to demonstrate the capability of the proposed method. It is worth noting that in this case, we do not need to specify $G(u) = -\|u_x\|_\infty$ or prioritize certain line configurations, but rather keep it simple as before. We compare the training results with those of the proposed method, vanilla DRM, and the penalty method with $\lambda G(u) = -\lambda\|u_x\|_\infty$ as the penalty term. The networks are set as $N_d = 4$ and $N_w = 50$, with 10000 training points for all methods, and set $N_u = 4$, $N_g = 1$ in knowledge-guided learning.

As shown in Figure 6, the training results of knowledge-guided learning are consistent well with the analytical solution while other methods fail to get a good approximation. More differences can be seen in Figure 6 (b) and (c). The proposed method obtains a solution that is constantly extended along the y-axis, meaning it does not affect the original loss functional landscape. However, the penalty method is different as it affects the landscape of the original loss functional, resulting in an oscillating, non-constant solution along the y-axis. This further demonstrates the capability of the proposed method and corroborates the intuitive explanations provided in Figure 5.

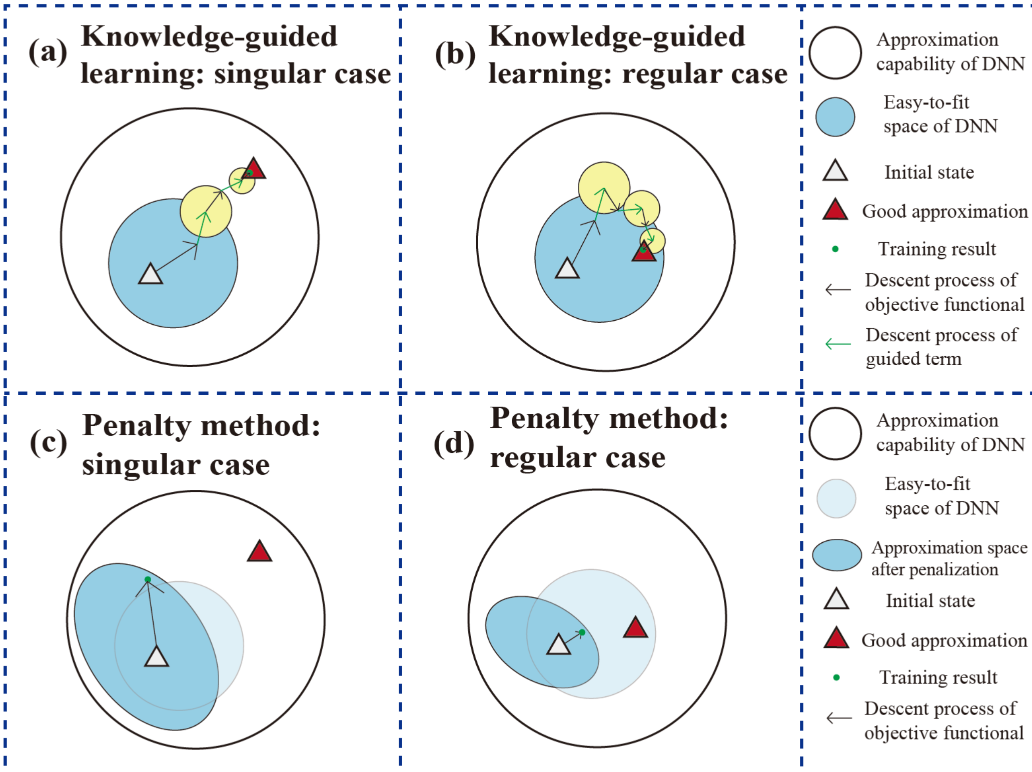

Figure 5: Schematic diagram of knowledge-guided learning and penalty method on problems (2) and (14). See Section 5.1 and 5.2.

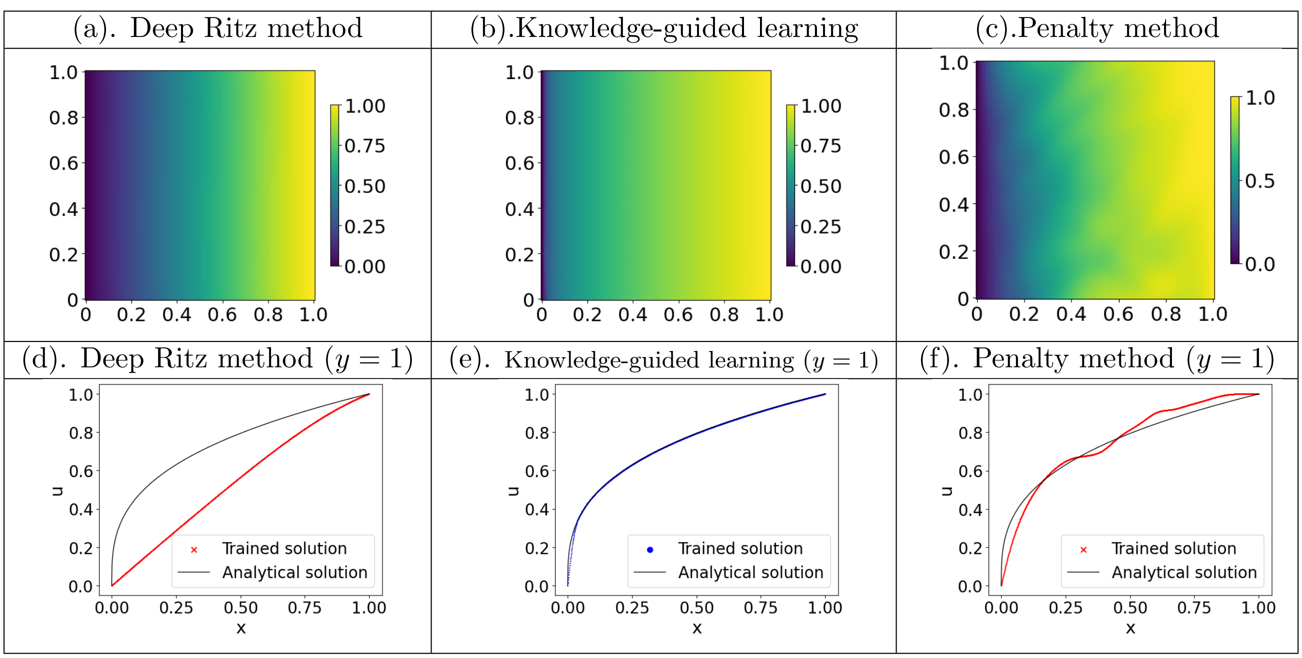

Figure 6: Training results for problem (15). The first row displays the training results, while the second row displays a slice of the training results alongside the analytical solution at $y = 1$. See Section 5.3

5.4. Foss's example

In this section, we consider the Foss's example [52] in 2-dimensional space which also exhibits the Lavrentiev phenomenon,

$$\begin{aligned} \min_{u}\ I(u) = 66(\frac{13}{14})^{14}\int_{\Omega}(\frac{y}{y-1})^{14}|u|^{\frac{14-3y}{y-1}}(|u|^{\frac{y}{y-1}}-x)^{2}(u_x)^{14}dxdy \\ s.t.\ u(0,\cdot)=0, \text{ and } u(1,\cdot)=1, \end{aligned} \tag{16}$$

where $\Omega = [0,1]\times[1.5,2.5]$. The analytic solution of this problem is $u(x,y) = x^{\frac{y-1}{y}}$. It was shown by Foss [52] that

$$0 = \inf_{u\in\mathbb{A}} I(u) < \inf_{u\in\mathbb{A}_\infty} I(u) = 1, \tag{17}$$

where $\mathbb{A} = \{u \in W^{1,1}(\Omega) : u(0,\cdot) = 0, \text{ and } u(1,\cdot) = 1\}$ and $\mathbb{A}_\infty = \{u \in W^{1,\infty}(\Omega) : u(0,\cdot) = 0, \text{ and } u(1,\cdot) = 1\}$. The singularity of this problem is around $\{(x,y)|x = 0, y \in [1.5, 2.5]\}$ with u_x blowing up. Similar to the previous example, we use the guiding term $G = -\|\nabla u\|_\infty$ and the penalty term $\lambda G(u) = -\lambda\|\nabla u\|_\infty$ to train the networks. The networks are set as $N_d = 4$ and $N_w = 50$, with 10000 training points for all methods, and set $N_u = 1$, $N_g = 1$ in knowledge-guided learning.

As shown in Figure 7 (a) and (d), the training results of DRM demonstrate that the main difficulty of Foss's example is around $(0, 1.5)$ with the infinity norm of the error around 0.075. As for the results of knowledge-guided learning and penalty method, as shown in Figure 7 (b), (c), (e), and (f), knowledge-guided learning gets a better approximation than DRM while the penalty method fails. For a more intuitive comparison, we provide some 1-dimensional cross-section results, including the bottom line ($y = 1.5$), and the diagonal line $y = x+1$, between DRM and knowledge-guided learning. As shown in Figure 7 (g), and (h), knowledge-guided learning improves the approximation around $(0, 1.5)$, with the infinity norm of the error around 0.04, while maintaining similar accuracy in other regions compared to the results of DRM.

6. Summary

In this paper, we use the deep learning methods to detect singular minimizers for a class of variational problems exhibiting the Lavrentiev phenomenon. We have demonstrated that the standard deep Ritz method and some of its variations fail to find the global minimizers of these problems. We introduce a knowledge-guided learning algorithm to explore the singular solutions resulting in better approximation. Numerical experiments show that the method works well for both regular and singular variational problems. The work also highlights a gap between approximation of a function and approximation of a solution of a variational problem. In the later case, the mathematical property of the variational problem and the optimization procedure play more significant roles. We will study more general methods and the corresponding mathematical theory in the machining learning approaches for singular problems in the future.

Acknowledgements

D. Wang was partially supported by National Natural Science Foundation of China (Grant No. 12101524, 12422116), Guangdong Basic and Applied Basic Research Foundation (Grant No.

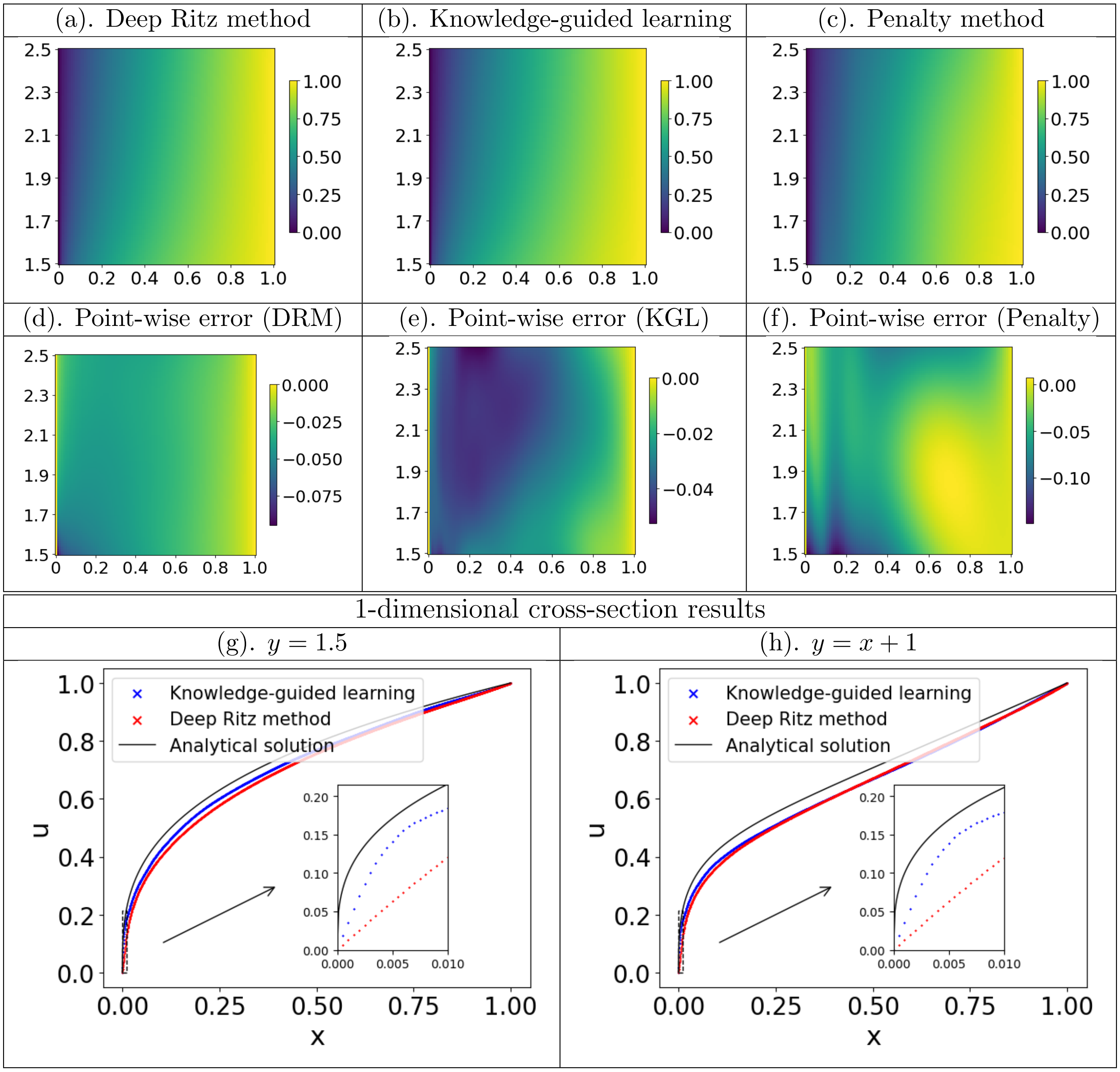

Figure 7: Comparisons of the results for problem (16) among knowledge-guided learning, penalty method, and vanilla DRM. See Section 5.4

2023A1515012199), Shenzhen Science and Technology Innovation Program (Grant No. JCYJ20220530143803007, RCYX20221008092843046), Guangdong Provincial Key Laboratory of Mathematical Foundations for Artificial Intelligence (2023B1212010001), and Hetao Shenzhen-Hong Kong Science and Technology Innovation Cooperation Zone Project (No.HZQSWS-KCCYB-2024016). X. Xu was partially supported by National Natural Science Foundation of China (Grant No. 12371415, 12461160275), and by the Beijing Natural Science Foundation (Grant No. Z240001)

Data availability

Enquiries about data availability should be directed to the authors.

Declarations

Conflict of interest We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work.

References

[1] M. Raissi, P. Perdikaris, G. 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. `doi:10.1016/j.jcp.2018.10.045`.

[2] J. Sirignano, K. Spiliopoulos, DGM: A deep learning algorithm for solving partial differential equations, Journal of Computational Physics 375 (2018) 1339–1364. `doi:10.1016/j.jcp.2018.08.029`.

[3] W. E, B. Yu, The deep Ritz method: A deep learning-based numerical algorithm for solving variational problems, Communications in Mathematics and Statistics 6 (1) (2018) 1–12. `doi:10.1007/s40304-018-0127-z`.

[4] Y. Zang, G. Bao, X. Ye, H. Zhou, Weak adversarial networks for high-dimensional partial differential equations, Journal of Computational Physics 411 (2020) 109409. `doi:10.1016/j.jcp.2020.109409`.

[5] G. Bao, X. Ye, Y. Zang, H. Zhou, Numerical solution of inverse problems by weak adversarial networks, Inverse Problems 36 (11) (2020) 115003. `doi:10.1088/1361-6420/abb447`.

[6] Y. Zang, G. Bao, ParticleWNN: a novel neural networks framework for solving partial differential equations, arXiv:2305.12433 (2023).

[7] G. Bao, D. Wang, B. Zou, WANCO: Weak adversarial networks for constrained optimization problems, arXiv:2407.03647 (2024).

[8] Y. Li, Z. Lin, Y. Wang, H. Xie, Tensor neural network interpolation and its applications, arXiv:2404.07805 (2024).

[9] Y. Wang, H. Xie, Computing multi-eigenpairs of high-dimensional eigenvalue problems using tensor neural networks, Journal of Computational Physics 506 (2024) 112928. `doi:10.1016/j.jcp.2024.112928`.

[10] 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. `doi:10.1016/j.cma.2021.114129`.

[11] J. Chen, X. Chi, W. E, Z. Yang, Bridging traditional and machine learning-based algorithms for solving PDEs: The random feature method, Journal of Machine Learning 1 (3) (2022) 268–298. `doi:10.4208/jml.220726`.

[12] K. Linka, A. Schäfer, X. Meng, Z. Zou, G. E. Karniadakis, E. Kuhl, Bayesian physics informed neural networks for real-world nonlinear dynamical systems, Computer Methods in Applied Mechanics and Engineering 402 (2022) 115346. `doi:10.1016/j.cma.2022.115346`.

[13] L. McClenny, U. Braga-Neto, Self-adaptive physics-informed neural networks, SSRN Electronic Journal (2022). doi:10.2139/ssrn.4086448.

[14] J. Yu, L. Lu, X. Meng, G. E. Karniadakis, Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems, Computer Methods in Applied Mechanics and Engineering 393 (2022) 114823. doi:10.1016/j.cma.2022.114823.

[15] B. Dherin, M. Munn, M. Rosca, D. Barrett, Why neural networks find simple solutions: The many regularizers of geometric complexity, in: Advances in Neural Information Processing Systems, Vol. 35, Curran Associates, Inc., 2022, pp. 2333–2349.

[16] H. Wang, B. Zou, J. Su, D. Wang, X. Xu, Variational methods and deep Ritz method for active elastic solids, Soft Matter 18 (32) (2022) 6015–6031. doi:10.1039/d2sm00404f.

[17] Z. Li, B. Zou, H. Wang, J. Su, D. Wang, X. Xu, Deep learning-based computational method for soft matter dynamics: Deep Onsager-Machlup method, arXiv:2308.14513 (2023).

[18] S. Wang, S. Sankaran, P. Perdikaris, Respecting causality for training physics-informed neural networks, Computer Methods in Applied Mechanics and Engineering 421 (2024) 116813. doi:10.1016/j.cma.2024.116813.

[19] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, M. Tegmark, KAN: Kolmogorov-Arnold networks, arXiv:2404.19756 (2024).

[20] 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. doi:10.1137/19m1274067.

[21] 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. doi:10.1038/s42254-021-00314-5.

[22] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, F. Piccialli, Scientific machine learning through physics–informed neural networks: Where we are and what's next, Journal of Scientific Computing 92 (3) (2022). doi:10.1007/s10915-022-01939-z.

[23] S. A. Faroughi, N. M. Pawar, C. Fernandes, M. Raissi, S. Das, N. K. Kalantari, S. Kourosh Mahjour, Physics-guided, physics-informed, and physics-encoded neural networks and operators in scientific computing: Fluid and solid mechanics, Journal of Computing and Information Science in Engineering 24 (4) (2024). doi:10.1115/1.4064449.

[24] C. Carstensen, C. Ortner, Computation of the Lavrentiev phenomenon, OxMoS Preprint (17) (2009).

[25] L. D. McClenny, U. M. Braga-Neto, Self-adaptive physics-informed neural networks, Journal of Computational Physics 474 (2023) 111722. doi:https://doi.org/10.1016/j.jcp.2022.111722.

[26] C. L. Wight, J. Zhao, Solving Allen-Cahn and Cahn-Hilliard equations using the adaptive physics informed neural networks, Communications in Computational Physics 29 (3) (2021) 930–954. doi:10.4208/cicp.oa-2020-0086.

[27] Z. Gao, L. Yan, T. Zhou, Failure-informed adaptive sampling for PINNs, SIAM Journal on Scientific Computing 45 (4) (2023) A1971–A1994. doi:10.1137/22m1527763.

[28] S. Zeng, Z. Zhang, Q. Zou, Adaptive deep neural networks methods for high-dimensional partial differential equations, Journal of Computational Physics 463 (2022) 111232. doi:https://doi.org/10.1016/j.jcp.2022.111232.

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

[30] T. Hu, B. Jin, Z. Zhou, Solving elliptic problems with singular sources using singularity splitting deep Ritz method, SIAM Journal on Scientific Computing 45 (4) (2023) A2043–A2074. doi:10.1137/22m1520840.

[31] T. Hu, B. Jin, Z. Zhou, Solving Poisson problems in polygonal domains with singularity enriched physics informed neural networks, SIAM Journal on Scientific Computing 46 (4) (2024) C369–C398. doi:10.1137/23m1601195.

[32] J. M. Taylor, D. Pardo, J. Muñoz-Matute, Regularity-conforming neural networks (ReCoNNs) for solving partial differential equations, arXiv:2405.14110 (2024).

[33] G. Cybenko, Approximation by superpositions of a sigmoidal function, Mathematics of Control, Signals, and Systems 2 (4) (1989) 303–314. doi:10.1007/bf02551274.

[34] J. M. Ball, Discontinuous equilibrium solutions and cavitation in nonlinear elasticity, Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences 306 (1496) (1982) 557–611. doi:10.1098/rsta.1982.0095.

[35] M. Lavrentieff, Sur quelques problèmes du calcul des variations, Annali di Matematica Pura ed Applicata 4 (1) (1927) 7–28. doi:10.1007/bf02409983.

[36] G. Buttazzo, M. Belloni, A survey on old and recent results about the gap phenomenon in the calculus of variations, Springer Netherlands, 1995, p. 1–27. doi:10.1007/978-94-015-8472-2_1.

[37] R. Cerf, C. Mariconda, The Lavrentiev phenomenon, arXiv:2404.02901 (2024).

[38] B. Manià, Sopra un esempio di Lavrentieff, Zanichelli, 1934.

[39] G. Buttazzo, M. Giaquinta, S. Hildebrandt, One-dimensional variational problems: An introduction, Oxford University PressOxford, 1999. doi:10.1093/oso/9780198504658.001.0001.

[40] J. M. Ball, G. Knowles, A numerical method for detecting singular minimizers, Numerische Mathematik 51 (1987) 181–197.

[41] Z. Li, Element removal method for singular minimizers in variational problems involving Lavrentiev phenomenon, Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences 439 (1905) (1992) 131–137. doi:10.1098/rspa.1992.0138.

[42] G. Knowles, Finite element approximation to singular minimizers, and applications to cavitation in non-linear elasticity, in: Differential Equations and Mathematical Physics: Proceedings of an International Conference held in Birmingham, Alabama, USA, Springer, 2006, pp. 236–247.

[43] X. Xu, D. Henao, An efficient numerical method for cavitation in nonlinear elasticity, Mathematical Models and Methods in Applied Sciences 21 (08) (2011) 1733–1760. doi:10.1142/s0218202511005556.

[44] Y. Liao, P. Ming, Deep Nitsche method: Deep Ritz method with essential boundary conditions, Communications in Computational Physics 29 (5) (2021) 1365–1384. doi:10.4208/cicp.oa-2020-0219.

[45] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv:1412.6980 (2014).

[46] N. Sukumar, A. Srivastava, Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks, Computer Methods in Applied Mechanics and Engineering 389 (2022) 114333. doi:10.1016/j.cma.2021.114333.

[47] C. Beck, M. Hutzenthaler, A. Jentzen, B. Kuckuck, An overview on deep learning-based approximation methods for partial differential equations, Discrete and Continuous Dynamical Systems - B 28 (6) (2023) 3697–3746. doi:10.3934/dcdsb.2022238.

[48] S. Ruder, An overview of multi-task learning in deep neural networks, arXiv:1706.05098 (2017).

[49] M. Crawshaw, Multi-task learning with deep neural networks: A survey, arXiv:2009.09796 (2020).

[50] Y. S. Abu-Mostafa, Learning from hints in neural networks, Journal of Complexity 6 (2) (1990) 192–198. doi:10.1016/0885-064x(90)90006-y.

[51] Y. S. Abu-Mostafa, Hints, Neural Computation 7 (4) (1995) 639–671. doi:10.1162/neco.1995.7.4.639.

[52] M. Foss, Examples of the Lavrentiev phenomenon with continuous Sobolev exponent dependence, Journal of Convex Analysis 10 (2) (2003) 445–464.