Advances in Applied Mathematics and Mechanics
Adv. Appl. Math. Mech., Vol. **15**, No. 6, pp. 1602-1630

DOI: 10.4208/aamm.OA-2022-0006
December 2023

High Order Deep Domain Decomposition Method for Solving High Frequency Interface Problems

Zhipeng Chang[1,2], Ke Li[3], Xiufen Zou[1,*] and Xueshuang Xiang[2,*]

[1] *School of Mathematics and Statistics, Wuhan University, Wuhan, Hubei 430072, China*
[2] *Qian Xuesen Laboratory of Space Technology, China Academy of Space Technology, Beijing 100875, China*
[3] *Information Engineering University, Zhengzhou, Henan 450001, China*

Received 9 January 2022; Accepted (in revised version) 20 October 2022

Abstract. This paper proposes a high order deep domain decomposition method (HOrderDeepDDM) for solving high-frequency interface problems, which combines high order deep neural network (HOrderDNN) with domain decomposition method (DDM). The main idea of HOrderDeepDDM is to divide the computational domain into some sub-domains by DDM, and apply HOrderDNNs to solve the high-frequency problem on each sub-domain. Besides, we consider an adaptive learning rate annealing method to balance the errors inside the sub-domains, on the interface and the boundary during the optimization process. The performance of HOrderDeepDDM is evaluated on high-frequency elliptic and Helmholtz interface problems. The results indicate that: HOrderDeepDDM inherits the ability of DeepDDM to handle discontinuous interface problems and the power of HOrderDNN to approximate high-frequency problems. In detail, HOrderDeepDDMs ($p>1$) could capture the high-frequency information very well. When compared to the deep domain decomposition method (DeepDDM), HOrderDeepDDMs ($p>1$) converge faster and achieve much smaller relative errors with the same number of trainable parameters. For example, when solving the high-frequency interface elliptic problems in Section 3.3.1, the minimum relative errors obtained by HOrderDeepDDMs ($p=9$) are one order of magnitude smaller than that obtained by DeepDDMs when the number of the parameters keeps the same, as shown in Fig. 4.

AMS subject classifications: 35Q68, 65N55, 68T99

Key words: Deep neural network, high order methods, high-frequency interface problems, domain decomposition method.

*Corresponding author.
Emails: xfzou@whu.edu.cn (X. Zou), xiangxueshuang@qxslab.cn (X. Xiang)

1 Introduction

High-frequency interface problems commonly appear in various scientific and engineering applications, such as the high-frequency scalar wave equation and Schrödinger equation with a sharp interface [15]. The traditional numerical methods, such as finite element methods (FEMs) and finite difference methods (FDMs), cannot effectively predict the behavior of the high-frequency waves due to the discontinuities caused by the interface. To solve the interface problems, [17] proposes a domain decomposition method (DDM), named "Schwarz alternating method", which transforms a discontinuous interface problem into some continuous sub-problems and then solves them separately. After the parallel computing capability becomes available, the Schwarz alternating method is further developed, and many efficient methods are proposed, such as the parallel Schwarz method [14], the multiplicative Schwarz method [1,18], and the additive Schwarz method [4]. Traditional numerical methods such as FEMs and FDMs are usually applied for sub-problems in these domain decomposition methods. However, they are mesh dependent, and mesh generation is expensive, especially for complex PDEs, e.g., those with complex interfaces or boundaries. In addition, the highly oscillatory nature of the high-frequency problems also brings significant challenges for finding numerical solutions of high accuracy.

Recently, deep-learning-based numerical methods for solving partial differential equations(PDEs) have received much attention due to their meshless advantage. They are also used in combination with DDM to solve PDEs on complex domains to pursuit greater accuracy and efficiency. One popular way is to replace the sub-domain solvers with deep-learning-based solvers such as physics-informed neural network(PINN) [16] and deep ritz method [22] in the classical overlapping Schwarz approach. Under this framework, [13] proposes a DeepDDM, which leverages PINN to discretize the sub-problems divided by DDM and exchanges the sub-problem information across the interface by adjusting the boundary term in the solution of each sub-problem. [10] proposes a cPINN, where the computational domain is also divided and the flux continuity in the strong form is enforced along the sub-domain interfaces. [9] further proposes a more generalized space-time domain decomposition approach. In fact, the idea of domain decomposition is also used to solve interface problems, where different neural networks are often employed for different sub-domains divided by the interface to deal with the dramatical change across the interface of the solution, and these neural networks are weakly coupled by the interface conditions [6,7]. Besides, other ways of dealing with discontinuous interfaces also exist. For example, the d-dimensional piecewise continuous solution of the interface problem can be continuously extended in $(d+1)$-dimensional space by augmenting a variable, so that the continuous augmented function can be approximated by a shallow neural network efficiently [8,12]. Although these methods have been successful for solving interface problems to a certain extent, challenges still exist when they are applied to high-frequency interface problems. For example, these conventional neural networks preferentially approximate the low-frequency components of the target

function but fail to approximate the high-frequency components [21], which causes bad performances in solving high-frequency interface problems. In order to provide conventional neural networks with stronger abilities to represent high-frequency components, we propose a high order deep neural network in [2], termed HOrderDNN, which incorporate high order idea from FEMs into conventional neural networks. Compared to conventional neural networks, HOrderDNN could characterize high-frequency information very well and converge to smaller errors with faster speed.

In this paper, we combine HOrderDNN with the deep-learning-based DDMs (DeepDDMs) and propose a high order deep domain decomposition method, named "HOrderDeepDDM", to solve high-frequency interface problems, where the computational domain is divided into some sub-domains by DDM and HOrderDNNs are applied to solve the high-frequency PDEs on sub-domains. HOrderDeepDDM inherits the advantages of DeepDDM to handle discontinuous interface problems and the power of HOrderDNN to approximate high-frequency problems. The adaptive learning rate annealing method in [19] is also considered to balance the errors inside the region, on the interface and the boundary during the optimization process, shown in Section 2.3. To test the performance of HOrderDeepDDM, we conduct a series of experiments. HOrderDeepDDM is first applied to solve the low-frequency and high-frequency elliptic interface problems, including the cases of high contrast coefficients and irregular interfaces. Then the high-frequency Helmholtz interface problem is also taken into account. We empirically found that the proposed HOrderDeepDDM has the following advantages:

1. HOrderDeepDDMs ($p>1$) maintain the advantages of DeepDDM and outperform DeepDDM in high-frequency elliptic and Helmholtz interface problems. This is two-fold. On the one hand, HOrderDeepDDMs ($p>1$) converge faster than DeepDDM, and the larger the order p, the faster the relative error decreases. In detail, with the same stop condition, HOrderDeepDDMs ($p>1$) require far fewer iterations than DeepDDM for both the outer and inner iteration of DDM, which can be seen in Fig. 5 and Fig. 6. On the other hand, HOrderDeepDDMs ($p>1$) can achieve minor relative errors with the same parameters of the network, and the error decreases as the order increases. Note that in Fig. 4, the minimum relative errors obtained by HOrderDeepDDM ($p=9$) are at least one order of magnitude smaller than that of DeepDDM.

2. HOrderDeepDDMs ($p>1$) could capture the high-frequency information very well. As we can see in Fig. 7, when solving the high-frequency elliptic interface problem in $[0,2]\times[0,2]$, HOrderDeepDDM ($p=9$) is able to approximate the two-dimensional target function with frequencies up to 40 very well and obtain a relative error of about 1%. This advantage of HOrderDeepDDM is clearly reflected in Fig. 12 and Fig. 17, where HOrderDeepDDMs ($p>1$) can capture the local oscillations in the target function, while DeepDDM cannot.

The layout of this paper is as follows. A detailed description of HOrderDeepDDM is

presented in Section 2. In Section 3, several examples, including high-frequency elliptic and Helmholtz interface problems, are presented to demonstrate the effectiveness of the proposed method. The paper ends with a summary and discussion in Section 4.

2 Method

In this paper, we consider the numerical solution of the high-frequency interface problem with the general form expressed as follows:

$$\mathcal{L}(u) = f \quad \text{in } \Omega, \tag{2.1a}$$
$$\mathcal{B}(u) = g \quad \text{on } \partial\Omega, \tag{2.1b}$$
$$[[u]] = p \quad \text{on } \Gamma, \tag{2.1c}$$
$$[[\partial_n u]] = q \quad \text{on } \Gamma, \tag{2.1d}$$

where Ω, as shown in Fig. 1, can be divided into two sub-domains Ω_1 and Ω_2 by the interface Γ. $\mathcal{L}$ represents the differential operator, $\mathcal{B}$ denotes the boundary conditions, and f, g, p, q are given functions (possilby high-frequency). The notation $[[\cdot]]$ denotes the jump across the interface Γ and is defined as $[[u]] = u_1|_\Gamma - u_2|_\Gamma$, $[[\partial_n u]] = \partial_n u_1|_\Gamma - \partial_n u_2|_\Gamma$, with n represents the unit outward normal vector of the interface Γ. We always assume that the problem (2.1a)-(2.1d) is well-posed. To solve this high-frequency interface problem, we combine HOrderDNN with the traditional DDM and propose the HOrderDeepDDM.

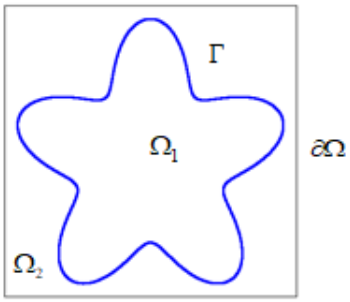


Figure 1: The diagram of domain Ω.

In this section, we will introduce the traditional DDM and the structure of HOrderDNN separately. Besides, we also present the minimization problems utilized to solve interface problems in the least square formulation and show the method of setting the penalty coefficients in the loss function for the interface problems.

2.1 Domain decomposition method for PDEs

We first introduce the traditional DDM for solving problem (2.1a)-(2.1d). As shown in Fig. 1, Ω is naturally decomposed into two non-overlapping subdomains Ω_1 and Ω_2 by the interface Γ. The DDM starts with initial guesses ω_1^0, ω_2^0 along Γ and then computes ω_1^i and ω_2^i, $i=1,2,\cdots$ in parallel, as follows,

$$\begin{cases} \mathcal{L}(u_1^i)=f & \text{in } \Omega_1, \\ u_1^i=\omega_1^{i-1}+p & \text{on } \Gamma, \end{cases} \tag{2.2a}$$

$$\begin{cases} \mathcal{L}(u_2^i)=f & \text{in } \Omega_2, \\ \mathcal{B}(u_2^i)=g & \text{on } \partial\Omega, \\ \partial_n u_2^i=\omega_2^{i-1}-q & \text{on } \Gamma, \end{cases} \tag{2.2b}$$

where $\omega_1^i=u_2^i$ and $\omega_2^i=\partial_n u_1^i$ on Γ. Note that, we show only a common way of dealing with interface conditions in the traditional DDM, which is usually different for different problems and different DDMs, see Sections 3.3 and 3.5 for more details.

One of the most popular stopping strategies is the relative error of the current solution with respect to the previous one on the artificial interface or inside the sub-domain less than a given tolerance tol_Γ or tol_Ω, in formulas

$$\frac{\|\omega_1^i-\omega_1^{i-1}\|}{\|\omega_1^i\|}<tol_\Gamma, \qquad \frac{\|\omega_2^i-\omega_2^{i-1}\|}{\|\omega_2^i\|}<tol_\Gamma, \tag{2.3a}$$

$$\frac{\|u_1^i-u_1^{i-1}\|}{\|u_1^i\|}<tol_\Omega, \qquad \frac{\|u_2^i-u_2^{i-1}\|}{\|u_2^i\|}<tol_\Omega. \tag{2.3b}$$

The parallelizable nature of DDM allows it to efficiently solve complex PDEs, especially those with irregular interfaces, and it is playing an increasingly important role in the field of engineering and scientific computing.

2.2 High order deep neural network

In a recent work [2], HOrderDNN is proposed, which adds a nonlinear transformation layer to the traditional fully connected neural network to improve the approximation ability of the network. The architecture of HOrderDNN can be expressed as

$$h_p(x;\theta)=\mathbf{F}_{L+1}\circ\sigma\circ\mathbf{F}_L\circ\sigma\circ\cdots\circ\mathbf{F}_2\circ\sigma\circ\mathbf{F}_1\circ\mathbf{T}_p(x),$$

where p is the order of HOrderDNN, σ is the activation function, $\mathbf{F}_l$, $1\le l\le L$ are the L hidden layers taking the form of

$$\mathbf{F}_l(x)=W_l x+b_l,$$

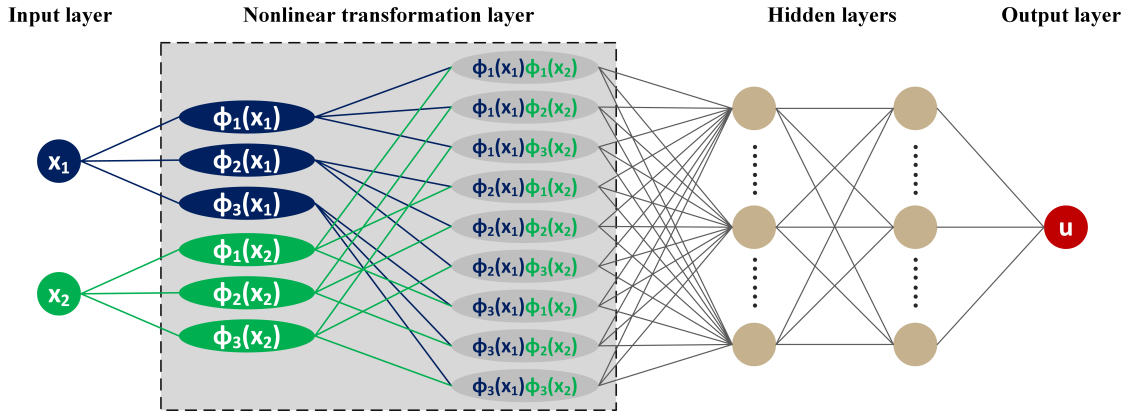


Figure 2: Illustration of the architecture of HOrderDNN ($p=2$) with $d=2$. The Only difference to commonly-used DNNs is the nonlinear transformation layer followed the input layer.

with $W_l \in \mathbb{R}^{n_l \times n_{l-1}}$, $b \in \mathbb{R}^{n_l}$, n_l being the number of neurons in the l-th layer, and $\theta := \{W_l, b_l\}$ is the collection of all parameters. The nonlinear layer T_p converts the input x into Lagrangian interpolation basis functions $\{\Phi_i\}$ from the polynomial space $\mathcal{Q}_p(\mathbb{R}^d)$, which consists of all polynomials on $\mathbb{R}^d$ with degrees not exceeding p. In detail, the basis functions are shown as follows:

$$\phi_i(x) = \prod_{j=1, j\neq i}^{p+1} (x - x_j) \Big/ \prod_{j=1, j\neq i}^{p+1} (x_i - x_j), \quad i = 1, 2, \cdots, p+1,$$

where the interpolating nodes x_i, $i = 1, 2, \cdots, p+1$ are chosen as Gauss-Lobatto-Legendre (GLL) points in the integration interval or calculation area. Fig. 2 illustrates the network architecture and the nonlinear transformation layer T_p when the order $p=2$ and the number of input neurons $n_0 = 2$.

The high order information introduced by the nonlinear layer makes it possible for HOrderDNN to approximate high-frequency information, which enables HOrderDNN to be used for various high-frequency problems. Besides, HOrderDNN also has the advantages of faster convergence and higher accuracy than conventional neural networks with the same number of trainable parameters [2].

2.3 HOrderDeepDDM for PDEs

In order to deal with high-frequency interface problems, we combine DDM with HOrderDNN and propose HOrderDeepDDM, where DDM is adopted to mainly deal with the complex interface and HOrderDNN is introduced to help characterize high-frequency information. In the case of the problem (2.1a)-(2.1d), it is natural to decompose Ω into Ω_1 and Ω_2, and subproblems will be solved by HOrderDNNs on Ω_1 and Ω_2, separately. To be specific, problem (2.2a) and (2.2b) will be reformulated as minimization

problems

$$\theta^* = \arg\min_{\theta} \mathcal{M}_{\Omega_1}(\theta) + \beta_\Gamma * \mathcal{M}_{\Gamma_1}(\theta), \tag{2.4a}$$

$$\theta^* = \arg\min_{\theta} \mathcal{M}_{\Omega_2}(\theta) + \beta_{\partial\Omega} * \mathcal{M}_{\partial\Omega}(\theta) + \beta_\Gamma * \mathcal{M}_{\Gamma_2}(\theta), \tag{2.4b}$$

where the discrete least squared residuals are defined as

$$\mathcal{M}_{\Omega_i}(\theta) := \frac{1}{N_r}\sum_{j=1}^{N_r} |\mathcal{L}(h_p^i(x_r^j;\theta)) - f(x_r^j)|^2, \qquad i=1,2, \tag{2.5a}$$

$$\mathcal{M}_{\partial\Omega}(\theta) := \frac{1}{N_b}\sum_{j=1}^{N_b} |\mathcal{B}(h_p^2(x_b^j;\theta)) - g(x_b^j)|^2, \tag{2.5b}$$

$$\mathcal{M}_{\Gamma_1}(\theta) := \frac{1}{N_{if}}\sum_{j=1}^{N_{if}} |(h_p^1(x_{if}^j;\theta)) - \omega_1^{i-1}(x_{if}^j) - p(x_{if}^j)|^2, \tag{2.5c}$$

$$\mathcal{M}_{\Gamma_2}(\theta) := \frac{1}{N_{if}}\sum_{j=1}^{N_{if}} |(\partial_n h_p^2(x_{if}^j;\theta)) - \omega_2^{i-1}(x_{if}^j) + q(x_{if}^j)|^2, \tag{2.5d}$$

with $\{x_r^j\}_{j=1}^{N_r}$, $\{x_b^j\}_{j=1}^{N_b}$ and $\{x_{if}^j\}_{j=1}^{N_{if}}$ being the collocation points inside Ω_i, on $\partial\Omega$ and Γ, respectively. $h_p^i(x;\theta)$ represents the approximate solution of HOrderDNN to u^i, $i=1,2$. The above optimization problems can be solved by the stochastic gradient descent method, such as Adam [11].

We remark that the weighting coefficients $\beta_{\partial\Omega}$, β_Γ in the loss function play a very important role in balancing the errors in the domain, on the boundary and the interface during the training process. In order to adjust these parameters properly, we adopt a strategy similar to the learning rate annealing method proposed by [19]. In the case of Eq. (2.4b), for general stochastic gradient descent methods, the update of the network parameters θ can be expressed as:

$$\theta^{k+1} = \theta^k - \tau\cdot\nabla_\theta\mathcal{M}_{\Omega_2} - \tau\cdot\beta_{\partial\Omega}\nabla_\theta\mathcal{M}_{\partial\Omega} - \tau\cdot\beta_\Gamma\nabla_\theta\mathcal{M}_{\Gamma_2}, \tag{2.6}$$

where τ denotes the learning rate, and k is the iteration step. In [19], the estimates of $\beta_{\partial\Omega}$ and β_Γ can be computed by:

$$\hat{\beta}_{\partial\Omega}^{k+1} = \frac{\max_\theta\{|\nabla_\theta\mathcal{M}_{\Omega_2}|\}}{\text{mean}_\theta\{|\nabla_\theta\beta_{\partial\Omega}^k\mathcal{M}_{\partial\Omega}|\}}, \qquad \hat{\beta}_\Gamma^{k+1} = \frac{\max_\theta\{|\nabla_\theta\mathcal{M}_{\Omega_2}|\}}{\text{mean}_\theta\{|\nabla_\theta\beta_\Gamma^k\mathcal{M}_{\Gamma_2}|\}}. \tag{2.7}$$

The above method is named as "M_1" and as an alternative, we propose the following method named "M_2" to compute $\beta_{\partial\Omega}$ and β_Γ:

$$\hat{\beta}_{\partial\Omega}^{k+1} = \frac{\max_\theta\{|\nabla_\theta\mathcal{M}_{\Omega_2}|\}}{\text{mean}_\theta\{|\nabla_\theta\beta_{\partial\Omega}^k\mathcal{M}_{\partial\Omega}|\}}, \qquad \hat{\beta}_\Gamma^{k+1} = \frac{\max_\theta\{|\nabla_\theta\mathcal{M}_{\Omega_2}|\}}{\max_\theta\{|\nabla_\theta\beta_\Gamma^k\mathcal{M}_{\Gamma_2}|\}}. \tag{2.8}$$

Then, the weighting coefficients for the next iteration are updated using the following form:

$$\beta_{\partial\Omega}^{k+1}=(1-\lambda)\beta_{\partial\Omega}^{k}+\lambda\hat{\beta}_{\partial\Omega}^{k+1},\qquad \beta_{\Gamma}^{k+1}=(1-\lambda)\beta_{\Gamma}^{k}+\lambda\hat{\beta}_{\Gamma}^{k+1},\tag{2.9}$$

with $\lambda=0.1$. A comparison of "M_1" and "M_2" is shown in Section 3.3.1, and the better performing strategy "$M2$" will be applied in the following experiment.

3 Numerical experiments

In this section, a series of numerical examples are presented, including the high-frequency elliptic and Helmholtz interface problem, to demonstrate the properties of HOrderDeepDDM. In Section 3.2, we first consider the low-frequency elliptic interface problem to illustrate the greater approximation ability of HOrderDeepDDM than DeepDDM. We then consider the high-frequency elliptic interface problem to illustrate the ability of HOrderDeepDDM to approximate high-frequency functions with strong discontinuities across the complex interface in Section 3.3. Further, the cases of high contrast coefficients, complex interfaces and high-dimensional interface problems are shown in Section 3.3.2, Section 3.4 and Section 3.6. The high-frequency Helmholtz interface problem is also taken into account, which is shown in Section 3.5. In all numerical experiments, DeepDDM [13] is chosen as the benchmark for comparison.

3.1 Settings

We summarize the standard setups in our experiments as follows. The stop condition of the outer iteration is that the relative $\mathcal{L}^2$ error of two adjacent transmission conditions at the interface is less than the given threshold, see Eqs. (2.3a)-(2.3b) for details. Here, we set $tol_\Gamma=tol_\Omega=0.01$. For each sub-problem, the hidden layers of the neural network are chosen as fully connected layers with an equal number of neurons. The number of hidden layers is defined as the depth, and the number of neurons in the hidden layers is defined as the width. MSE is chosen as the loss function to train the neural network, and the Xavier initialization method is used to initialize the network parameters. The activation function is chosen as Tanh, and the optimizer is chosen as Adam [11]. For the learning rate τ, it starts with a best initial value between $10^{-2}\sim10^{-3}$ and decays every 100 steps with a base of 0.99. At the same time, we calculate the standard deviation of the loss every 100 epochs, and the inner iteration stops when the standard deviation is less than 5×10^{-3} or epoch$=$10000. To demonstrate the advantages of our algorithm, we define the relative $\mathcal{L}^2$ error

$$err=\frac{\|u(x;\theta)-u_*\|_2}{\|u_*\|_2},\tag{3.1}$$

where u_* represents the target function and $u(x;\theta)$ represents the approximate solution of the neural network, and

$$\|u_*\|_2^2=\int_\Omega|u_*|^2dx.$$

Table 1: Summary of parameters for HOrderDeepDDM.

Notation	Stands for ⋯
τ	The learning rate
L	The depth of the network
W	The width of the network
p	The order of HOrderDeepDDM
N_r	Number of training sampling points in the region
N_{if}	Number of training sampling points on the interface
N_b	Number of training sampling points on the boundary
$\beta_{\partial\Omega}$	The penalty coefficient for the boundary term in Eq. (2.4b)
β_Γ	The penalty coefficient for the interface term in Eq. (2.4b)

The relative $\mathcal{L}^2$ error is calculated over equidistant grid points selected for the test data. Notations of parameters are summarized in Table 1, and the value of these parameters are given in the following experiments.

3.2 Low-frequency elliptic interface problem

To verify the effectiveness of our proposed method, we perform a simple test on the following low-frequency elliptic interface problem in Section 4.4.2 in [7]:

$$-\nabla(\alpha(x)\nabla u)=f \qquad \text{in } \Omega_1\cup\Omega_2, \tag{3.2a}$$
$$u=g \qquad \text{on } \partial\Omega, \tag{3.2b}$$
$$[[u]]=0 \qquad \text{on } \Gamma, \tag{3.2c}$$
$$[[\alpha\partial_n u]]=0 \qquad \text{on } \Gamma, \tag{3.2d}$$

where $\Omega:=[-1,1]\times[-1,1]$, $\Gamma:=\{(x,y)|x^2+y^2=0.25\}$,

$$\alpha(x)=\begin{cases}\alpha_1, & (x,y)\in\Omega_1:=\{(x,y)|x^2+y^2<0.25\},\\ \alpha_2, & (x,y)\in\Omega_2:=\Omega\backslash\Omega_1.\end{cases}$$

The exact solution is

$$u_*(x,y)=\begin{cases}\dfrac{r^3}{\alpha_1}, & (x,y)\in\Omega_1,\\ \dfrac{r^3}{\alpha_2}+\Big(\dfrac{1}{\alpha_1}-\dfrac{1}{\alpha_2}\Big)r_0^3, & (x,y)\in\Omega_2,\end{cases}$$

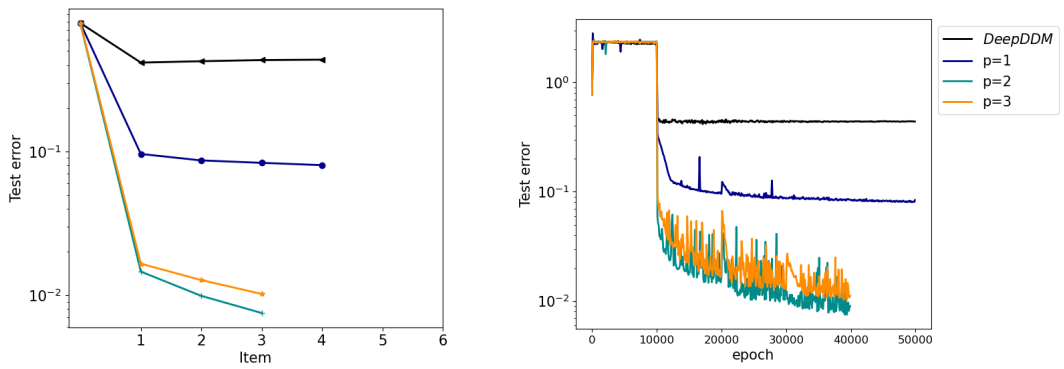
where $r=\sqrt{x_1^2+x_2^2}$, $r_0=0.5$ $\alpha_1=1$, $\alpha_2=1000$. The functions f, g can be directly computed from $u_*(x,y)$.

DeepDDM and HOrderDeepDDMs with $L=1$, $W=10$, $p=1,2,3$ are utilized to fit the numerical solution and the hyper-parameters are set as follows: $N_r=N_{if}=N_b=128$.

Table 2: The relative $\mathcal{L}^2$ errors and the number of outer iterations obtained by DeepDDM and HOrderDeepDDMs (p) on problem (3.2a)-(3.2d). The figures in parentheses are the numbers of outer iterations. Here, $L=1$, $W=10$, $N_r=N_b=N_{if}=128$.

	DeepDDM	$p=1$	$p=2$	$p=3$
err	4.36E-1(5)	8.03E-2(5)	7.50E-3(4)	1.02E-2(4)

The results are shown in Table 2. As we can see, although the proposed method aims at solving high-frequency interface problems, it still outperforms DeepDDM for low-frequency problems when the architecture keeps the same. In addition, we show the convergence process of the relative $\mathcal{L}^2$ error with the outer and inner iterations in Fig. 3, and it is clear that the convergence speed of HOrderDeepDDMs is much faster than that of DeepDDM.



(a) the outer iteration (b) the inner iteration

Figure 3: The change in relative $\mathcal{L}^2$ error along with outer and inner iterations for DeepDDM and HOrderDeepDDM on problem (3.2a)-(3.2d). Here, $L=1$, $W=10$, $N_r=N_b=N_{if}=128$.

3.3 High-frequency elliptic interface problem

To test the performance of HOrderDeepDDM on high-frequency interface problems, we first consider the following two-dimensional high-frequency elliptic interface problem:

$$-\nabla(\alpha(x)\nabla u)=f \quad \text{in } \Omega_1\cup\Omega_2, \tag{3.3a}$$
$$u=g \quad \text{on } \partial\Omega, \tag{3.3b}$$
$$[[u]]=p \quad \text{on } \Gamma, \tag{3.3c}$$
$$[[\alpha\partial_n u]]=q \quad \text{on } \Gamma, \tag{3.3d}$$

where $\Omega:=[0,2]\times[0,2]$, $\Gamma:=\{(x,y)|(x-1)^2+(y-1)^2=0.25\}$,

$$\alpha(x)=\begin{cases}\alpha_1, & (x,y)\in\Omega_1:=\{(x,y)|(x-1)^2+(y-1)^2<0.25\},\\ \alpha_2, & (x,y)\in\Omega_2:=\Omega\backslash\Omega_1.\end{cases}$$

We consider the following high-frequency exact solution with discontinuities at the interface:

$$u_*(x,y)=\begin{cases}\sin(20x)\cdot\cos(20y), & (x,y)\in\Omega_1,\\ \sin(40x)\cdot\cos(40y), & (x,y)\in\Omega_2.\end{cases}$$

The boundary condition g and the interface conditions p, q can be directly computed from $u_*(x,y)$. For the high-frequency elliptic equation, we consider the parallel DDM, the details of which can be seen in Algorithm 1 of [13]. Therefore, Eqs. (3.3a)-(3.3d) can be written as a multidomain formulation:

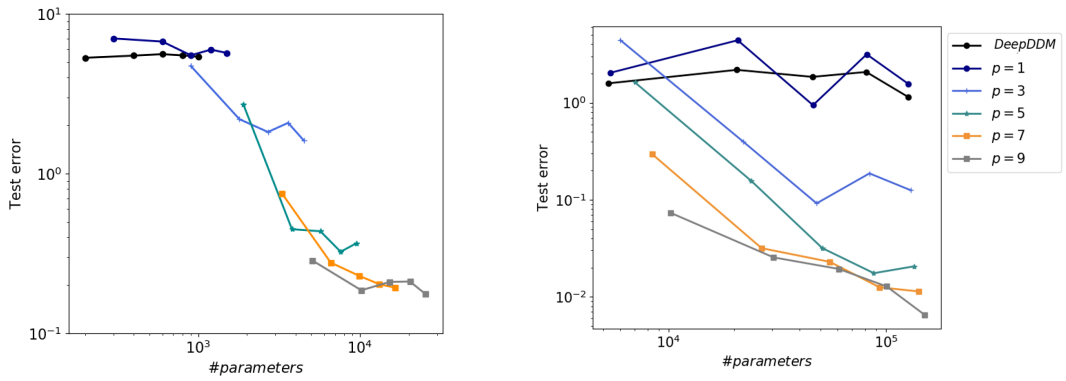
$$\begin{cases}-\alpha_1\cdot\Delta u_1=f_1 & \text{in } \Omega_1,\\ u_1=u_2+p & \text{on } \Gamma,\end{cases}\qquad\begin{cases}-\alpha_2\cdot\Delta u_2=f_2 & \text{in } \Omega_2,\\ u=g & \text{on } \partial\Omega_2\backslash\Gamma,\\ \alpha_2\partial_n u_2=\alpha_1\partial_n u_1-q & \text{on } \Gamma.\end{cases}\tag{3.4}$$

3.3.1 Case with $\alpha_1=1$, $\alpha_2=2$

As a benchmark test, we apply HOrderDeepDDM to solve problem (3.3a)-(3.3d) with the interface coefficients $\alpha_1=1$, $\alpha_2=2$. DeepDDM and HOrderDeepDDMs with network architectures defined by $L=3$, $W=200$, $p=[1,3,5,7,9]$ are examined, and the numbers of training points are chosen as $N_r=N_b=N_{if}=4096$.

In this experiment, we compare the two adaptive learning rate annealing methods given in Section 2.3. The number of outer iterations when the training process stops and the corresponding relative $\mathcal{L}^2$ errors for different methods are shown in Table 3. As expected, HOrderDeepDDMs ($p>1$) always outperform DeepDDM with the same architecture, and when p increases, more accurate numerical solutions are obtained by HOrderDeepDDMs, regardless of which adaptive learning rate annealing method is used. In addition, compared to the "$M1$" method, HOrderDeepDDMs ($p=7,9$) with the "$M2$" method require fewer outer iterations to achieve the same error, which means that choosing the "$M2$" method can save about $1/8$ of the training time. Therefore, the method "$M2$" is selected as the adaptive learning rate annealing method in the following experiments.

In fact, since HOrderDNNs use a fully connected network architecture, the increase of p leads to the increase of the number of network parameters [2]. So, we show the change of the relative $\mathcal{L}^2$ errors along with the number of parameters for different p in Fig. 4, where DeepDDMs and HOrderDeepDDMs with $L=1,3$, $W=50,100,150,200,250$, $p=1,3,5,7,9$ are considered. It can be observed that if the parameters are fixed, increasing the value of p yields a smaller relative error in almost all cases; and to achieve the same



(a) $L=1$ (b) $L=3$

Figure 4: The change of the relative $\mathcal{L}^2$ errors along with the number of parameters for HOrderDeepDDMs on problem (3.3a)-(3.3d) with $\alpha_1=1$, $\alpha_2=2$. Here, the hyper-parameters are set as follows: $L=1,3$, $W=50,100,150,200,250$, $N_r=N_b=N_{if}=4096$.

error, HOrderDeepDDMs with larger p obviously require fewer parameters. It is easy to find that HOrderDeepDDM ($p=9$) achieves the smallest relative $\mathcal{L}^2$ error 1.28×10^{-2} when the network architecture is $L=3$, $W=200$. Therefore, the network architecture is fixed to $L=3$, $W=200$ in the subsequent experiments.

The change in relative $\mathcal{L}^2$ error along with outer iterations for DeepDDM and HOrderDeepDDM is shown in Fig. 5. Note that the relative $\mathcal{L}^2$ errors of DeepDDM and HOrderDeepDDM ($p=1$) are always larger than 1 as the outer iteration proceeds, which indicates that the performance of DeepDDM is very poor in solving high-frequency interface problems, while the errors of HOrderDeepDDMs ($p=7,9$) rapidly decrease to 1%. The convergence processes of losses and relative $\mathcal{L}^2$ errors are shown in Fig. 6. It is easy to see that the convergence rate of HOrderDeepDDMs significantly improves with increasing p, while the errors of DeepDDM remain almost undiminished, and the number of training iterations required for error convergence is also decreasing as p increases. In addition, it is observed that there is a temporary rise in the error on Ω_2 after the interface information is exchanged, while as the outer iteration proceeds, the error rise gradually decreases, which implies the increasing accuracy of the interface predictions provided by Ω_1 to Ω_2.

Table 3: The relative $\mathcal{L}^2$ errors and the number of outer iterations obtained by DeepDDM and HOrderDeepDDMs (p) on problem (3.3a)-(3.3d) when $\alpha_1=1$, $\alpha_2=2$. The figures in parentheses are the numbers of outer iterations. Here, $L=3$, $W=200$, $N_r=N_b=N_{if}=4096$.

	DeepDDM	$p=1$	$p=3$	$p=5$	$p=7$	$p=9$
err_{M1}	2.16E+0(8)	2.80E+0(10)	1.21E-1(9)	2.40E-2(8)	1.31E-2(8)	1.22E-2(8)
err_{M2}	1.18E+0(9)	3.15E+0(9)	1.87E-1(9)	1.75E-2(8)	1.25E-2(7)	**1.28E-2(7)**

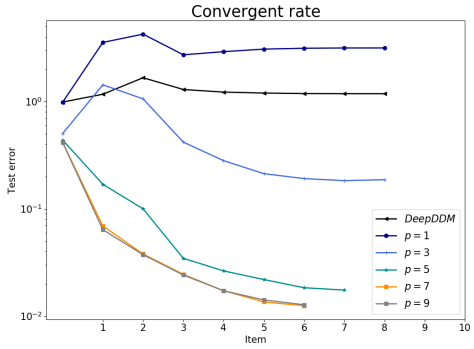
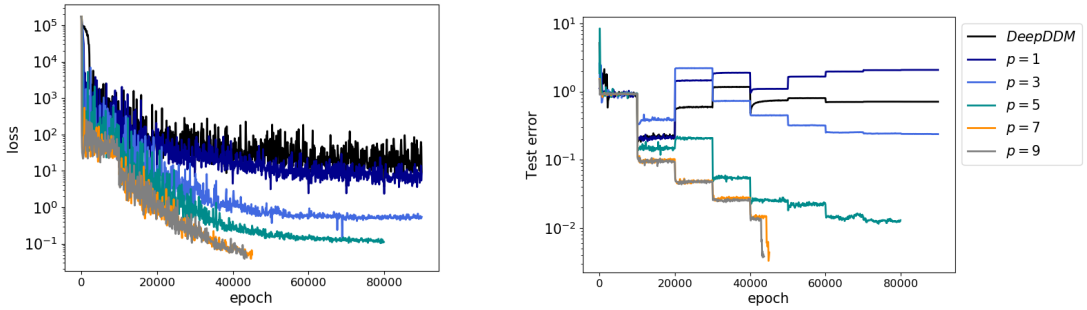
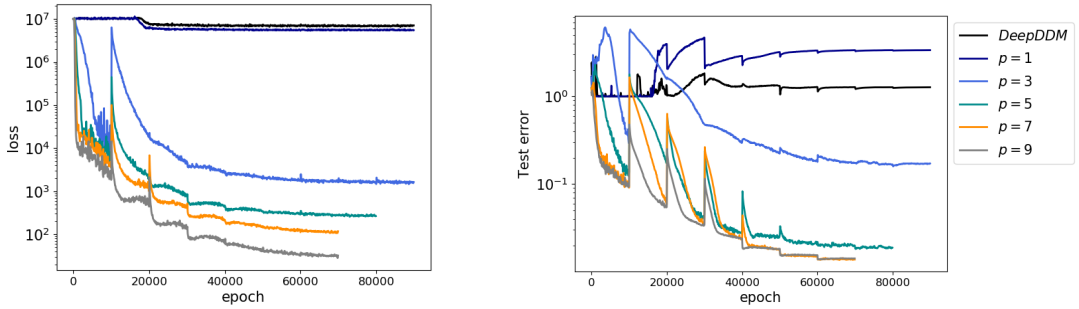


Figure 5: The change in relative $\mathcal{L}^2$ error along with outer iterations for DeepDDM and HOrderDeepDDM on problem (3.3a)-(3.3d). Here, the hyper-parameters are set as follows: $\alpha_1 = 1$, $\alpha_2 = 2$, $L = 3$, $W = 200$, $N_r = N_b = N_{if} = 4096$.

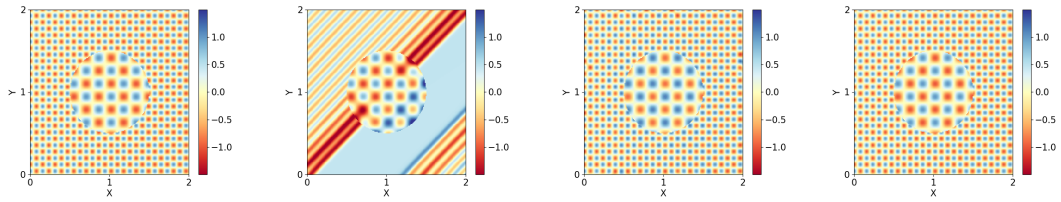


(a) Ω_1 loss (b) Ω_1 error



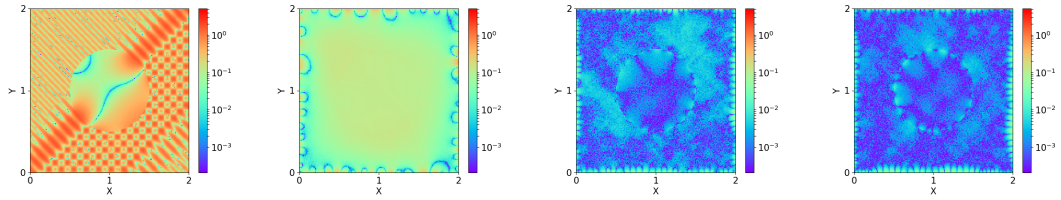
(c) Ω_2 loss (d) Ω_2 error

Figure 6: The convergence processes of losses and relative $\mathcal{L}^2$ errors in inner iterations on problem (3.3a)-(3.3d) with $\alpha_1 = 1$, $\alpha_2 = 2$. Here, the hyper-parameters are set as follows: $L = 3$, $W = 200$, $N_r = N_b = N_{if} = 4096$.



(a) Exact (b) DeepDDM (c) $p=3$ (d) $p=9$

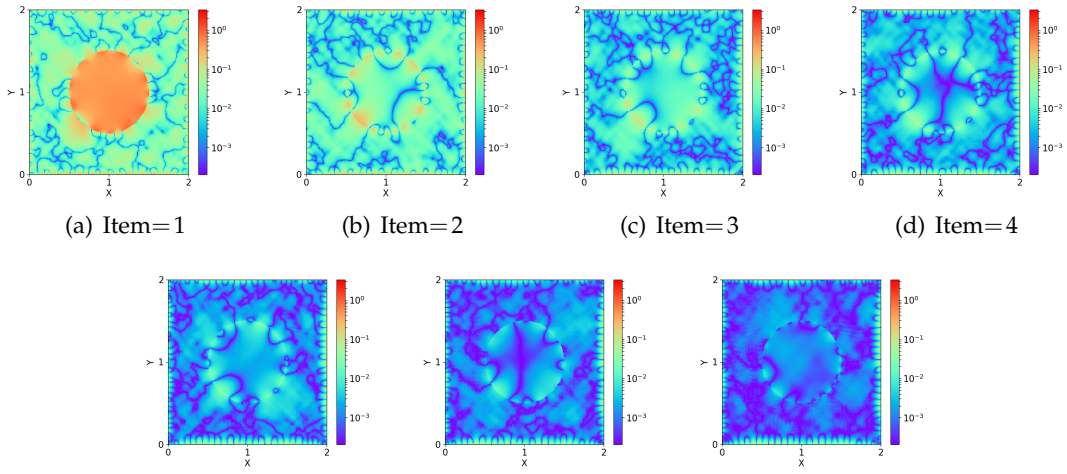
Figure 7: Exact and numerical solutions of DeepDDM and HOrderDeepDDMs with different p on problem (3.3a)-(3.3d). (a) illustrates the target function and (b)-(d) are fitting functions given by DeepDDM and HOrderDeepDDMs with $p=3,9$, respectively. Here, the hyper-parameters are set as follows: $\alpha_1=1$, $\alpha_2=2$, $L=3$, $W=200$, $N_r=N_b=N_{if}=4096$.



(a) DeepDDM (b) $p=3$ (c) $p=7$ (d) $p=9$

Figure 8: Fitting errors given by DeepDDM and HOrderDeepDDMs with different p on problem (3.3a)-(3.3d). The hyper-parameters are set as follows: $\alpha_1=1$, $\alpha_2=2$, $L=3$, $W=200$, $N_r=N_b=N_{if}=4096$.

We also compare the fitting effects of DeepDDM and HOrderDeepDDMs with different values of p. The exact solution and the approximate solutions are shown in Fig. 7, and the point-by-point errors are shown in Fig. 8. The numerical solution obtained by DeepDDM in Fig. 7(b) fails to capture the local oscillations, while the numerical solutions obtained by HOrderDeepDDMs ($p>1$) in Figs. 7(c)-(d) capture the oscillations of different scales well. This advantage of HOrderDeepDDMs can also be seen in Fig. 8, that the errors on Ω_1 and Ω_2 both decrease significantly as p increases. These phenomena indicate that HOrderDeepDDMs ($p>1$) can approximate the high-frequency information of the target function, and this approximation ability increases as p increases. In addition, we show the point-by-point errors of HOrderDeepDDM ($p=9$) at each outer iteration in Fig. 9. As expected, in the beginning, the errors on Ω_1 are very large, which is due to the wrong transmission conditions at the interface, while it decreases rapidly after exchanging information with Ω_2. The errors at the interface also decrease significantly as the outer iteration increases. After seven outer iterations, the algorithm reaches the stop criterion, and the errors reach the order of 10^{-3}.



(a) Item=1 (b) Item=2 (c) Item=3 (d) Item=4

(e) Item=5 (f) Item=6 (g) Item=7

Figure 9: Fitting errors in the whole domains for outer iterations given by HOrderDeepDDM ($p=9$) on problem (3.3a)-(3.3d). The hyper-parameters are set as follows: $\alpha_1=1$, $\alpha_2=2$, $L=3$, $W=200$, $N_r=N_b=N_{if}=4096$.

3.3.2 Case with $\alpha_1=1$, $\alpha_2=200$

High contrast interface coefficients often lead to strong discontinuities at the interface, which can cause additional difficulties in the numerical solution. In this section, we test the performance of HOrderDeepDDM on high-frequency elliptic equations with high contrast coefficients. We apply HOrderDeepDDMs to solve problem (3.3a)-(3.3d) with $\alpha_1=1$, $\alpha_2=200$. DeepDDM and HOrderDeepDDMs with $p=[1,3,5,7,9]$, $L=3$, $W=200$ are utilized to fit the numerical solution. The numbers of training data are fixed as $N_r=N_b=N_{if}=4096$.

Based on the experimental results in Section 3.3.1, we use the adaptive learning rate annealing method 'M2', and the relative $\mathcal{L}^2$ errors for DeepDDM and HOrderDeepDDMs are shown in Table 4. As we can see, for problem (3.3a)-(3.3d) with high contrast coefficients, HOrderDeepDDMs($p\geq 7$) also outperform DeepDDM with the same network architecture, and the errors of HOrderDeepDDM ($p=9$) reach the order of 10^{-2} after seven outer iterations, while the errors of DeepDDM always remain larger than 1. The processes of error convergence of DeepDDM and HOrderDeepDDMs along with the outer

Table 4: The relative $\mathcal{L}^2$ errors and the number of outer iterations obtained by DeepDDM and HOrderDeepDDMs (p) on problem (3.3a)-(3.3d) when $\alpha_1=1$, $\alpha_2=200$. The figures in parentheses are the numbers of outer iterations. Here, $L=3$, $W=200$, $N_r=N_b=N_{if}=4096$.

	DeepDDM	$p=1$	$p=3$	$p=5$	$p=7$	$p=9$
err	2.67E+0(8)	6.14E+0(9)	2.67E+0(9)	2.83E+0(9)	2.69E-1(9)	**7.33E-2(7)**

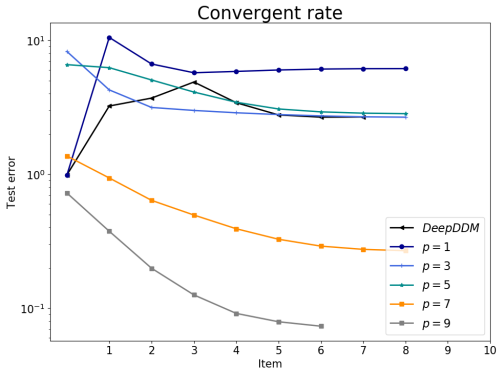
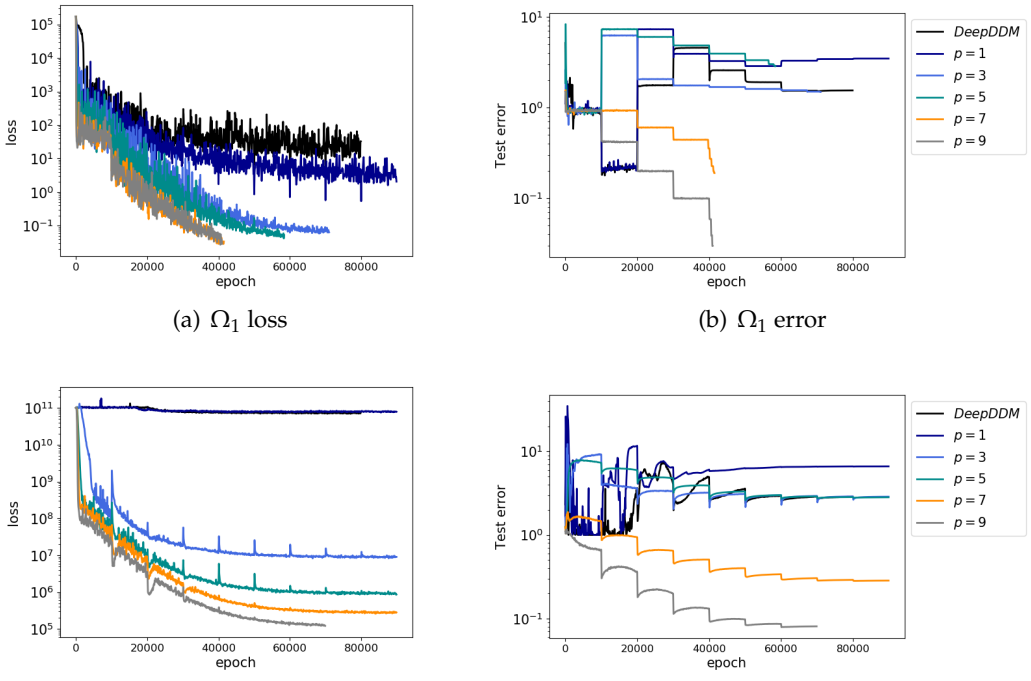


Figure 10: The change in relative $\mathcal{L}^2$ error along with outer iterations for DeepDDM and HOrderDeepDDM on problem (3.3a)-(3.3d). Here, the hyper-parameters are set as follows: $\alpha_1=1$, $\alpha_2=200$, $L=3$, $W=200$, $N_r=N_b=N_{if}=4096$.

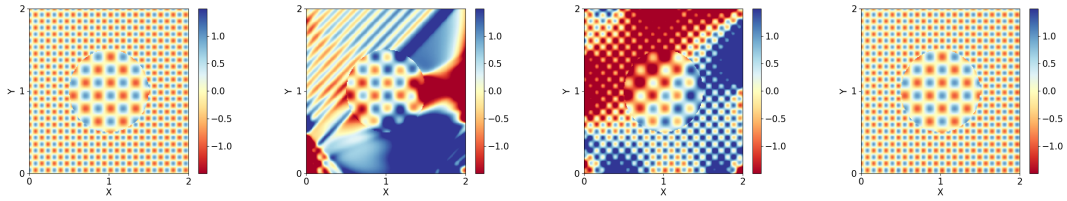


(a) Ω_1 loss (b) Ω_1 error

(c) Ω_2 loss (d) Ω_2 error

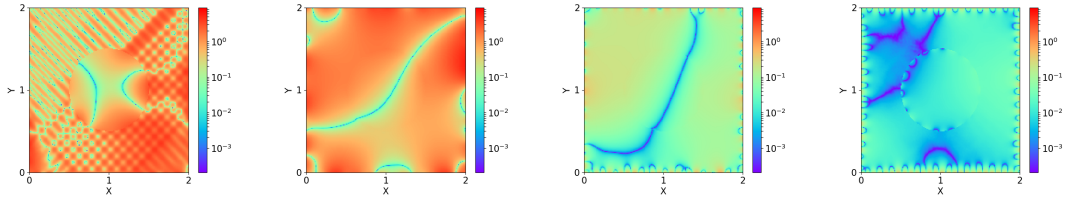
Figure 11: The convergence processes of losses and relative $\mathcal{L}^2$ errors in inner iterations on problem (3.3a)-(3.3d) with $\alpha_1=1$, $\alpha_2=200$. Here, the hyper-parameters are set as follows: $L=3$, $W=200$, $N_r=N_b=N_{if}=4096$.

and inner iterations are shown in Fig. 10 and Fig. 11. As before, HOrderDeepDDMs with larger p converge faster to lower errors, while the errors of DeepDDM remain almost



(a) Exact (b) DeepDDM (c) $p=3$ (d) $p=9$

Figure 12: Exact and numerical solutions of DeepDDM and HOrderDeepDDMs with different p on problem (3.3a)-(3.3d). (a) illustrates the target function and (b)-(d) are fitting functions given by DeepDDM and HOrderDeepDDMs with $p=3,9$, respectively. Here, the hyper-parameters are set as follows: $\alpha_1=1$, $\alpha_2=200$, $L=3$, $W=200$, $N_r=N_b=N_{if}=4096$.



(a) DeepDDM (b) $p=3$ (c) $p=7$ (d) $p=9$

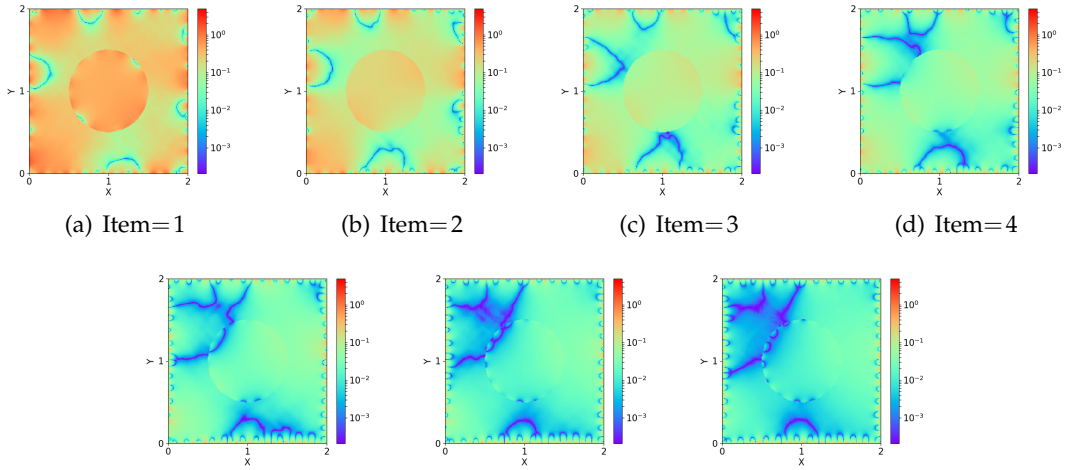
Figure 13: Fitting errors given by DeepDDM and HOrderDeepDDMs with different p on problem (3.3a)-(3.3d). The hyper-parameters are set as follows: $\alpha_1=1$, $\alpha_2=200$, $L=3$, $W=200$, $N_r=N_b=N_{if}=4096$.

unchanged.

The fitting effects of DeepDDM and HOrderDeepDDMs with different p are shown in Fig. 12, and the point-by-point errors are shown in Fig. 13. Same with Section 3.3.1, as p increases, the fit of HOrderDeepDDMs to the high-frequency part of the target function becomes better, which indicates that the performance of HOrderDeepDDMs in high contrast coefficients case is still promising. In the end, the point-by-point errors of HOrderDeepDDM ($p=9$) at each outer iteration are shown in Fig. 14, and the entire convergence process along with outer iterations can be clearly observed.

3.4 Flower shape interface problem

Problems with complex interfaces have always been challenging for traditional mesh-based methods such as FEM and FDM, while DNN-based methods are more flexible in handling complex interfaces. To test the effectiveness of HOrderDeepDDM for complex interface problems, we consider the high-frequency elliptic interface problem (3.3a)-(3.3d) with $\alpha_1=1$, $\alpha_2=200$ in domain $\Omega=[-1,1]\times[-1,1]$. The interface Γ is given by the



(a) Item=1 (b) Item=2 (c) Item=3 (d) Item=4

(e) Item=5 (f) Item=6 (g) Item=7

Figure 14: Fitting errors in the whole domains for outer iterations given by HOrderDeepDDM (p=9) on problem (3.3a)-(3.3d). The hyper-parameters are set as follows: α_1=1, α_2=200, L=3, W=200, $N_r=N_b=N_{if}$=4096.

following polar coordinate:

$$\begin{cases} x=r(\theta)\cdot\cos(\theta), \\ y=r(\theta)\cdot\sin(\theta), \end{cases} \tag{3.5}$$

where

$$r(\theta)=\frac{1}{2}+\frac{\sin(5\theta)}{7}.$$

DeepDDM and HOrderDeepDDMs with L=3, W=200, p=1,3,5,7,9 are utilized to fit the numerical solution. The numbers of training points are set as $N_r=N_b=N_{if}$=4096.

The relative $\mathcal{L}^2$ errors and the number of outer iterations obtained by DeepDDM and HOrderDeepDDMs (p) are shown in Table 5, and the change in relative $\mathcal{L}^2$ errors along with outer and inner iterations for DeepDDM and HOrderDeepDDMs is shown in Fig. 15 and Fig. 16. As before, HOrderDeepDDMs ($p>1$) perform much better than DeepDDM with the same architecture. The error of HOrderDeepDDM ($p=9$) reaches the order of 10^{-1} after three outer iterations and stays at 7.52×10^{-2} when the iterations stop. More

Table 5: The relative $\mathcal{L}^2$ errors and the number of outer iterations obtained by DeepDDM and HOrderDeepDDMs (p) on problem (3.5) when α_1=1, α_2=200. The figures in parentheses are the numbers of outer iterations. Here, L=3, W=200, $N_r=N_b=N_{if}$=4096.

	DeepDDM	$p=1$	$p=3$	$p=5$	$p=7$	$p=9$
err	9.36E+0(9)	4.07E+0(9)	3.31E-1(9)	3.38E-1(10)	1.18E-1(9)	**7.50E-2(6)**

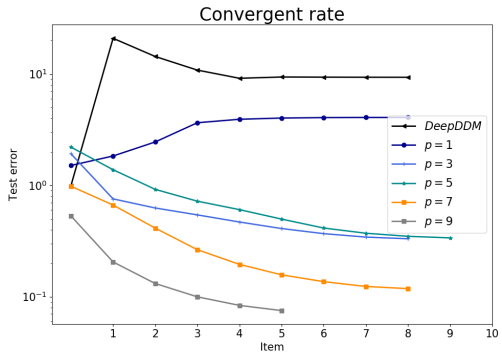
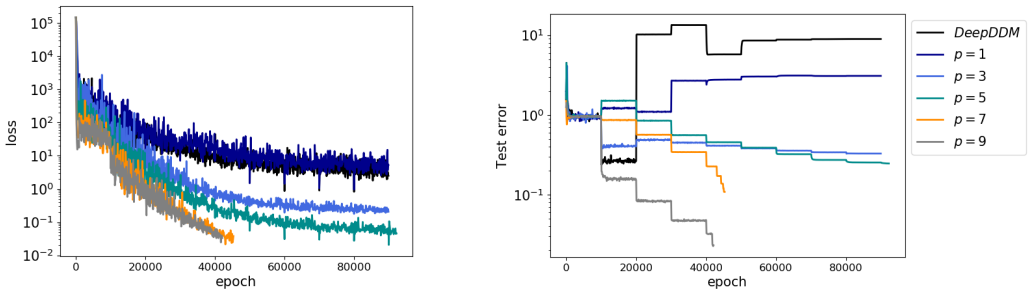
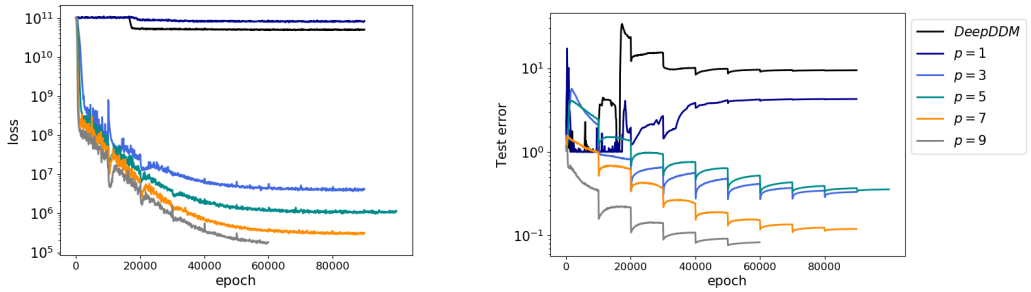


Figure 15: The change in relative $\mathcal{L}^2$ error along with outer iterations for DeepDDM and HOrderDeepDDM on problem (3.5). Here, the hyper-parameters are set as follows: $\alpha_1=1$, $\alpha_2=200$, $L=3$, $W=200$, $N_r=N_b=N_{if}=4096$.



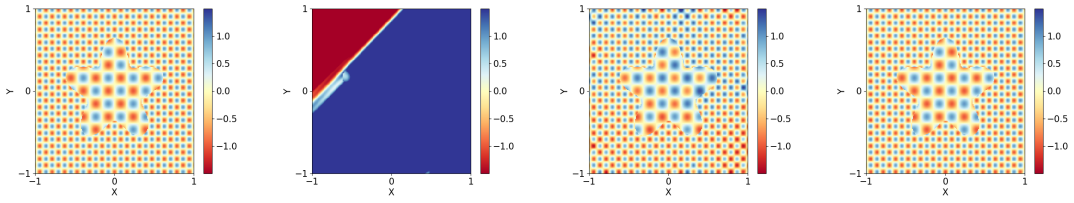
(a) Ω_1 loss (b) Ω_1 error



(c) Ω_2 loss (d) Ω_2 error

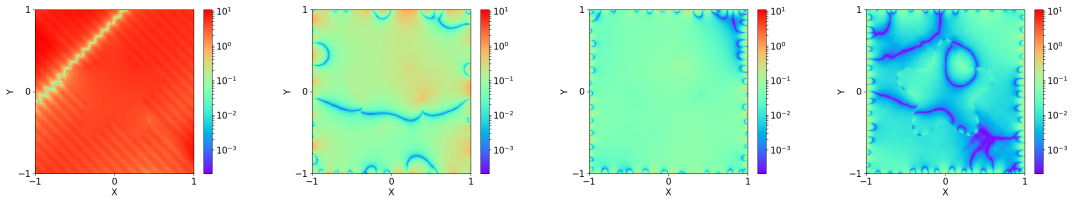
Figure 16: The convergence processes of losses and relative $\mathcal{L}^2$ errors in inner iterations on problem (3.5) with $\alpha_1=1$, $\alpha_2=200$. Here, the hyper-parameters are set as follows: $L=3$, $W=200$, $N_r=N_b=N_{if}=4096$.

importantly, the total number of training iterations required for HOrderDeepDDM (p=9) to converge is much smaller than DeepDDM, which indicates that the incorporation of



(a) Exact (b) DeepDDM (c) $p=3$ (d) $p=9$

Figure 17: Exact and numerical solutions of DeepDDM and HOrderDeepDDMs with different p on problem (3.5). (a) illustrates the target function and (b)-(d) are fitting functions given by DeepDDM and HOrderDeepDDMs with $p=3,9$, respectively. Here, the hyper-parameters are set as follows: $\alpha_1=1$, $\alpha_2=200$, $L=3$, $W=200$, $N_r=N_b=N_{if}=4096$.



(a) DeepDDM (b) $p=3$ (c) $p=7$ (d) $p=9$

Figure 18: Fitting errors given by DeepDDM and HOrderDeepDDMs with different p on problem (3.5). The hyper-parameters are set as follows: $\alpha_1=1$, $\alpha_2=200$, $L=3$, $W=200$, $N_r=N_b=N_{if}=4096$.

higher order ideas greatly improves the convergence speed of DeepDDM.

The comparison of the exact solution and numerical solutions of DeepDDM and HOrderDeepDDMs is shown in Fig. 17, and the corresponding point-by-point errors are shown in Fig. 18. The fitting effects of HOrderDeepDDMs ($p>1$) are much better than DeepDDM, which cannot fit any local oscillations in the exact solution, and the effects improve with increasing p. The point-by-point errors of HOrderDeepDDM ($p=9$) at each outer iteration are shown in Fig. 19, and we can see the process of error convergence. These phenomena show that the performance of HOrderDeepDDM is also guaranteed when solving high-frequency elliptic equations with complex interfaces, which is a significant advantage of HOrderDeepDDM over traditional DDMs.

3.5 High-frequency Helmholtz interface problem

To demonstrate the effectiveness of HOrderDeepDDM for high-frequency Helmholtz interface problems, we consider the following problem:

$$-\Delta u - k^2 u = 0 \qquad \Omega_1 \cup \Omega_2, \tag{3.6a}$$

$$\frac{\partial u}{\partial \mathbf{n}} + iku = g \qquad \text{on } \partial\Omega, \tag{3.6b}$$

$$[[u]] = 0 \qquad \text{on } \Gamma, \tag{3.6c}$$

$$[[\partial_n u]] = 0 \qquad \text{on } \Gamma, \tag{3.6d}$$

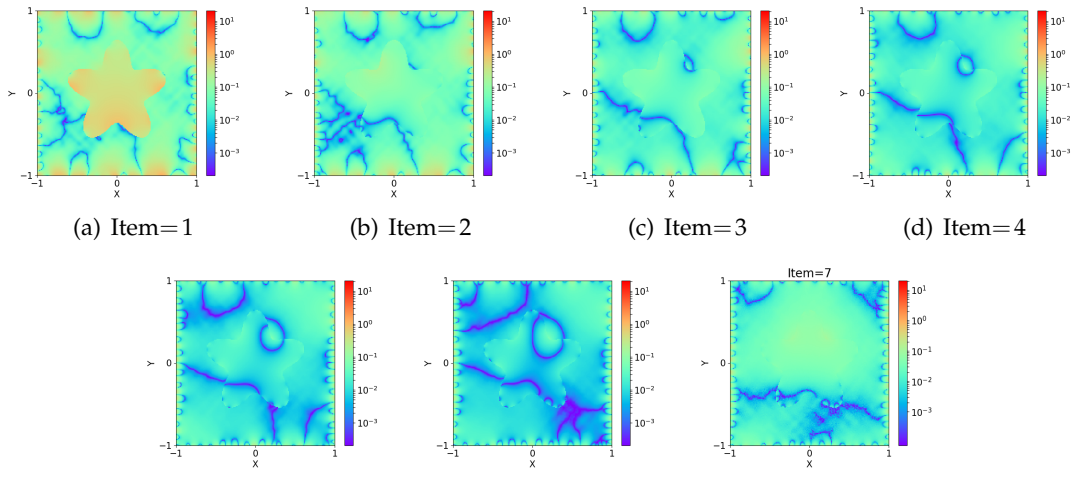
where $\Omega := [-0.5,0.5]\times[-0.5,0.5]$, $\Omega_1 := [-0.5,0]\times[-0.5,0.5]$, $\Omega_2 := [0,0.5]\times[-0.5,0.5]$, $\Gamma := \{(x,y)|x=0, y\in[-0.5,0.5]\}$, and

$$k = \begin{cases} k_1, & (x,y)\in\Omega_1, \\ k_2, & (x,y)\in\Omega_2. \end{cases}$$

We consider the following exact solution proposed in [20]:

$$u_* = \begin{cases} \sum_{i=1}^{d} (e^{ik_1 I_i^T x} - a_i e^{ik_1 R_i^T x}) & \text{in } \Omega_1, \\ \sum_{i=1}^{d} (1-a_i) e^{ik_2 T_i^T x} & \text{in } \Omega_2, \end{cases} \tag{3.7}$$

where $I_i = (\cos\theta_{I,i}, \sin\theta_{I,i})^T$, $R_i = (-\cos\theta_{I,i}, \sin\theta_{I,i})^T$ and $T_i = (\cos\theta_{T,i}, \sin\theta_{T,i})^T$ represent the directions of the incident, reflected and transmitted waves respectively. We have $k_2 \backslash k_1 = \sin\theta_{I,i} \backslash \sin\theta_{T,i}$ for each $0 < i < d$ and $a_i = \sin(\theta_{I,i} - \theta_{T,i}) \backslash \sin(\theta_{I,i} + \theta_{T,i})$. Here we set $d = 10$ and randomly generate $\{I_i\}$, $\{R_i\}$ and $\{T_i\}$.The boundary value g_1, g_2 is naturally computed by the exact solution 3.7. For the Helmholtz interface problem, we consider



(a) Item=1 (b) Item=2 (c) Item=3 (d) Item=4

(e) Item=5 (f) Item=6 (g) Item=7

Figure 19: Fitting errors in the whole domains for outer iterations given by HOrderDeepDDM (p=9) on problem (3.5). Here, the hyper-parameters are set as follows: $\alpha = 1$, $\alpha_2 = 200$, $L = 3$, $W = 200$, $N_r = N_b = N_{if} = 4096$.

the robust Robin-Robin DDM proposed in [3,5], which is more efficient than the Schwarz method. Thus, problem (3.6a)-(3.6d) can be written as follows:

$$\begin{cases} -\Delta u - k_1^2 u = 0 & \text{in } \Omega_1, \\ \dfrac{\partial u}{\partial \mathbf{n}} + ik_1 u = g_1 & \text{on } \partial\Omega_1 \backslash \Gamma, \\ \dfrac{\partial u}{\partial \mathbf{n}} + \gamma_1 u = h_1 & \text{on } \Gamma, \end{cases} \quad \begin{cases} -\Delta u - k_2^2 u = 0 & \text{in } \Omega_2, \\ \dfrac{\partial u}{\partial \mathbf{n}} + ik_2 u = g_2 & \text{on } \partial\Omega_2 \backslash \Gamma, \\ \dfrac{\partial u}{\partial \mathbf{n}} + \gamma_2 u = h_2 & \text{on } \Gamma, \end{cases} \tag{3.8}$$

where we naively set $\gamma_1 = \gamma_2 = 10(1+i)$, and a more appropriate choice of parameters would yield a lower error [3]. In addition, the interface terms in the loss functions (2.4a) and (2.4b) can be written in the following form:

$$\mathcal{M}_{\Gamma_1}(\theta) := \frac{1}{N_{if}} \sum_{j=1}^{N_{if}} \left| \frac{\partial h_p^1(x_{if}^j;\theta)}{\partial n} + \gamma_1 u_2^{i-1}(x_{if}^j) - h_1(x_{if}^j) \right|^2, \tag{3.9a}$$

$$\mathcal{M}_{\Gamma_2}(\theta) := \frac{1}{N_{if}} \sum_{j=1}^{N_{if}} \left| \frac{\partial h_p^2(x_{if}^j;\theta)}{\partial n} + \gamma_2 u_1^{i-1}(x_{if}^j) - h_2(x_{if}^j) \right|^2, \tag{3.9b}$$

with $\{x_{if}^j\}_{j=1}^{N_{if}}$ being the collocation points on Γ and $h_p^i(x;\theta)$ represents the approximate solution of HOrderDNN to u^i, $i=1,2$.

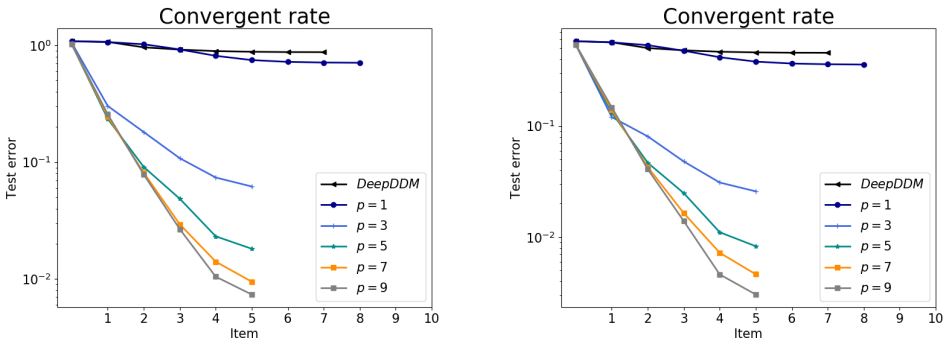
In this example, we set U to denote the real part of the solution and V to denote the imaginary part. DeepDDM and HOrderDeepDDMs (p) with $p=[1,3,5,7,9]$, $L=3$, $W=200$ are used to solve problem (3.6a)-(3.6d) and the total numbers of training points are chosen as $N_r=N_b=3072$, $N_{if}=1024$. The relative $\mathcal{L}^2$ errors of DeepDDM and HOrderDeepDDMs for $k_1=20$, $k_2=60$ and $k_1=60$, $k_2=100$ are shown in Table 6 and Table 7. Easy to see,

Table 6: The relative $\mathcal{L}^2$ errors and the number of outer iterations obtained by DeepDDM and HOrderDeepDDMs (p) on problem (3.6a)-(3.6d) with $k_1=20$, $k_2=60$. Here, $L=3$, $W=200$, $N_r=N_b=3072$, $N_{if}=1024$, err_U and $error_V$ represents the relative $\mathcal{L}^2$ error of U and V, see Eq. (3.1) for more details.

	DeepDDM	$p=1$	$p=3$	$p=5$	$p=7$	$p=9$
err_U	7.41E-1(8)	5.24E-1(9)	5.96E-2(7)	1.38E-2(7)	8.94E-3(6)	**5.82E-3(6)**
err_V	3.89E-1(8)	2.47E-1(9)	2.89E-2(7)	5.36E-3(7)	4.00E-3(6)	**2.87E-3(6)**

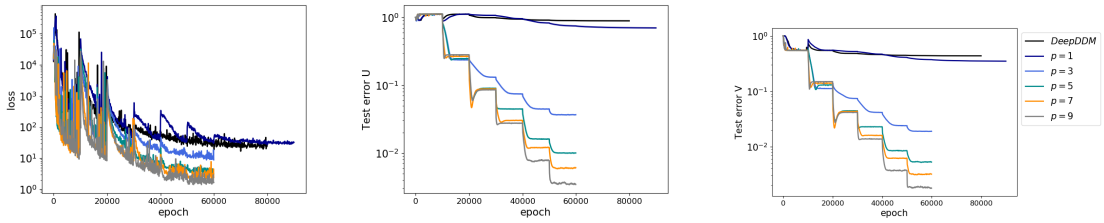
Table 7: The relative $\mathcal{L}^2$ errors and the number of outer iterations obtained by DeepDDM and HOrderDeepDDMs (p) on problem (3.6a)-(3.6d) with $k_1=60$, $k_2=100$. Here, high-frequency initial interface conditions are given for the outer iteration and the hyper-parameters are set as follows: $L=4$, $W=300$, $N_r=N_b=3072$, $N_{if}=1024$.

	DeepDDM	$p=1$	$p=5$	$p=7$	$p=9$	$p=11$
err_U	7.41E-1(8)	9.34E-1(8)	4.76E-1(8)	2.93E-1(8)	1.15E-1(8)	5.11E-2(8)
err_V	3.89E-1(8)	9.25E-1(8)	4.17E-1(8)	2.62E-1(8)	1.03E-1(8)	4.86E-2(8)

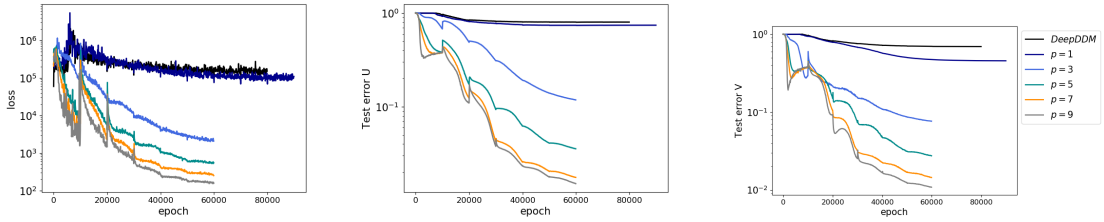


(a) U (b) V

Figure 20: The change in relative $\mathcal{L}^2$ error along with outer iterations for DeepDDM and HOrderDeepDDM on problem (3.6a)-(3.6d) with $k_1=20$, $k_2=60$. Here, the hyper-parameters are set as follows: $L=3$, $W=200$, $N_r=N_b=3072$, $N_{if}=1024$.



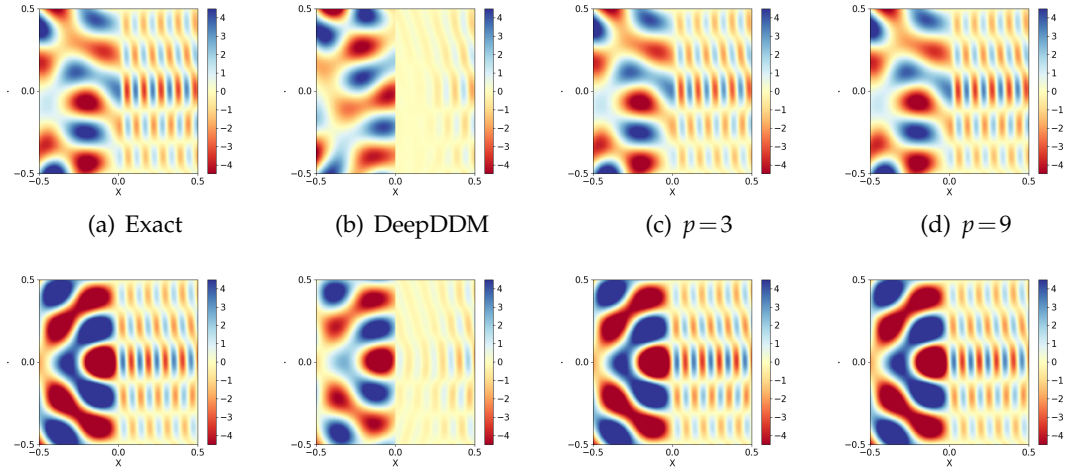
(a) Ω_1 loss (b) Ω_1 $error_U$ (c) Ω_1 $error_V$



(d) Ω_2 loss (e) Ω_2 $error_U$ (f) Ω_2 $error_V$

Figure 21: The convergence processes of losses and relative $\mathcal{L}^2$ errors in inner iterations on problem (3.6a)-(3.6d) with $k_1=20$, $k_2=60$. Here, the hyper-parameters are set as follows: $L=3$, $W=200$, $N_r=N_b=3072$, $N_{if}=1024$.

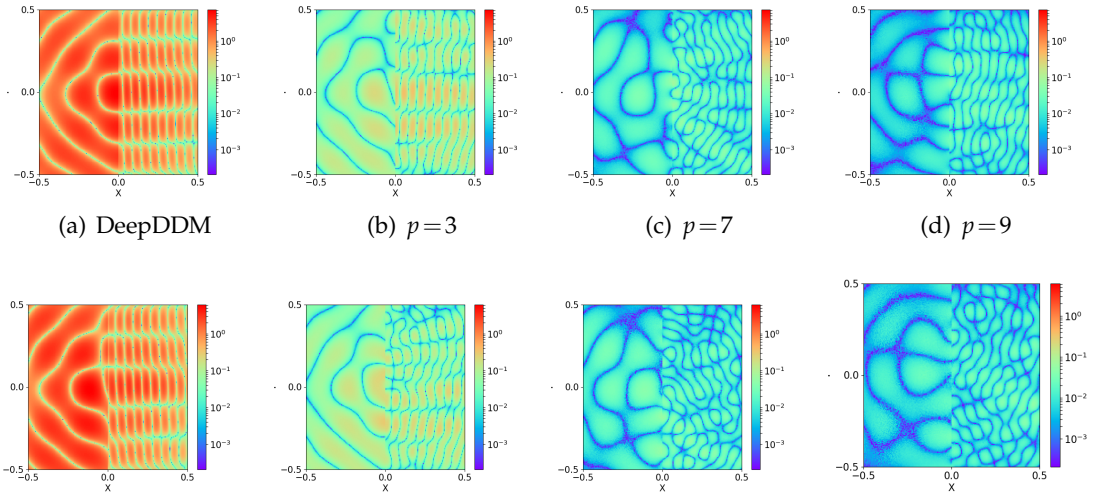
the errors of HOrderDeepDDMs ($p>1$) are much smaller than DeepDDM, and the error can be reduced from the order of 10^{-1} to 10^{-2} as p increases. What's more, the number of outer iterations decreases with increasing p, which demonstrates that the application of HOrderDNN has greatly accelerated the convergence speed of DDM. To more clearly reflect the acceleration effect brought by the high order, we show the process of error convergence along with the outer and inner iterations for $k_1=20$, $k_2=60$ in Fig. 20 and Fig. 21.



(a) Exact (b) DeepDDM (c) $p=3$ (d) $p=9$

(e) Exact (f) DeepDDM (g) $p=3$ (h) $p=9$

Figure 22: Exact and numerical solutions of DeepDDM and HOrderDeepDDMs with different p on problem (3.6a)-(3.6d) with $k_1=20$, $k_2=60$. The top row illustrates the fitting effect of the real part U, and the bottom row represents the fitting effect of the imaginary part V, respectively. Here, the hyper-parameters are set as follows: $L=3$, $W=200$, $N_r=N_b=3072$, $N_{if}=1024$.

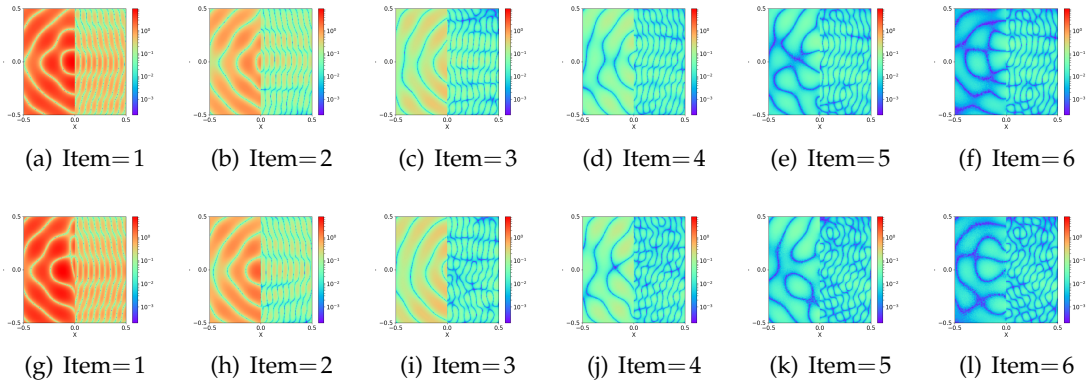


(a) DeepDDM (b) $p=3$ (c) $p=7$ (d) $p=9$

(e) DeepDDM (f) $p=3$ (g) $p=7$ (h) $p=9$

Figure 23: Fitting errors given by DeepDDM and HOrderDeepDDMs with different p on problem (3.6a)-(3.6d) with $k_1=20$, $k_2=60$. The top row (a)-(d) indicates the results of the real part U, and the bottom row (e)-(h) represents the results of the imaginary part V. The hyper-parameters are set as follows: $L=3$, $W=200$, $N_r=N_b=3072$, $N_{if}=1024$.

As expected, HOrderDeepDDMs ($p>1$) achieve smaller errors after fewer iterations than DeepDDM.



(a) Item=1 (b) Item=2 (c) Item=3 (d) Item=4 (e) Item=5 (f) Item=6

(g) Item=1 (h) Item=2 (i) Item=3 (j) Item=4 (k) Item=5 (l) Item=6

Figure 24: Fitting errors in the whole domains for outer iterations given by HOrderDeepDDM (p=9) on problem (3.6a)-(3.6d) with $k_1=20$, $k_2=60$. The top row (a)-(f) indicates the results of the real part U, and the bottom row (g)-(l) represents the results of the imaginary part V. Here, the hyper-parameters are set as follows: $L=3$, $W=200$, $N_r=N_b=3072$, $N_{if}=1024$.

In addition, the fitting effects of DeepDDM and HOrderDeepDDMs (p), as well as the point-by-points errors, are shown in Fig. 22 and Fig. 23. Same with the high-frequency elliptic problem (3.3a)-(3.3d), HOrderDeepDDMs ($p>1$) can easily capture the high-frequency information of the target function, and this ability increases as p increases. In the end, we show the point-by-point errors of HOrderDeepDDM ($p=9$) at each outer iteration in Fig. 24 to observe the convergence process of the error in the domain decomposition process. These phenomena show that the performance of HOrderDeepDDM is still much better than DeepDDM when dealing with high-frequency Helmholtz interface problems.

3.6 High-dimensional interface problem

Finally, to illustrate the capability of our method to handle high-frequency interface problems in high-dimensional space, we consider the high-frequency elliptic interface problem (3.3a)-(3.3d) in 3- and 5-dimensional spaces, where,

$$\Omega:=\left\{\mathbf{X}\in\mathcal{R}^d\middle|\sum_{i=1}^d x_i^2<1\right\},\quad \Gamma:=\left\{\mathbf{X}\in\mathcal{R}^d\middle|\sum_{i=1}^d x_i^2=0.25\right\},$$

$$\alpha(x)=\begin{cases}\alpha_1, & (x,y)\in\Omega_1:=\left\{\mathbf{X}\in\mathcal{R}^d\middle|\sum_{i=1}^d x_i^2<0.25\right\},\\ \alpha_2, & (x,y)\in\Omega_2:=\Omega\backslash\Omega_1.\end{cases}$$

Table 8: The relative $\mathcal{L}^2$ errors and the number of outer iterations obtained by DeepDDM and HOrderDeepDDMs (p) on problem (3.10) with $d=3$. The figures in parentheses are the numbers of outer iterations. Here, $L=1$, $W=300$, $N_r=N_b=N_{if}=1024$.

	DeepDDM	$p=1$	$p=3$	$p=5$	$p=7$
err	1.30E+0(6)	1.13E+0(8)	9.37E-2(7)	6.51E-2(6)	3.95E-2(6)

Table 9: The relative $\mathcal{L}^2$ errors and the number of outer iterations obtained by DeepDDM and HOrderDeepDDMs (p) on problem (3.10) with $d=5$. The figures in parentheses are the numbers of outer iterations. Here, $L=1$, $W=300$, $N_r=N_b=N_{if}=3125$.

	DeepDDM	$p=1$	$p=2$	$p=3$
err	9.89E-1(6)	7.75E-1(7)	6.91E-1(6)	1.06E-1(7)

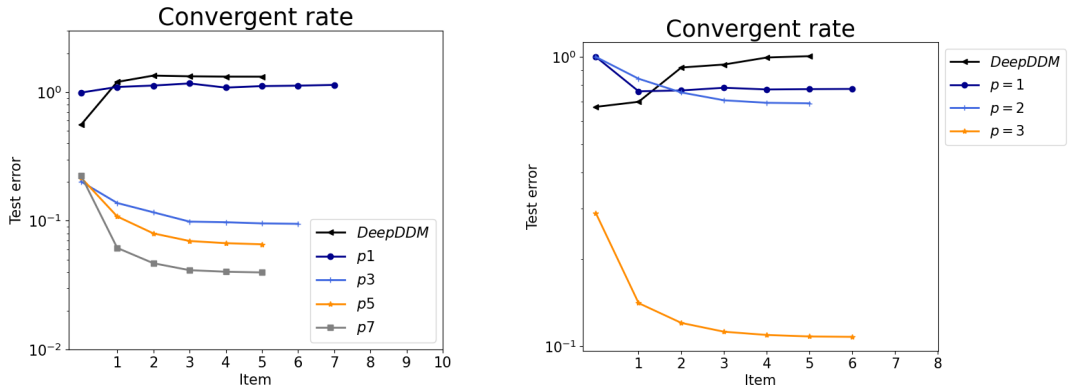
We consider the following high-frequency exact solution with discontinuities at the interface:

$$u_*(\mathbf{X})=\begin{cases}\sum_{i=1}^{d}\sin(k\cdot x_i), & \mathbf{X}\in\Omega_1,\\ \sum_{i=1}^{d}\cos(k\cdot x_i), & \mathbf{X}\in\Omega_2.\end{cases} \tag{3.10}$$

The boundary condition g and the interface conditions p,q can be directly computed from u_*.

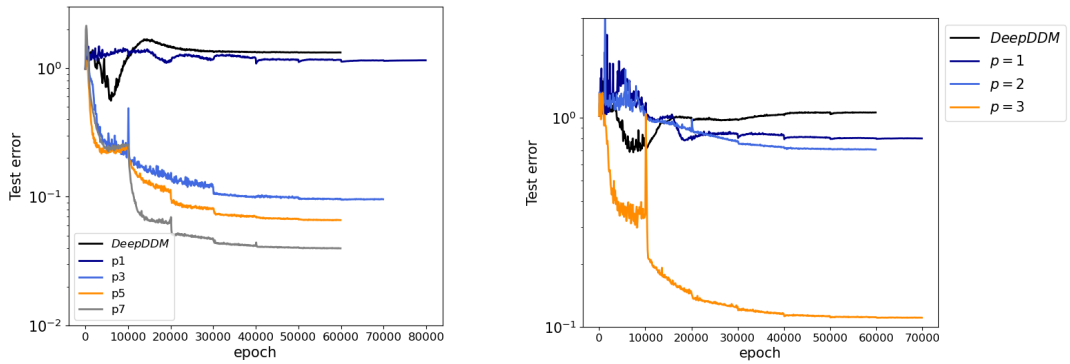
DeepDDM and HOrderDeepDDMs with $L=1$, $W=300$, $p=1,2,3,5,7$ are utilized to fit the numerical solution. The numbers of training points are set as $N_r=N_b=N_{if}=1024$ for 3-dimensional problem and $N_r=N_b=N_{if}=3125$ for 5-dimensional problem. Other hyperparameters are set as follows: $\alpha_1=1$, $\alpha_2=2$, $k=15$. The results are shown in Tables 8 and 9. Easy to see, HOrderDeepDDMs ($p>1$) outperform DeepDDM with the same architecture and the relative $\mathcal{L}^2$ errors decrease significantly as the order p increases, both for 3-dimensional problems and for more complex 5-dimensional problems. To demonstrate the accelerated convergence effect brought by the higher order, we show the convergence of the relative $\mathcal{L}^2$ error with the outer and inner iterations in Figs. 25 and 26. Obviously, the convergence speed of HOrderDeepDDM ($p\geq 3$) is much greater than that of DeepDDM in both 3-dimensional and 5-dimensional high-frequency interface problems, and the convergence speed is further accelerated as the order p increases.

We finally remark that HOrderDeepDDM, as an extension of HOrderDNN, may suffer the same curse of dimensionality as HOrderDNN. Indeed, the proposed HOrderDeepDDM aims at high-frequency interface problems such as wave problems, which usually reside in two or three dimensions in nature. For these problems, the proposed method is more flexible in handling complex interfaces when compared to mesh-dependent methods and has stronger approximation capability to characterize high-frequency components in the solution when compared to conventional DNNs.



(a) $d=3$ (b) $d=5$

Figure 25: The change in relative $\mathcal{L}^2$ error along with outer iterations for DeepDDM and HOrderDeepDDM on problem (3.10) with $d=3$ and $d=5$. Here, the hyper-parameters are set as follows: $\alpha_1=1$, $\alpha_2=2$, $L=1$, $W=300$, $N_r=N_b=N_{if}=1024,3125$ for $d=3,5$.



(a) $d=3$ (b) $d=5$

Figure 26: The convergence processes of relative $\mathcal{L}^2$ errors in inner iterations on problem (3.10) with $d=3$ and $d=5$. Here, the hyper-parameters are set as follows: $\alpha_1=1$, $\alpha_2=2$, $L=1$, $W=300$, $N_r=N_b=N_{if}=1024,3125$ for $d=3,5$.

4 Summary

We have introduced HOrderDeepDDM, which combines traditional DDM and HOrder-DNN to solve high-frequency interface problems. Using DDM, we overcome the difficulties caused by discontinuities at the interface, and by introducing HOrderDNN, we improve the ability of the neural network to approximate high-frequency information. We have demonstrated the effectiveness and superiority of HOrderDeepDDM in solv-

ing high-frequency interface problems through a series of numerical experiments for the solution of high-frequency interface PDEs. HOrderDeepDDMs with relatively large p far outperform DeepDDMs in terms of accuracy and efficiency. In particular, the minimum relative errors obtained by HOrderDeepDDMs ($p=9$) are one order of magnitude smaller than that obtained by DeepDDMs when the number of the parameters keeps the same. These results suggest that HOrderDeepDDM is an efficient and easily implemented method for solving high-frequency interface problems. Future work includes extending HOrderDeepDDM to more complex practical interface problems, giving theoretical guidance for determining p, and designing more efficient optimization algorithms.

Acknowledgements

The first and second authors contributed equally to this work. This research is supported partly by National Key R&D Program of China (grants Nos. 2019YFA0709600 and 2019YFA0709602), National Natural Science Foundation of China (grants Nos. 11831016 and 12101609) and the Innovation Foundation of Qian Xuesen Laboratory of Space Technology.

References

[1] TONY F. CHAN AND TAREK P. MATHEW, *Domain decomposition algorithms,* Acta Numer., 3 (1994), pp. 61–143.
[2] ZHIPENG CHANG, KE LI, XIUFEN ZOU, AND XUESHUANG XIANG, *High order deep neural network for solving high frequency partial differential equations,* Commun. Comput. Phys., 31(2) (2022), pp. 370–397.
[3] WENBIN CHEN, YONGXIANG LIU, AND XUEJUN XU, *A robust domain decomposition method for the helmholtz equation with high wave number,* Math. Model. Numer. Anal., 50 (2016), pp. 921–944.
[4] MAKSYMILIAN DRYJA AND OLOF B. WIDLUND, *Some domain decomposition algorithms for elliptic problems,* Iterative Methods for Large Linear Systems, pages 273–291. Elsevier, 1990.
[5] MARTIN J. GANDER, LAURENCE HALPERN, AND FRÉDÉRIC MAGOULES, *An optimized schwarz method with two-sided robin transmission conditions for the helmholtz equation,* Int. J. Numer. Methods Fluids, 55(2) (2007), pp. 163–175.
[6] HAILONG, GUO, AND XU YANG, *Deep unfitted nitsche method for elliptic interface problems,* Commun. Comput. Phys., 31(4) (2022), pp. 1162–1179.
[7] CUIYU HE, XIAOZHE HU, AND LIN MU, *A mesh-free method using piecewise deep neural network for elliptic interface problems,* J. Comput. Appl. Math., 412 (2022), p. 114358.
[8] WEI-FAN HU, TE-SHENG LIN, AND MING-CHIH LAI, *A discontinuity capturing shallow neural network for elliptic interface problems,* J. Comput. Phys., 469 (2022), 111576.
[9] AMEYA D. JAGTAP AND GEORGE EM KARNIADAKIS, *Extended physics-informed neural networks (xpinns): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations,* Commun. Comput. Phys., 28(5) (2020), pp. 2002–2041.

[10] AMEYA D. JAGTAP, EHSAN KHARAZMI, AND GEORGE EM KARNIADAKIS, *Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems,* Comput. Methods Appl. Mech. Eng., 365 (2020), p. 113028.

[11] DIEDERIK P. KINGMA AND JIMMY BA, *Adam: A method for stochastic optimization,* CoRR, abs/1412.6980, (2015).

[12] MING-CHIH LAI, CHE-CHIA CHANG, WEI-SYUAN LIN, WEI-FAN HU, AND TE-SHENG LIN, *A shallow ritz method for elliptic problems with singular sources,* J. Comput. Phys., 469 (2022), 111547.

[13] WUYANG LI, XUESHUANG XIANG, AND YINGXIANG XU, *Deep domain decomposition method: Elliptic problems,* Mathematical and Scientific Machine Learning, pages 269–286. PMLR, 2020.

[14] PIERRE-LOUIS LIONS ET AL., *On the schwarz alternating method i,* First International Symposium on Domain Decomposition Methods for Partial Differential Equations, volume 1, page 42. Paris, France, 1988.

[15] LUC MILLER, *Refraction of high-frequency waves density by sharp interfaces and semiclassical measures at the boundary,* Journal de Mathématiques Pures et Appliquées, 79(3) (2000), pp. 227–269.

[16] MAZIAR RAISSI, PARIS PERDIKARIS, AND GEORGE 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.

[17] HERMANN AMANDUS SCHWARZ, *Ueber einen Grenzübergang durch alternirendes Verfahren*, Zürcher u. Furrer, (1870).

[18] BARRY F. SMITH, PETTER E. BJØRSTAD, AND WILLIAM GROPP, *Domain decomposition: Parallel multilevel methods for elliptic partial differential equations,* Cambridge University Press, Cambridge, U.K. (1996), Comput. Math. Appl., 33 (1997), p. 132.

[19] SIFAN WANG, YUJUN TENG, AND PARIS PERDIKARIS, *Understanding and mitigating gradient flow pathologies in physics-informed neural networks,* SIAM J. Sci. Comput., 43(5) (2021), pp. A3055–A3081.

[20] WUYANG LI, ZIMING WANG, TAO CUI, XUESHUANG XIANG, AND YINGXIANG XU, *Deep domain decomposition methods: Helmholtz equation,* Adv. Appl. Math. Mech., Adv. Appl. Math. Mech., 15 (2023), pp. 118–138.

[21] ZHI-QIN JOHN XU, YAOYU ZHANG, TAO LUO, YANYANG XIAO, AND ZHENG MA, *Frequency principle: Fourier analysis sheds light on deep neural networks,* Commun. Comput. Phys., 28 (2020), pp. 1746–1767.

[22] WEINAN E AND BING YU, *The deep ritz method: a deep learning-based numerical algorithm for solving variational problems,* Commun. Math. Stat., 6 (2018), pp. 1–12.