



---

*Research article*

## **A deep learning framework for predicting the spread of diffusion diseases**

**Xiao Chen<sup>1</sup>, Fuxiang Li<sup>1</sup>, Hairong Lian<sup>1,\*</sup> and Peiguang Wang<sup>2,\*</sup>**

<sup>1</sup> School of Science, China University of Geosciences (Beijing), Beijing 100083, China

<sup>2</sup> College of Mathematics and Information Science, Hebei University, Baoding 071002, China

\* **Correspondence:** Email: [lianhr@cugb.edu.cn](mailto:lianhr@cugb.edu.cn), [pgwang@mail.hbu.edu.cn](mailto:pgwang@mail.hbu.edu.cn).

**Abstract:** In this paper, we considered a partitioned epidemic model with reaction-diffusion behavior, analyzed the dynamics of populations in various compartments, and explored the significance of spreading parameters. Unlike traditional approaches, we proposed a novel paradigm for addressing the dynamics of epidemic models: Inferring model dynamics and, more importantly, parameter inversion to analyze disease spread using Reaction-Diffusion Disease Information Neural Networks (RD-DINN). This method leverages the principles of hidden disease spread to overcome the black-box mechanism of neural networks relying on large datasets. Through an embedded deep neural network incorporating disease information, the RD-DINN approximates the dynamics of the model while predicting unknown parameters. To demonstrate the robustness of the RD-DINN method, we conducted an analysis based on two disease models with reaction-diffusion terms. Additionally, we systematically investigated the impact of the number of training points and noise data on the performance of the RD-DINN method. Our results indicated that the RD-DINN method exhibits relative errors less than 1% in parameter inversion with 10% noise data. In terms of dynamic predictions, the absolute error at any spatiotemporal point does not exceed 5%. In summary, we present a novel deep learning framework RD-DINN, which has been shown to be effective for reaction-diffusion disease modeling, providing an advanced computational tool for dynamic and parametric prediction of epidemic spread. The data and code used can be found at <https://github.com/yuanfanglila/RD-DINN>.

**Keywords:** compartmental model; epidemiology; inverse problem; neural networks; deep learning; Physics-Informed Neural Network (PINN)

---

### **1. Introduction**

The early spread dynamics of infectious diseases have never garnered as much attention as they do today. The use of mathematical models to understand the dynamics of epidemiology can be traced back to 1927 when Kermack first introduced the SIR epidemic model [1]. In a closed environment, we assume

that the total population size  $N$  remains constant and employ a set of ordinary differential equations (ODEs) to describe the mutual transitions among three categories of populations.  $S(t)$ ,  $I(t)$ ,  $R(t)$  are functions of time  $t$ , where  $S(t)$  represents susceptible individuals,  $I(t)$  represents infected individuals, and  $R(t)$  represents recovered individuals. The dynamics of these three populations can be expressed as the following equations

$$\begin{cases} \frac{dS}{dt} = -\beta SI, \\ \frac{dI}{dt} = \beta SI - \alpha I, \\ \frac{dR}{dt} = \alpha I. \end{cases} \quad (1.1)$$

The model in Eq (1.1) assumes that births and deaths are not allowed in a closed environment. Parameter  $\beta$  represents the disease transmission rate and  $\beta SI$  denotes the number of individuals infected with the disease per unit of time.  $\alpha$  represents the recovery rate, signifying the number of individuals recovering and transitioning from the infectious category  $I(t)$  to the recovered category  $R(t)$  per unit of time.

However, diseases do not initially spread uniformly across the region but also diffuse in space over time. Moreover, different diseases have different ways of spreading. For instance, the COVID-19 pandemic, reached 30 countries within 45 days, which saw the virus geographically spreading. Countries with a higher per-capita reported case counts tended to be highly globalized and had a significant degree of population mobility. Additionally, countries with fewer cases are mostly characterized by lower levels of globalization and domestic mobility. This implies that the disease gradually spreads, and the variation in the number of infections changes over time and space [2]. Using PDE to study diffuse disease models. Ndolane combines fractional derivative with diffusion term to study the dynamic behavior of fractional SIR model, which provides a new perspective for the application of fractional diffusion term in epidemic models [3].

Reaction-diffusion systems are employed to describe the spatial spread of infectious diseases. We assume that the transmission and recovery rates remain constant over time and space, described by the following partial differential equations

$$\begin{cases} \frac{\partial S}{\partial t} - d_1 \frac{\partial^2 S}{\partial x^2} = B - \beta SI - \mu S, & x \in \Omega, \quad t > 0, \\ \frac{\partial I}{\partial t} - d_2 \frac{\partial^2 I}{\partial x^2} = \beta SI - \alpha I - \mu I, & x \in \Omega, \quad t > 0, \\ \frac{\partial R}{\partial t} - d_3 \frac{\partial^2 R}{\partial x^2} = \alpha I - \mu R, & x \in \Omega, \quad t > 0, \\ \partial_\eta S = \partial_\eta I = \partial_\eta R = 0, & x \in \partial\Omega, \quad t > 0, \\ S(x, 0) = S_0(x), \quad I(x, 0) = I_0(x), \quad R(x, 0) = R_0(x), & x \in \bar{\Omega}, \end{cases} \quad (1.2)$$

where  $\Omega$  is a fixed and bounded domain with a smooth boundary  $\partial\Omega$  in  $\mathbb{R}^n$ ,  $\eta$  represents the outward unit normal vector on the boundary. The homogeneous Neumann boundary conditions imply that the considered diffusion SIR system is self-contained and has no migration across the boundary. In reality, populations always involve new births and deaths, so we consider births and deaths within the region  $\Omega$ .  $B$  denotes the number of newborn individuals and  $\mu$  represents the natural death rate. Here we assume the death rates are the same for each population category. The constants  $d_i$  ( $i = 1, 2, 3$ ) are diffusion coefficients representing the diffusion capabilities of each population category.  $S(x, t)$ ,  $I(x, t)$ ,  $R(x, t)$  are the population quantities dependent on time and space  $S(x, t) + I(x, t) + R(x, t) = N$ . Since the top

two equations in the system do not contain the recovered individuals  $R$ , we can analyze the dynamics without the third equation in the system.

Understanding the dynamics of disease spread and accurately estimating the transmission and recovery rates are crucial for epidemiological simulation models. However, accurately estimating these rates often require large amounts of expensive data [4]. Moreover, estimating parameters in most infectious disease models remains challenging because disease transmission is inherently stochastic and interdependent with other parameters [5].

As is well-known, traditional mathematical methods require only limited data to solve problems. These problems involve inferring the transmission dynamics of infectious diseases based on finite disease dynamic data, including the use of finite difference approximations [6]. Sindhu et al. discussed the application of classical statistical distributions [7] and parametric methods in medical data analysis [8,9]. However, traditional numerical methods are expensive and difficult to obtain data, and they especially perform poorly when dealing with noisy data. Therefore, there is an urgent need to explore new algorithms that can consider limited disease data while leveraging computational models to improve the accuracy, reliability, and predictability of disease dynamics and transmission parameters. Shafiq et al. compared the performance of artificial neural networks (ANN) and traditional parametric methods in epidemic data modeling [10], while combining maximum likelihood estimation (MLE) with ANN to explore breast cancer survival data analysis [11]. The scientific machine learning approach that combines neural networks with physics-informed modeling has emerged as a new tool in situations with limited or even no data. This in turn extends to the study of a wide range of problems in physics, biology, chemistry, and infectious diseases, including fluid dynamics, solid mechanics, and biochemistry, and the ecology of infectious diseases [12]. The Physics-Informed Neural Network (PINN) proposed by Raissi et al. proved to be effective in combining data with physical information [13]. Specifically, Physics-Informed Neural Network (PINN) has been widely used to solve inverse problems or ill-posed problems by incorporating partial differential equations (PDEs) models with sparse data [14]. By modeling nonlinear systems with neural networks and including PDEs in the loss function as a soft penalty term, the amount of required data is reduced. This is achieved using existing knowledge (PDE) to constrain the search space of the model. Yazdani et al. successfully embedded systems biology information into neural networks, creating Systems Biology-Informed Neural Networks (SBINN) [15]. Using relatively fewer experimental measurements, they inferred the dynamics of unobserved species and unknown model parameters. Raissi et al. initially applied the Physics-Informed Neural Network method to a standard SIR(ODE) infectious disease system to accurately estimate model parameters [16]. Subsequently, Shaier et al. extended the application of the Physics-Informed Neural Network method to various variants of SIR models, generating the unified Differential Information Neural Network (DINN) framework [17]. Yin et al. used a generalized partial differential approach to embed ODE disease mechanisms in neural networks, seamlessly coupling deep neural networks and dynamic systems and predicting disease states [18]. Difonzo et al. verified that even in the presence of challenging kernel functions, Physics-Informed Neural Network (PINN) can effectively solve inverse problems in periodic dynamics [19]. Difonzo et al. proposed applying radial basis functions (RBFs) as activation functions in suitably designed Physics-Informed Neural Network (PINN) to solve the inverse problem [20]. These frameworks effectively learn about disease transmission and provide estimates for unknown parameters.

Inspired by the Physics-Informed Neural Network (PINN) framework, we propose the RD-DINN method, which is a deep neural network embedded with reaction-diffusion disease information. This model embeds information about the diffusion disease into a neural network to accurately estimate the dynamics of different population categories and unknown parameters. We consider both the direct (forward) and the inverse problems. The direct problem involves inferring unknown dynamics at any time and location related with known initial and boundary conditions, while the inverse problem aims to deduce unknown parameters, dynamics, and model parameters with limited observed data. We analyze benchmark cases of SIR [21] and SEIR [22] infectious disease models with diffusion terms, demonstrating the robustness and reliability of our method through various network architectures and the addition of different levels of noise perturbations. We also examine the limitations and shortcomings of our proposed approach in terms of accuracy. The structure of this paper is as follows: In Section 2, we review essential background information on infectious diseases and the development of neural networks and Physics-Informed Neural Network (PINN). In Section 3, we introduce the technical approach of our proposed diffusion disease information neural network. Section 4 showcases the performance of our method on direct and inverse problems in two benchmark cases through computational experiments. In Section 5, we verify the feasibility of the model under real data on the new crown outbreak using five cities in Michigan in April 2020 as distributional conditions for the space  $x^*$ . Finally, in Section 6, we summarize and generalize our approach.

## 2. Background models and methods

In the field of mathematical biology, PDE has a better understanding and description of infectious disease propagation, epidemiological dynamics, and other related phenomena exhibiting diffusion behavior. However, when solving most coupled nonlinear PDE systems, it becomes quite challenging, and numerical solutions are the preferred approach. Typically, expensive commercial numerical solvers, such as finite difference method (FDM) [23] and finite element method (FEM). Moreover, numerical methods have some drawbacks, including high time consumption, repetitiveness, and a lack of autonomy. To better analyze the dynamic behavior of infectious diseases, it is crucial to build models that can quickly predict diseases. Easily, the SIR equation with a diffusion term and an initial function being positive, has only positive for all  $t \in (0, T_{max})$ . Therefore, there are two scenarios for the disease spread in the spatiotemporal domain: the disease disappears or persists, which leads to disease-free equilibrium and endemic equilibrium one. We also point out the concept of the basic reproduction number  $R_0$  [24]. In the diffusion SIR model,  $R_0$  is often expressed as  $R_0 = \frac{\beta B}{\mu(\mu + \alpha)}$  and serves as a crucial indicator measuring the potential of infectious disease transmission. It represents the average number of susceptible individuals per infected person per unit of time. If  $R_0$  is greater than 1, an outbreak is likely and otherwise the epidemic may gradually subside. In our two benchmark cases, considering births and deaths, it is expressed as

$$R_0 = \frac{\beta B}{\mu(\mu + \alpha)}. \quad (2.1)$$

**Theorem 2.1.** *Let  $\beta$  denote the infection rate,  $\alpha$  the recovery rate,  $B$  the birth rate, and  $\mu$  the natural mortality rate, where  $\beta, \alpha, B, \mu \in \mathbb{R} > 0$ . Then, the basic reproduction number  $R_0$  of the system at the*

---

\*<https://www.michigan.gov/coronavirus/stats/archived-coronavirus-data>



disease-free equilibrium (DFE) is given by:

$$R_0 = \frac{\beta B}{\mu(\mu + \alpha)}.$$

*Proof.* At the disease-free equilibrium, infected individuals  $I = 0$ . Thus, the system equation simplifies to

$$\frac{\partial S}{\partial t} - d_1 \frac{\partial^2 S}{\partial x^2} = B - \mu S. \quad (2.2)$$

At equilibrium, the time derivative is zero and, assuming a uniform spatial distribution, the spatial derivative is also zero. Therefore, the disease-free equilibrium point  $S^* = \frac{B}{\mu}$ . In the vicinity of the disease-free equilibrium, we can linearize the system. Let  $S = S^* + s$  and  $I = i$ , where  $s$  and  $i$  are small perturbations. By substituting these into the infection equation, ignoring the higher-order terms  $\beta si$ , we get

$$\frac{\partial i}{\partial t} - d_2 \frac{\partial^2 i}{\partial x^2} = \beta S^* i - (\alpha + \mu)i. \quad (2.3)$$

In the neighborhood of the disease-free equilibrium, the generation rate of infected individuals is  $\beta S^* i$  and the disappearance rate is  $(\alpha + \mu)i$ . Thus, the basic reproduction number  $R_0$  is the ratio of the generation rate to the disappearance rate

$$R_0 = \frac{\beta S^*}{\alpha + \mu} = \frac{\beta B}{\mu(\alpha + \mu)}. \quad (2.4)$$

The stability analysis of diseases and the estimation of the basic reproduction number are fundamental explanations for disease dynamics, playing a crucial role in predicting disease spread and formulating preventive measures. However, the ability to track computed values of various populations may vary with different parameter combinations, and there could be variations in real data for different demographic groups. Furthermore, all discussed methods require known initial subpopulation sizes.

With the advancement of science and technology, mathematical biology and epidemiology face a significant challenge in developing a coherent framework that enables them to integrate differential equations with the existing data. We introduce deep learning as a powerful approach, adopting a neural network architecture that mimics brain mechanisms [25]. The multi-layer neural network structure is the first perceptron layer that initially processes time and space through weighted sums and biases. Subsequently, it undergoes multiple layers of intricate decision-making until reaching the final decision layer. This decision layer generates outputs corresponding to the dynamic compartments  $S(x, t)$ ,  $I(x, t)$ , and  $R(x, t)$ . Deep neural networks seek to minimize the regression mean square error relative to weights and biases by incorporating diffusion terms  $\frac{\partial}{\partial x^2}$  into the neural network as PDE residuals through automatic differentiation during backpropagation [26] using software such as PyTorch, TensorFlow and incorporating gradient descent methods.

### 2.1. Finite difference method (FDM)

The finite difference method (FDM) approximates the partial derivatives in partial differential equations using differences, thereby converting a continuous problem into a discrete one. Discretizing space and time enables computing values at grid points via recursive updates.

**Theorem 2.2.** Consider the spatial interval  $x \in \Omega = [0, L]$  divided into  $N$  equidistant grid points with step size  $\Delta x$ , and time discretized into steps  $\Delta t$ . The spatial and temporal nodes are given by

$$x_i = i\Delta x, \quad i = 0, 1, 2, \dots, N, \quad t^n = n\Delta t, \quad n = 0, 1, 2, \dots \quad (2.5)$$

The time derivative is approximated using forward difference

$$\frac{\partial S}{\partial t} \approx \frac{S_i^{n+1} - S_i^n}{\Delta t}, \quad \frac{\partial I}{\partial t} \approx \frac{I_i^{n+1} - I_i^n}{\Delta t}. \quad (2.6)$$

The spatial second derivative is approximated using central difference

$$\frac{\partial^2 S}{\partial x^2} \approx \frac{S_{i+1}^n - 2S_i^n + S_{i-1}^n}{(\Delta x)^2}, \quad \frac{\partial^2 I}{\partial x^2} \approx \frac{I_{i+1}^n - 2I_i^n + I_{i-1}^n}{(\Delta x)^2}. \quad (2.7)$$

The no-flow boundary conditions  $\partial_\eta S = \partial_\eta I = 0$  are discretized as

$$\frac{S_1^n - S_0^n}{\Delta x} = 0 \Rightarrow S_1^n = S_0^n, \quad \frac{I_1^n - I_0^n}{\Delta x} = 0 \Rightarrow I_1^n = I_0^n. \quad (2.8)$$

The initial conditions  $S(x, 0) = S_0(x)$  and  $I(x, 0) = I_0(x)$  are discretized as

$$S_i^0 = S_0(x_i), \quad I_i^0 = I_0(x_i). \quad (2.9)$$

Using the discretized equations,  $S_i^{n+1}$  and  $I_i^{n+1}$  can be solved iteratively (e.g., explicit or implicit methods). Explicit methods require stability conditions (e.g., CFL condition), while implicit methods are more stable but require solving linear systems.

## 2.2. Physics-Informed Neural Network (PINN)

Physics-Informed Neural Network (PINN) is a data-driven algorithm, which was initially introduced as an alternative model for approximating the solution of differential equations and determining model parameters [27]. It utilizes any neural network architecture as the primary framework, such as Fully Connected Neural Networks (FNN) and Convolutional Neural Networks (CNN). The activation function and optimization methodology applied in Physics-Informed Neural Network (PINN) are the same as those of the usual deep learning techniques, and the most intriguing part of the algorithm is the loss function, which is unlike the traditional deep learning methods. Physics-Informed Neural Network (PINN) incorporates physical the residuals of differential equations in the loss function to make the output of the neural network to approximate the system of differential equations. Automatic differentiation is employed to perform derivative calculations with respect to input coordinates and discover underlying dynamics system laws.

### 2.2.1. Loss function

$$L_{\text{data}} = \sum_{i=1}^N |u_\theta(x_i, t_i) - u_i|^2, \quad L_{\text{physics}} = \sum_{i=1}^M \left| \frac{\partial u_\theta}{\partial t} - \mathcal{L}(u_\theta; \lambda) \right|^2. \quad (2.10)$$

Here,  $u_\theta(x_i, t_i)$  is the neural network's prediction at the data point  $(x_i, t_i)$ ,  $u_i$  is the true value at that point, and  $N$  is the total number of data points.  $\mathcal{L}(u; \lambda)$  is an operator expressed by PDE, and  $\lambda$  is the unknown parameters.  $M$  is the number of points used to calculate PDE residuals. The total loss is the weighted sum of the data fitting loss and the physical information loss.

$$L_{\text{total}} = \omega_1 L_{\text{data}} + \omega_2 L_{\text{physics}}, \quad (2.11)$$

where  $\omega_1$  and  $\omega_2$  are hyperparameters that govern the relative importance of data fitting loss and physical information loss in the total loss.

### 2.2.2. Parameter inversion

By minimizing the loss function  $L_{\text{total}}$ , the parameters  $\theta$  and  $\lambda$  of the neural network are optimized simultaneously.

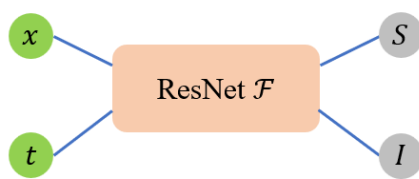
$$\theta^*, \lambda^* = \arg \min_{\theta, \lambda} L_{\text{total}}, \quad (2.12)$$

where  $\theta^*$  is the optimal parameter of the neural network and  $\lambda^*$  is the optimal parameter obtained by inversion. We use the Adam optimizer to optimize both  $\theta$  and  $\lambda$ .

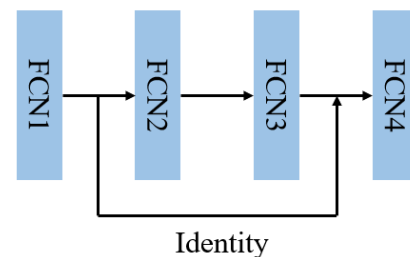
### 2.3. Residual connections in deep learning

Residual connections [28] have become a cornerstone of modern deep neural networks. These connections address the vanishing gradient problem by facilitating the flow of gradients through the network. Figure 1 shows the residual connection adds the input of a layer directly to its output, enabling the training of much deeper architectures. This approach not only enhances gradient propagation but also improves the overall performance and convergence of the model.

(a) Neural Networks



(b) ResNet Block



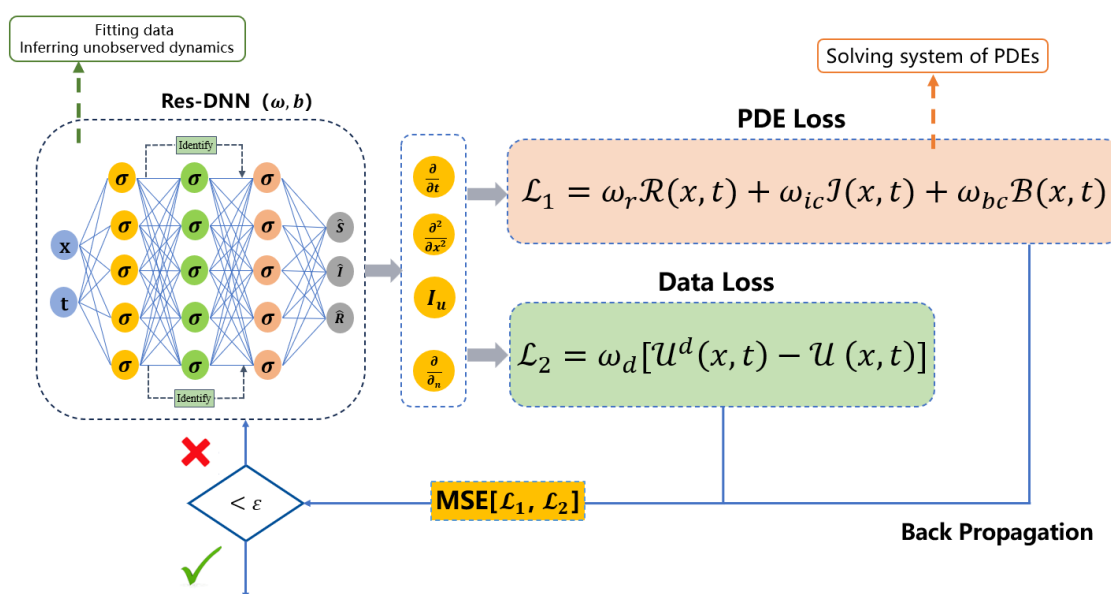
**Figure 1.** (a) Schematic of the neural networks and (b) adopted residual network architecture.

### 2.4. Reaction-Diffusion Disease Informed Neural Network (RD-DINN)

Inspired by the latest developments in Physics-Informed Deep Learning and the incorporation of hidden physics models, we propose the RD-DINN framework. This framework utilizes concealed disease information to spread information about diffusive diseases. It employs a fully connected neural network with two-dimensional input and multidimensional output to represent surrogate models for the partial differential equation solutions  $S(x, t)$ ,  $I(x, t)$ , and  $R(x, t)$ . The RD-DINN loss function

incorporates the encoded disease information model, serving as the computational model for infectious diseases introduced in Section 1. Of course, the embedded disease information could also be a variant of the SIR model, which we later validated with the SEIR model.

Drawing inspiration from the neural attention mechanism in computer vision tasks [29], we incorporate residual network modules in addition to fully connected layers, which ensure the preservation of the initial data structure and enhance the weight of hidden states to a significant extent while reducing errors. Figure 2 depicts the schematic diagram of the RD-DINN. We use the residual connection network for both forward and inverse problem experiments.



**Figure 2.** RD-DINN framework: On the leftmost side is a fully connected neural network with residual connections, where the inputs are observable data, and the unknown dynamics are inferred by the neural network; in the middle are some derivative terms and fitted data, originated from PyTorch automated micromultiplicity mechanism, and on the right side is the loss term consisting of two parts: The data loss of mismatches between the observable data and the predictions of the neural network and the loss of the residuals of the PDE, which are weighted and summed up. These two components are weighted and summed to fit the data by minimizing the loss function mean square error (MSE) and inferring the unobserved dynamics and the equation PDE coefficients.

In Figure refRD-DINN, Res-DNN represents a deep neural network with spatial variable  $x$  and temporal variable  $t$  as inputs, providing disease dynamics and transmission parameters as outputs. ‘PDE Loss’ denotes the residual loss of the disease model control equation with a diffusion term,  $\omega_r$ ,  $\omega_{ic}$ , and  $\omega_{bc}$  are the weight values of equation residuals, initial and boundary condition residuals in the loss function.  $\mathcal{R}(x, t)$  indicates that the output  $u(x, t)$  of neural network meets the dynamic behavior of PDE

$$\mathcal{R}(x, t) = \frac{\partial u}{\partial t} - \mathcal{L}(u), \quad (2.13)$$

where  $u$  is the prediction solution of neural network at  $\{x, t\}$ ,  $\mathcal{L}(u)$  is differential operator of PDE.  $\mathcal{I}(x, t)$  and  $\mathcal{B}(x, t)$  are the dynamic behavior of neural network prediction solution  $u(x, t)$ , satisfying initial and boundary conditions

$$\mathcal{I}(x, t) = u(x, 0) - u_0(x), \quad (2.14)$$

$$\mathcal{B}(x, t) = u|_{\partial\Omega} - u_b. \quad (2.15)$$

‘Data Loss’ accounts for the residual loss between the true values corresponding to randomly sampled points in the spatiotemporal domain in the inverse problem and the network predictions.  $\omega_d$  represents the weight of the data residual in the loss function of the neural network.  $\mathcal{U}^d(x, t) - \mathcal{U}(x, t)$  represents the difference between the predicted solution and the real solution of the neural network at  $\{x, t\}$ . We separate the direct and inverse problems as follows:

- For the direct problem, the loss function comprises three parts: PDE residual loss, initial value loss, and boundary loss.

- For the inverse problem, the loss function consists of two parts: The loss between sampled data true values and Res-DNN outputs, and PDE residual loss.

In all benchmark experiments, the Tanh activation function is utilized due to its smooth gradient properties, which effectively address nonlinear terms (e.g.,  $S * I$ ) in the equations and accurately capture complex nonlinear dynamics. Both direct and inverse problems are addressed by minimizing the loss function  $L$  using the Adam optimizer, which is employed with hyperparameters: Learning rate  $\alpha = 1e - 03$ , momenta  $(\beta_1, \beta_2) = (0.9, 0.999)$ , and  $\epsilon = 1e - 08$ . Convergence is achieved when the total loss falls below  $1e - 04$ . This configuration combines adaptive learning rates with momentum optimization to handle the non-convexity of the loss landscape effectively. The neural network fits the data and infers unobserved disease dynamics and transmission parameters by satisfying the PDE system.

### 3. RD-DINN resolves diffusion disease model dynamics and identifies equations

#### 3.1. Direct problem

Noting that parameters can generate the propagation systems of biological diseases, we restrict the parameters within a specific range near their actual values and within an acceptable margin of error. Therefore, we assume a diffusion disease model with transmission rate  $\beta$  and recovery rate  $\alpha$  obtained from the literature. For the forward problem, we consider known initial conditions  $S(x, 0), I(x, 0)$  and Neumann boundary conditions. We aim to obtain the solutions for the unknown compartments  $S(x, t), I(x, t)$ .

The next crucial step is to constrain the neural network to comply with the dispersed observed values of  $S$  and  $I$ , and the dynamical system of the control equation. It is necessary to incorporate terms corresponding to the observational data and partial differential equation to construct the loss function.

Let  $x \in [0, L]$ , the density distribution of the unknown compartments  $S(x, t)$  and  $I(x, t)$  in the spatiotemporal domain satisfies the partial differential equation system

$$\begin{cases} \frac{\partial S}{\partial t} - d_1 \frac{\partial^2 S}{\partial x^2} = B - \beta SI - \mu S, & x \in [0, L], \quad t > 0, \\ \frac{\partial I}{\partial t} - d_2 \frac{\partial^2 I}{\partial x^2} = \beta SI - \alpha I - \mu I, & x \in [0, L], \quad t > 0, \\ \partial_\eta S = \partial_\eta I = 0, & x = 0, L, \quad t > 0, \\ S(x, 0) = S_0(x), I(x, 0) = I_0(x), & x \in [0, L], \end{cases} \quad (3.1)$$

subject to the initial conditions

$$S(x, 0) = S_0(x), \quad I(x, 0) = I_0(x), \quad (3.2)$$

and Neumann boundary conditions

$$\begin{aligned} \frac{\partial S}{\partial x}(0, t) &= g_1^{(1)}(t), \quad \frac{\partial I}{\partial x}(0, t) = g_1^{(2)}(t), \\ \frac{\partial S}{\partial x}(L, t) &= g_2^{(1)}(t), \quad \frac{\partial I}{\partial x}(L, t) = g_2^{(2)}(t). \end{aligned} \quad (3.3)$$

Here, the Neumann boundary conditions represent the migration of different population groups at the boundary. We assume that the latent solution  $u(x, t)$  can be approximated by a deep neural network  $u_\theta(x, t)$  with parameters  $\theta$ . The parameters of the neural network  $u_\theta(x, t)$  can be estimated by minimizing the mean square loss function of the form

$$\mathcal{L}(\theta) = \mathcal{L}_f^{(1)}(\theta) + \mathcal{L}_f^{(2)}(\theta) + \mathcal{L}_{S_{ic}}(\theta) + \mathcal{L}_{I_{ic}}(\theta) + \mathcal{L}_{S_{bc}}(\theta) + \mathcal{L}_{I_{bc}}(\theta), \quad (3.4)$$

where the PDE residual  $\mathcal{L}_f(\theta)$ , the boundary residual  $\mathcal{L}_{bc}(\theta)$ , and initial residual  $\mathcal{L}_{ic}(\theta)$  can be defined as follows:

$$\begin{aligned} \mathcal{L}_f^{(1)}(\theta) &= \frac{1}{N} \sum_{i=1}^N \left| \frac{\partial S_\theta}{\partial t}(x_f^i, t_f^i) - d_1 \frac{\partial^2 S_\theta}{\partial x^2}(x_f^i, t_f^i) - B + \beta S_\theta(x_f^i, t_f^i) I(x_f^i, t_f^i) + \mu S_\theta(x_f^i, t_f^i) \right|^2, \\ \mathcal{L}_f^{(2)}(\theta) &= \frac{1}{N} \sum_{i=1}^N \left| \frac{\partial I_\theta}{\partial t}(x_f^i, t_f^i) - d_2 \frac{\partial^2 I_\theta}{\partial x^2}(x_f^i, t_f^i) - \beta S_\theta(x_f^i, t_f^i) I_\theta(x_f^i, t_f^i) + \alpha I_\theta(x_f^i, t_f^i) + \mu I_\theta(x_f^i, t_f^i) \right|^2, \\ \mathcal{L}_{S_{ic}}(\theta) &= \frac{1}{N_{ic}} \sum_{i=1}^{N_{ic}} |S_\theta(x_d^i, 0) - S_0(x_d^i)|^2, \\ \mathcal{L}_{I_{ic}}(\theta) &= \frac{1}{N_{ic}} \sum_{i=1}^{N_{ic}} |I_\theta(x_d^i, 0) - I_0(x_d^i)|^2, \\ \mathcal{L}_{S_{bc}}(\theta) &= \frac{1}{N_{bc}} \sum_{i=1}^{N_{bc}} \left| \frac{\partial S_\theta}{\partial x}(0, t_d^i) - g_1^{(1)}(t_d^i) \right|^2 + \left| \frac{\partial S_\theta}{\partial x}(L, t_d^i) - g_1^{(2)}(t_d^i) \right|^2, \\ \mathcal{L}_{I_{bc}}(\theta) &= \frac{1}{N_{bc}} \sum_{i=1}^{N_{bc}} \left| \frac{\partial I_\theta}{\partial x}(0, t_d^i) - g_2^{(1)}(t_d^i) \right|^2 + \left| \frac{\partial I_\theta}{\partial x}(L, t_d^i) - g_2^{(2)}(t_d^i) \right|^2. \end{aligned}$$

Here,  $\{x_d^i, t_d^i\}_{i=1}^{N_{ic, bc}}$  denotes the random sampled coordinates of the initial and boundary conditions, and  $\{x_f^i, t_f^i\}_{i=1}^N$  are a set of spatial-temporal points random sampled within the computational domain  $[0, L] \times [0, T]$  for enforcing the partial differential equation. Here, we employ Latin hypercube sampling as our random sampling method to ensure full coverage of the sampling point range [30]. The interval  $[x_{min}, x_{max}]$  is divided into  $N_{ic}$  layers, with one random point per layer selected to define initial values  $S(x_i, t_0)$ ,  $I(x_i, t_0)$  at  $\{(x_i, t_0)\}_{i=1}^{N_{ic}}$ . The interval  $[t_0, t_{end}]$  is divided into  $N_{bc}$  layers, with one random point per layer selected to define boundary values at  $\{(x_{boundary}, t_j)\}_{j=1}^{N_{bc}}$  for both  $x_{min}$  and  $x_{max}$ .

In the benchmark experiments, reference solutions for the partial differential equations were computed using a high-order finite difference method. For the SIR model, datasets were generated by systematically varying the parameters  $\{\beta, \alpha\} \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ , while for the SEIR model,

parameter sets  $\{\beta, \sigma, \alpha\} \in \{0.6, 0.8, 1.0, 1.2, 1.4\}$  were employed. A comprehensive sensitivity analysis was conducted across these parameter configurations to rigorously evaluate the robustness of the proposed RD-DINN method under diverse conditions.

To demonstrate the reliability of our algorithm, let's focus on two specific benchmark examples. The parameters for these experiments are given in Table 1.

**Table 1.** Parameters in the two benchmark infectious disease experiments.

Parameter	Notation	Value	
		Example 1	Example 2
Recruitment rate	$B$	0.1	1.0
Natural death rate	$\mu$	0.1	0.1
Infection rate	$\beta$	0.5	1.0
Recovery rate	$\alpha$	0.5	1.0
Latent rate	$\sigma$	0.0	1.0
S-diffusion capacity	$d_1$	0.001	0.005
E-diffusion capacity	$d_2$	0.000	0.005
I-diffusion capacity	$d_3$	0.001	0.005
R-diffusion capacity	$d_4$	0.001	0.005

### 3.1.1. Direct SIR model

This benchmark experiment is for the direct problem. We start by considering the SIR model with diffusion

$$\begin{cases} \frac{\partial S(x,t)}{\partial t} - d_1 \frac{\partial^2 S}{\partial x^2} = B - \beta S(x,t)I(x,t) - \mu S(x,t), \\ \frac{\partial I(x,t)}{\partial t} - d_2 \frac{\partial^2 I}{\partial x^2} = \beta S(x,t)I(x,t) - \alpha I(x,t) - \mu I(x,t), \end{cases} \quad (3.5)$$

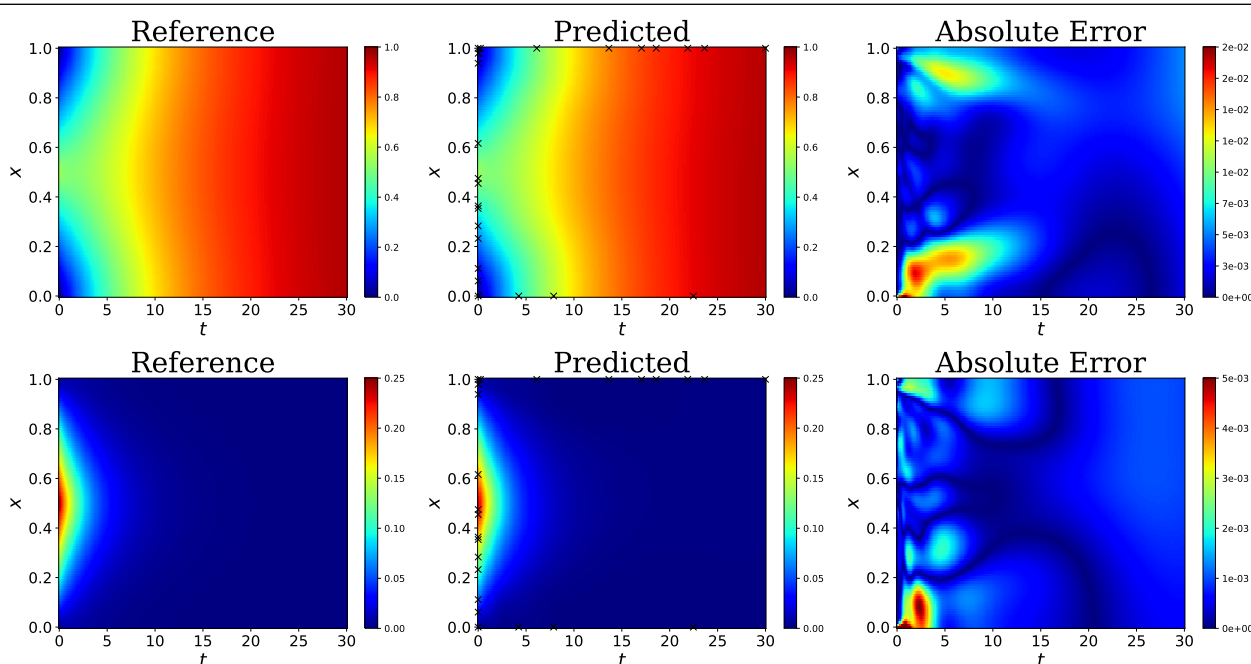
subject to the initial conditions

$$S(x,0) = \begin{cases} 1.1x, & 0 \leq x < 0.5, \\ 1.1(1-x), & 0.5 \leq x \leq 1, \end{cases} \quad I(x,0) = \begin{cases} 0.5x, & 0 \leq x < 0.5, \\ 0.5(1-x), & 0.5 \leq x \leq 1, \end{cases} \quad (3.6)$$

and the Neumann boundary conditions

$$\frac{\partial S(x,t)}{\partial \eta} = \frac{\partial I(x,t)}{\partial \eta} = 0, \quad (x,t) \in [0,1] \times [0,30]. \quad (3.7)$$

Here,  $\eta$  is the unit normal vector. Boundary condition equal to 0 means the domain with no population migration in and out of the boundary.  $d_1$  and  $d_2$  represent the diffusion capabilities of susceptible and infected individuals.  $\beta$  represents the transmission rate, and  $\alpha$  represents the recovery rate. We assume that these coefficients remain constant. Using the RD-DINN method, we predict the latent solution of the equation. The results are given by Figure 3.



**Figure 3.** In order from top to bottom  $S(x, t)$ ,  $I(x, t)$ . Left: Reference solution of SIR equation using the finite difference solver. Middle: Prediction solution of SIR using 6-layer RD-DINN. Right: Absolute point-wise error.

We can observe that when  $t$  near zero and  $x$  near the boundary, the RD-DINN method exhibits a slightly larger deviation compared to the traditional finite difference method, especially at  $(x, t) \in [0, 10] \times [0.25, 1.0]$ . It is reasonable to have a slightly larger deviation due to the abrupt change in population at the early stage of the infectious disease outbreak, as predicting the spread of the disease is challenging during the initial phase. RD-DINN produces excellent predictive solutions, with a relative  $L_2$  error of 3%.

### 3.1.2. Direct SEIR model

To demonstrate the reliability and reproducibility of our proposed method, the second case selects the SEIR infectious disease model with an added incubation period. The exposed population  $E(x, t)$  refers to individuals who have been infected with the virus but have not yet developed symptoms or transmitted the virus to other individuals. Generally, researchers consider the latent period and incubation period as the same concept, with an average latent period of 1.9 days [31]. Ndolane studied the numerical method of fractional SEIR model, and discussed the potential application of fractional derivative in spatial propagation model. Fractional derivatives can be combined with diffusion term to describe spatial heterogeneity [32]. The SEIR model is described by the following equations

$$\begin{cases} \frac{\partial S(x, t)}{\partial t} - d_1 \frac{\partial^2 S}{\partial x^2} = B - \beta S(x, t)I(x, t) - \mu S(x, t), \\ \frac{\partial E(x, t)}{\partial t} - d_2 \frac{\partial^2 E}{\partial x^2} = \beta S(x, t)I(x, t) - (\mu + \sigma)E(x, t), \\ \frac{\partial I(x, t)}{\partial t} - d_3 \frac{\partial^2 I}{\partial x^2} = \sigma E(x, t) - (\alpha + \mu)I(x, t), \end{cases} \quad (3.8)$$

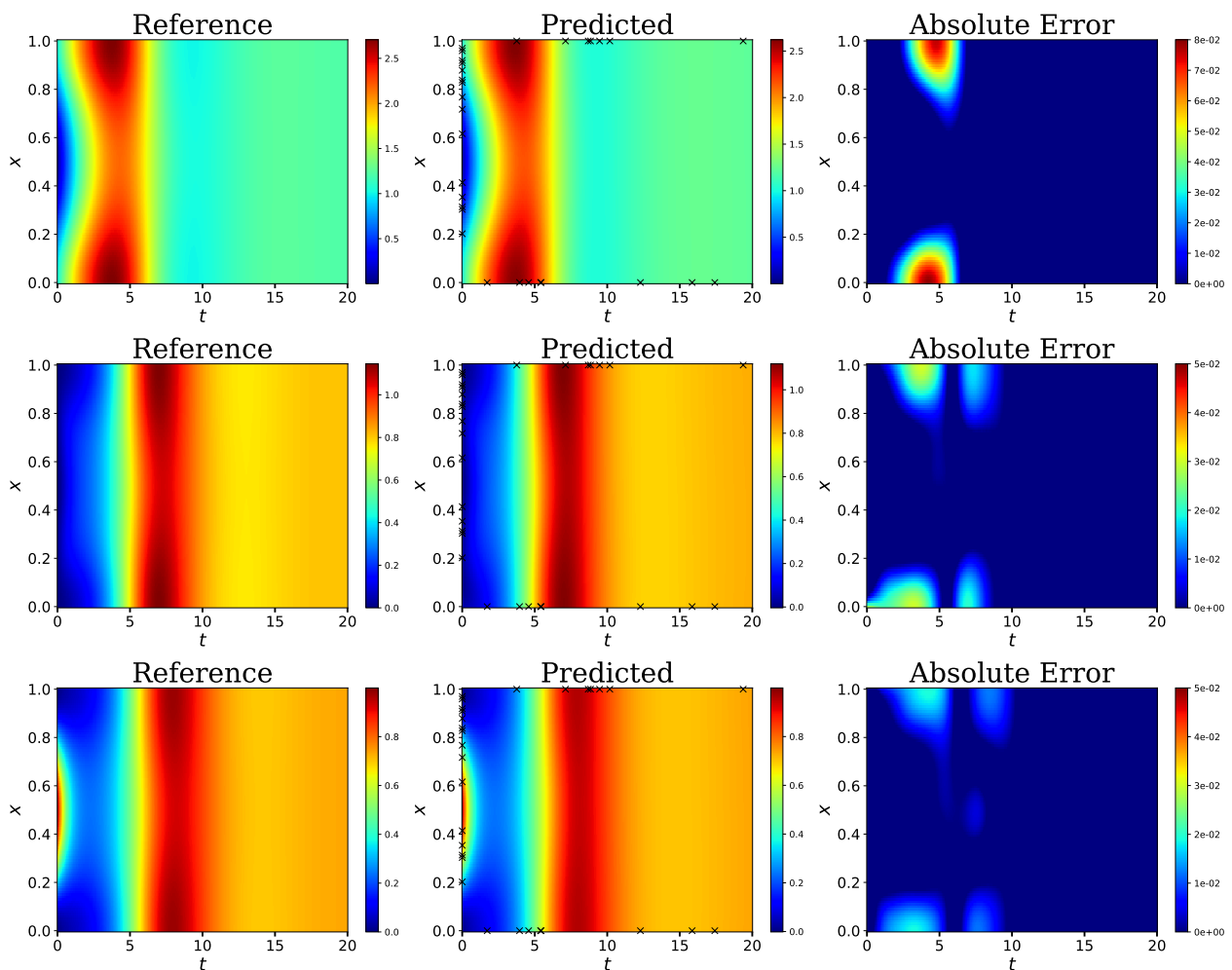


where the recovered population  $R$  are not involved. The initial conditions and Neumann boundary conditions are given by

$$S(x, 0) = 1 - \sin^2(\pi x), \quad I(x, 0) = \sin^2(\pi x), \quad E(x, 0) = 0, \quad R(x, 0) = 0, \quad x \in [0, 1], \quad (3.9)$$

$$\frac{\partial S(x, t)}{\partial \eta} = \frac{\partial E(x, t)}{\partial \eta} = \frac{\partial I(x, t)}{\partial \eta} = \frac{\partial R(x, t)}{\partial \eta} = 0, \quad (x, t) \in [0, 1] \times [0, 20], \quad (3.10)$$

where  $\eta$  is the unit normal vector. That Neumann boundary conditions equal to 0 indicates no migration of population at the boundary. Parameters  $B, \mu$  are consistent with those in the SIR model.  $\beta$  represents the disease latent rate,  $\sigma$  represents the latent exposure rate, and  $\alpha$  represents the disease recovery rate.



**Figure 4.** In order from top to bottom  $S(x, t), E(x, t), I(x, t)$ . Left: SEIR equation: Reference solution of SEIR using a conventional finite difference solver. Middle: Prediction of 6-layer RD-DINN using initial, boundary, and partial differential equations. Right: absolute point-wise error.

Figure 4 shows a schematic diagram of SEIR model prediction. Based on these two disease models mentioned above, we can easily observe that the RD-DINN method, leveraging only a minimal number

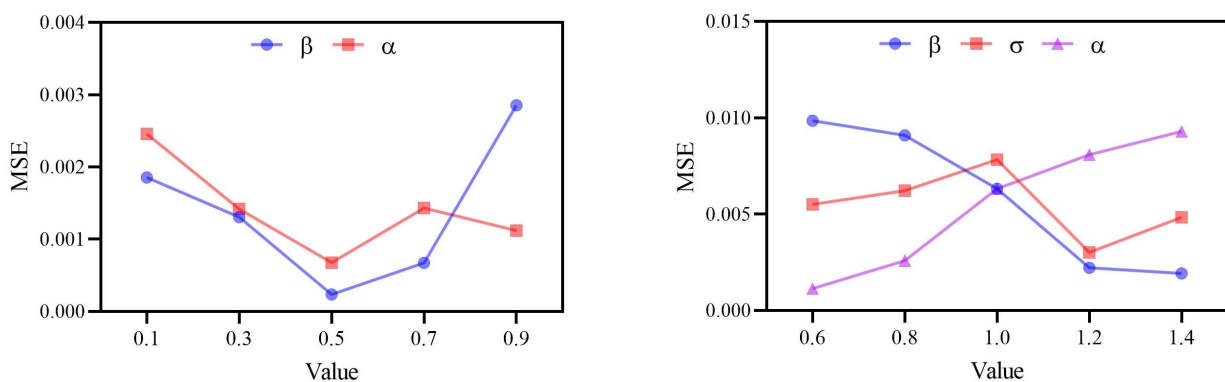
of initial and boundary data points, along with the selective use of limited data coordinates within the domain to enforce the disease control equations, can effectively predict the unknown dynamics of infectious diseases. The initial and boundary conditions of the model are unknown at the beginning in most cases, which requires inverse problem, leveraging only a small portion of data and incompletely known control equations to find suitable model parameters for modeling.

### 3.1.3. Sensitivity analysis

Sensitivity analysis is an important tool for assessing the sensitivity of model outputs to changes in input parameters. Small changes in parameters such as infection rate  $\beta$ , latency rate  $\sigma$ , and recovery rate  $\alpha$  may lead to significant differences in system behavior.  $\frac{\partial}{\partial x}$  describes spatial heterogeneity, and  $\{\beta, \sigma, \alpha\}$  are the parameters to be analyzed. We perform individual parameter sensitivity analyses by assigning a specific value to each target parameter  $p \in \{\beta, \sigma, \alpha\}$  and keeping the other parameters as defaults. The loss  $\mathcal{L}$  is minimized using PINN, the MSE between the predicted and true solution on the test set is calculated and the sensitivity is quantified by analyzing the curve of MSE as a function of parameter  $p$

$$\text{MSE} = \frac{1}{K} \sum_{k=1}^K (\hat{Q}^{(k)} - Q^{(k)})^2, \quad (3.11)$$

where  $K$  is the total number of data points in the test set,  $\hat{Q}^{(k)}$  and  $Q^{(k)}$  are the predicted and true observed values of the model for the  $k$ th sample, respectively.



**Figure 5.** Left: Fix the SIR models  $\{\beta, \alpha\} = 0.5$  separately and calculate the MSE of another parameter within the range of 0.1–0.9. Right: The MSE of two parameters in the fixed SEIR model  $\{\beta, \sigma, \alpha\} = 1.0$  and the other parameter within the range of 0.6–1.4.

In Figure 5, the SIR model maintains MSE consistently below 0.003 across parameter ranges, while the SEIR model maintains MSE under 0.01. This demonstrates the robustness of the RD-DINN framework under parameter variations, where its residual-driven mechanism and integration of prior knowledge mitigate sensitivity effects and enhance adaptability to parameter uncertainties.

### 3.2. Inverse problem

In this section, we consider the inverse problem to identify the unknown parameters  $(\beta, \alpha)$  contained in the equations and infer each compartment's unknown dynamics. Specifically, given a small amount of measured values  $\{S(x_d^i, t_d^i), I(x_d^i, t_d^i)\}_{i=1}^N$ , which are dispersed and may contain noise, We wish to infer a solution  $S(x, t), I(x, t)$  in the region  $\Omega$  that satisfies the diffusion disease model, and determine the unknown parameters of the model. It is worth emphasizing that for any given data point  $\{S(x_d^i, t_d^i), I(x_d^i, t_d^i)\}_{i=1}^N$ , the disease information conditions that are met during training are unknown.

So what are the parameters  $\beta, \alpha$  that best describe the observed data? To answer this question, we apply RD-DINN to infer the unknown parameters  $\beta, \alpha$ . Importantly, no initial or boundary conditions are required, which is close to many practical applications where obtaining such information is challenging. We conduct the inverse analysis of two diffusion disease models, assuming that the spatiotemporal domain is subject to dispersed and latent noisy measurements. We explore different network architectures and varying noise levels to demonstrate the robustness and reliability of the RD-DINN in terms of accuracy. To achieve this, we employ a deep neural network to approximate the latent solutions  $S(x, t), I(x, t)$  and the unknown parameters  $\beta, \alpha$ . The training parameters for both are shared, and by minimizing the composite loss below, we train a disease information neural network model that can simultaneously determine unknown parameters and unknown dynamics.

$$\mathcal{L}(\theta, \beta, \alpha) = \lambda_{f_1} \mathcal{L}_f^{(1)}(\theta, \beta, \alpha) + \lambda_{f_2} \mathcal{L}_f^{(2)}(\theta, \beta, \alpha) + \lambda_{d_1} \mathcal{L}_d^{(1)}(\theta) + \lambda_{d_2} \mathcal{L}_d^{(2)}(\theta), \quad (3.12)$$

$$\mathcal{L}_f^{(1)}(\theta, \beta, \alpha) = \frac{1}{N} \sum_{i=1}^N \left| \frac{\partial S_\theta}{\partial t}(x_d^i, t_d^i) - d_1 \frac{\partial^2 S_\theta}{\partial x^2}(x_d^i, t_d^i) + \beta_\theta S_\theta(x_d^i, t_d^i) I_\theta(x_d^i, t_d^i) \right|^2,$$

$$\mathcal{L}_f^{(2)}(\theta, \beta, \alpha) = \frac{1}{N} \sum_{i=1}^N \left| \frac{\partial I_\theta}{\partial t}(x_d^i, t_d^i) - d_2 \frac{\partial^2 I_\theta}{\partial x^2}(x_d^i, t_d^i) - \beta_\theta S_\theta(x_d^i, t_d^i) I_\theta(x_d^i, t_d^i) + \alpha_\theta I_\theta(x_d^i, t_d^i) \right|^2,$$

$$\mathcal{L}_d^{(1)}(\theta) = \frac{1}{N} \sum_{i=1}^N |S_\theta(x_d^i, t_d^i) - S(x_d^i, t_d^i)|^2,$$

$$\mathcal{L}_d^{(2)}(\theta) = \frac{1}{N} \sum_{i=1}^N |I_\theta(x_d^i, t_d^i) - I(x_d^i, t_d^i)|^2.$$

Here,  $\{x_d^i, t_d^i\}_{i=1}^N$  represents a set of spatial and temporal locations randomly sampled in  $[0, L] \times [0, T]$  during each training iteration. The corresponding population values  $S(x_d^i, t_d^i), I(x_d^i, t_d^i)$  serve as training datas. We enforce these training datas to the structure dictated by the PDE control equations. The number and position of these location points are identical to the training datas.

To validate the effectiveness of the method, we randomly created a training dataset with  $N = 3000$  points based on numerical solutions corresponding to  $\beta$  and  $\alpha$ . Subsequently, a 6-layer deep neural network with 50 neurons in each hidden layer was trained using this dataset. The Adam optimizer is utilized to minimize the loss function. After training completion, the network was fine-tuned to predict the potential solutions  $S(x, t)$  and  $I(x, t)$  and the unknown parameters  $\beta, \alpha$ . In the following two benchmark cases, we provide an intuitive evaluation of the accuracy of the Diffusion Disease Information Neural Network. Even in the presence of 5%, 10% non-correlated noisy data interference, the network still accurately identifies the unknown parameters and dynamics of the diffusion disease model.

To further evaluate the algorithm performance, we systematically studied the total amount of training data, neural network architecture, and different noise concentrations. The main observation here is that the method exhibits strong robustness to the noise level in the data, achieving reasonable identification accuracy even with noise levels as high as 10%. This enhanced robustness seems to outperform competing methods such as Gaussian process regression and methods relying on sparse regression [33].

### 3.2.1. Inverse SIR model

In the inverse problem, two benchmark cases are considered from the forward problem. The difference lies on the fact that the disease control equations, and the initial and boundary conditions are unknown. Only a tiny, scattered, and possibly noisy dataset is given for the hidden states  $S(x, t)$  and  $I(x, t)$  to infer the unknown parameters and the unknown dynamics of the equation. The form of the model control equation is given by

$$\begin{cases} \frac{\partial S(x, t)}{\partial t} - d_1 \frac{\partial^2 S}{\partial x^2} = B - \lambda_1 S(x, t)I(x, t) - \mu S(x, t), \\ \frac{\partial I(x, t)}{\partial t} - d_2 \frac{\partial^2 I}{\partial x^2} = \lambda_1 S(x, t)I(x, t) - \lambda_2 I(x, t) - \mu I(x, t), \end{cases} \quad (3.13)$$

define  $f(x, t)$  and  $g(x, t)$  with

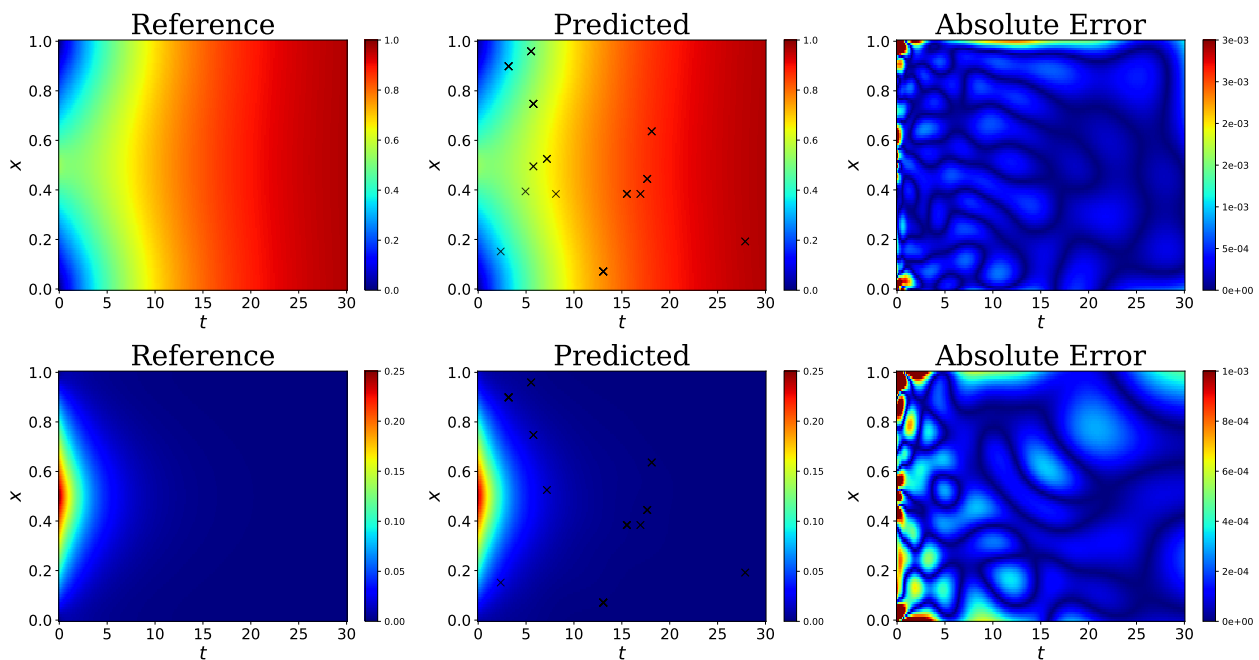
$$\begin{aligned} f &:= \frac{\partial S(x, t)}{\partial t} - d_1 \frac{\partial^2 S}{\partial x^2} - B + \lambda_1 S(x, t)I(x, t) + \mu S(x, t), \\ g &:= \frac{\partial I(x, t)}{\partial t} - d_2 \frac{\partial^2 I}{\partial x^2} - \lambda_1 S(x, t)I(x, t) + \lambda_2 I(x, t) + \mu I(x, t). \end{aligned}$$

Consider the epidemiological parameters  $(\beta, \alpha) = (\lambda_1, \lambda_2)$  be unknown, while  $d_1, d_2, B$ , and  $\mu$  are known. The solutions  $S(x, t), I(x, t)$  are predicted through the RD-DINN method, and  $\lambda_1$  and  $\lambda_2$  are shared parameters with the neural network during training. The predicted solutions, reference solutions, and the pointwise absolute error between them are shown in Figure 6. The RD-DINN method accurately learns the latent solutions  $S(x, t), I(x, t)$  of the unknown disease system. The  $L_2$  error for  $S$  is  $5.74e - 04$ , and  $I$  is  $7.89e - 03$ .

Figure 6 illustrates the dynamics of the DSIR model across the spatiotemporal domain. The basic regeneration number  $R_0 = \frac{\beta B}{\mu(\mu + \alpha)} \approx 0.83 < 1$  under this set of parameters means that the average number of people infected per infected person is less than 1. The spread of the disease will tend to decline. As infected individuals cannot sustain the transmission of the infection, the disease eventually fades away in the population. It can be observed that starting from the disease outbreak, the number of infected individuals  $I(x, t)$  gradually approaches 0, signifying the disappearance of the disease by  $t = 10$ .

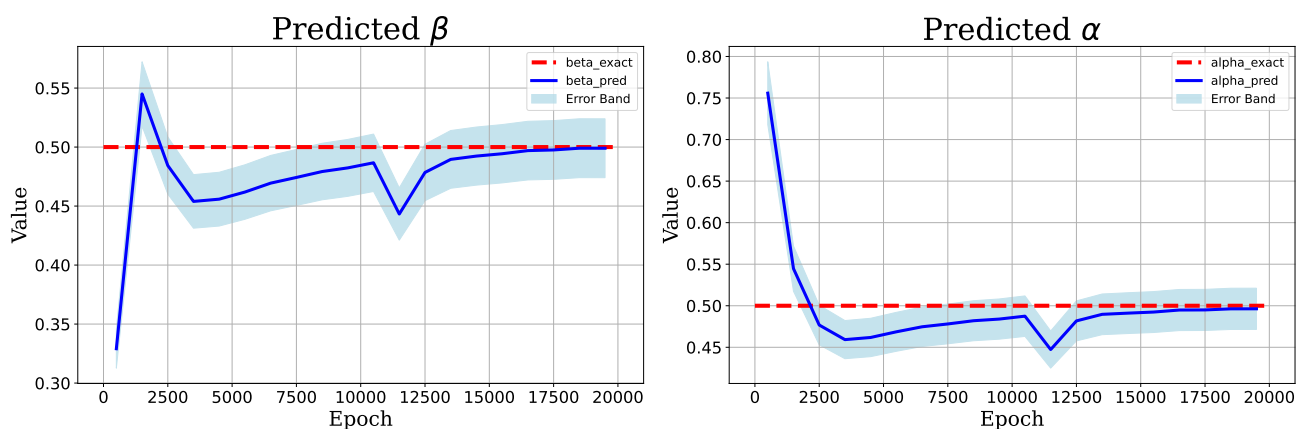
When employing the RD-DINN method for predicting disease dynamics, we simultaneously utilize a shared-weight deep neural network  $f_\theta(x, t), g_\theta(x, t)$  to invert the model's two key parameters: The propagation rate  $\beta$  and the recovery rate  $\alpha$ . The relative error is calculated as (3.14).

$$RelativeError = \frac{\|\lambda_{\text{pred}} - \lambda_{\text{true}}\|}{\|\lambda_{\text{true}}\|} \times 100\%. \quad (3.14)$$



**Figure 6.** In order from top to bottom  $S(x,t), I(x,t)$ . Left: Reference solution of SIR equation using a conventional finite difference solver. Middle: Predict the solutions  $S(x,t), I(x,t)$  using a small set of numerical solution data, where the black markers represent the spatiotemporal randomly sampled points  $\{(x_d^i, t_d^i)\}_{i=1}^N$ . Right: Absolute point-wise error.

In Figure 7, we can see that the two parameters fit well, and because of the initialization of the parameters of the neural network,  $\beta$  and  $\alpha$  start from zero, and as we keep iterating, the parameters are approaching the actual value. Multiple experiments revealed parameter convergence to true values at 20,000 iterations.



**Figure 7.** Comparison between the predicted values and reference values of parameters  $\beta$  and  $\alpha$  over training iterations. The shaded area corresponds to a 5% standard deviation, while the red line represents the reference values.

We also conduct a systematic study by maintaining the architecture of the neural network unchanged while varying the noise level and the number of sampled data points to demonstrate the excellent robustness of the algorithm. Regarding the noise level in the data, the method is highly robust. We set 0%, 5%, and 10% noise to data points at 1000, 2000, and 3000, and the MSE of the key parameters of the equation increases with the noise concentration, demonstrating that even if the noise pollution is as high as 10%, the corresponding data points can achieve a reasonable recognition accuracy.

**Table 2.** Percentage errors in identifying parameters  $\alpha$  and  $\beta$  for different numbers of hidden layers and neurons per layer. The training data is considered to be noise-free and fixed to  $N = 3000$ .

Layers	% error $\beta$				% error $\alpha$			
	40%	60%	80%	100%	40%	60%	80%	100%
6	0.177	0.400	0.267	0.336	0.433	0.772	0.629	0.710
8	0.224	0.230	0.329	0.424	0.503	0.600	0.563	0.841
10	0.045	0.291	0.258	0.618	0.106	0.617	0.433	0.981
12	0.127	0.247	0.224	0.273	0.171	0.405	0.371	0.509

**Table 3.** Percentage errors in identifying parameters  $\alpha$  and  $\beta$  for different numbers of training data  $N_d$  under varying levels of noise. In this case, the neural network architecture is fixed at 6 layers with 50 neurons per layer.

$N_d$	% error $\beta$			% error $\alpha$		
	0%	5%	10%	0%	5%	10%
1000	0.172	0.599	0.383	0.360	0.669	0.795
2000	0.259	0.465	0.393	0.535	0.853	1.089
3000	0.463	0.560	0.012	1.035	0.962	0.103

**Table 4.** Comparison of prediction solutions and equation parameters of Fully Connected Neural Networks and Residual Neural Networks when the number of neurons per layer is 100.

Network Layer	Predict Variable	Network Type	
		FCNN	ResNet
7	S	2.34e-03	1.95e-03
	I	2.53e-02	2.40e-02
	$\beta$	2.240%	2.098%
	$\alpha$	1.619%	1.506%
	S	1.94e-03	1.45e-03
9	I	2.00e-02	1.91e-02
	$\beta$	0.604%	0.299%
	$\alpha$	1.230%	0.321%

We can observe in Table 2 that none of the  $L_2$  errors of RD-DINN in parameter identification exceeds 1%, with 10 repeated trials demonstrating consistent error fluctuations within 0.5% and

confirming robustness to neural architecture variations. In Table 3, for adding different concentrations of noise data, none of the  $L_2$  errors in parameter identification exceeds 2%, as validated through 10 replications with a maximum error deviation of 0.5%, confirming the method's noise resilience.

In Table 4, it is evident that the residual neural network has a remarkable ability to reduce prediction errors. Specifically, when the number of network layers rises, the inversion error of the architectural parameters of the residual neural network shows a marked decrease. This indicates that the residual neural network is highly effective in improving prediction accuracy and maintaining stability in the face of increased network complexity.

### 3.2.2. Inverse SEIR model

In this inverse problem, we employ the SEIR model from the second forward problem, where the parameters of the model  $\beta, \sigma, \alpha$  and the dynamics are unknown. Only a small and scattered set of observable data with potential noises are provided for the hidden states  $S(x, t), E(x, t), I(x, t)$ , and  $R(x, t)$ . Unlike the first inverse problem case, we introduce an inversion for an additional parameter  $\sigma$ , which implies that we have less knowledge about the governing laws of the disease information presented by the model. Let

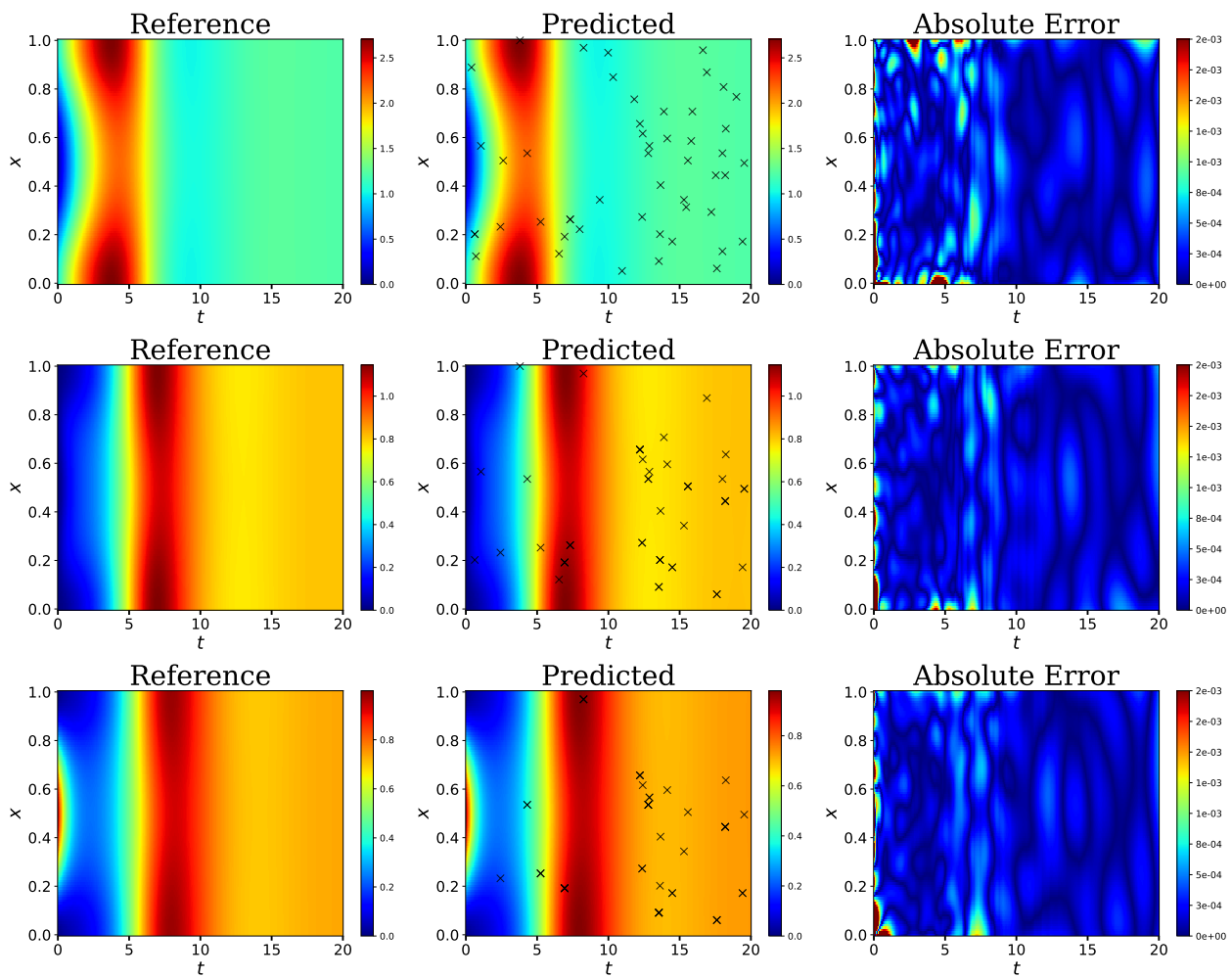
$$\begin{aligned} f(t, x) &:= \frac{\partial S(x, t)}{\partial t} - d_1 \frac{\partial^2 S}{\partial x^2} - B + \lambda_1 S(x, t) I(x, t) + \mu S(x, t), \\ g(t, x) &:= \frac{\partial E(x, t)}{\partial t} - d_2 \frac{\partial^2 E}{\partial x^2} - \lambda_1 S(x, t) I(x, t) + (\mu + \lambda_2) E(x, t), \\ h(t, x) &:= \frac{\partial I(x, t)}{\partial t} - d_3 \frac{\partial^2 I}{\partial x^2} - \lambda_2 E(x, t) + (\lambda_3 + \mu) I(x, t). \end{aligned}$$

Here, consistent with the inverse problem of the SIR model, we denote  $f(t, x), g(t, x), h(t, x)$  as surrogate models for the three equations. The disease dynamics parameters  $(\beta, \sigma, \alpha) = (\lambda_1, \lambda_2, \lambda_3)$  are all unknown and predicted through a deep neural network for the unknown dynamics of the compartments  $S(x, t), E(x, t)$ , and  $I(x, t)$ . During training,  $\lambda_1, \lambda_2$ , and  $\lambda_3$  are shared parameters with the neural network.

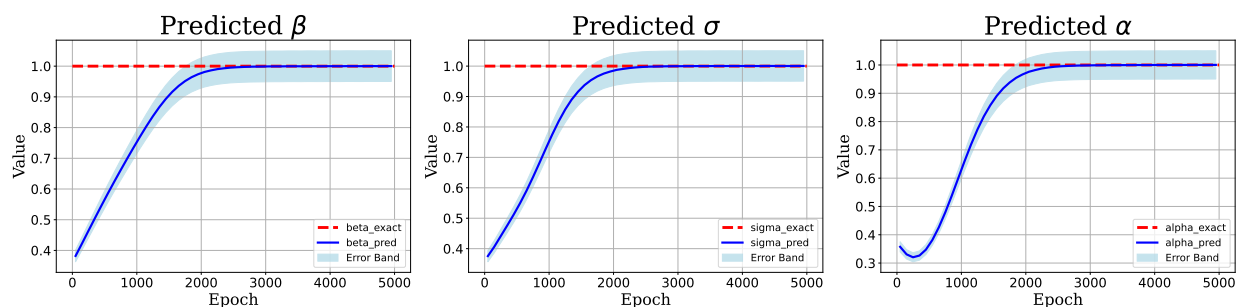
For the SEIR model, we give the representation (3.15) for the basic regeneration number, which is similar to the epidemic models thoroughly described by ordinary differential equations.

$$R_0 = \frac{B\beta\sigma}{(\mu + \sigma)(\mu + \alpha)\mu} \approx 8.26 \quad (3.15)$$

Figure 8 shows the constant values observed at  $t = 20$  in the entire domain, representing their respective steady states. We compute the basic regeneration number  $R_0 \geq 1$  by bringing in the parameters  $\beta, \alpha$ , a state in which the populations in each partition have reached a stable equilibrium, signaling the persistence of the disease.



**Figure 8.** In order from top to bottom  $S(x,t)$ ,  $E(x,t)$ ,  $I(x,t)$ . Left: Reference solution of SEIR equation using a conventional finite difference solver. Middle: Predicted solutions  $S(x,t)$ ,  $E(x,t)$ ,  $I(x,t)$  generated using a small amount of data, where the black marks represent spatial random sampling points  $\{(x_d^i, t_d^i)\}_{i=1}^N$ . Right: Absolute point-wise error.



**Figure 9.** Comparison between the predicted values and reference values of parameters  $\beta$ ,  $\sigma$ , and  $\alpha$  over training iterations. The red line represents the reference values and the shaded region corresponds to a 1% standard deviation of the parameters. As the number of training iterations increases, the inverted parameters converge towards the reference values.



In Figure 9, the three-parameter fits are the same as the actual values, and the neural network parameters are initialized with a value of 1. At the beginning of the iteration, the neural network is randomly fitted, and the values of the fitted parameters are in a decreasing trend. The parameters are gradually approaching the actual values as we keep on iterating and they are the same as the actual parameters at the 50,000 epoch.

**Table 5.** Percentage errors in identifying parameters  $\beta$ ,  $\sigma$ ,  $\alpha$  for different numbers of hidden layers and neurons per layer. The training data is considered to be noise-free and fixed to  $N = 3000$ .

Layers	% error $\beta$			% error $\sigma$			% error $\alpha$		
	60%	80%	100%	60%	80%	100%	60%	80%	100%
6	0.003	0.004	0.001	0.014	0.001	0.005	0.013	0.009	0.001
8	0.004	0.004	0.002	0.016	0.008	0.006	0.006	0.002	0.003
10	0.002	0.001	0.002	0.014	0.004	0.009	0.010	0.002	0.008
12	0.002	0.001	0.004	0.003	0.005	0.010	0.003	0.000	0.008

**Table 6.** Percentage errors in identifying parameters  $\beta$ ,  $\sigma$ ,  $\alpha$  for different numbers of training data  $N_d$  under varying levels of noise. The neural network architecture is fixed at 6 layers with 100 neurons per layer.

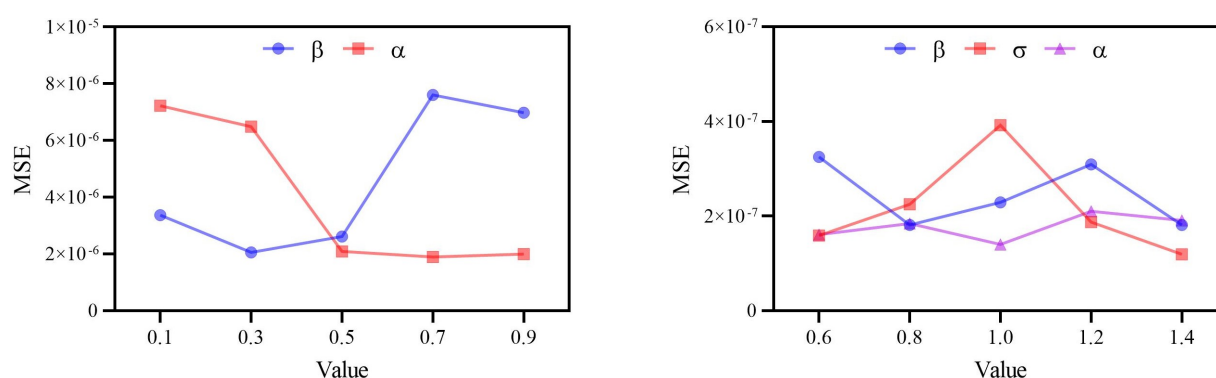
$N_d$	% error $\beta$			% error $\sigma$			% error $\alpha$		
	0%	5%	10%	0%	5%	10%	0%	5%	10%
1000	0.003	0.352	0.526	0.003	0.092	0.152	0.002	0.432	0.369
2000	0.002	0.153	0.222	0.002	0.024	0.151	0.002	0.140	0.227
3000	0.001	0.004	0.087	0.001	0.037	0.078	0.001	0.037	0.049

Table 5 demonstrates that RD-DINN achieves nearly zero  $L_2$  error in parameter identification across 10 experimental replications, confirming its accuracy under the basic neural network architecture. As shown in Table 6, while the parameter identification error increases with higher noise concentrations, all errors remain below 1% in 10 replicate experiments, demonstrating consistent performance under noisy conditions.

Model predictions exhibit biases during rapid compartment population shifts, particularly in early outbreaks or unforeseen events, reflecting inherent disease unpredictability and estimation challenges.

### 3.2.3. Sensitivity analysis

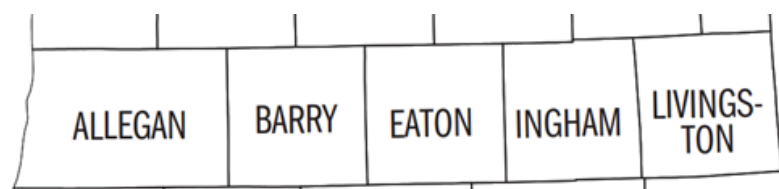
Figure 10 shows that in the SIR inverse problem for  $\alpha = 0.5$ , the error rises significantly with increasing  $\beta$ , indicating that the model is sensitive to high propagation rates.  $\beta = 0.5$ , the error decreases significantly with increasing  $\alpha$ , indicating that the model is sensitive to low recovery rates. At high  $\beta$ , disease outbreak dynamics dominate and model predictions are more stable. The SEIR model inversion errors reached an overall order of magnitude of  $1e-07$ , indicating a low sensitivity to different parameters. Regarding inter-parameter coupling effects (e.g., simultaneous changes in  $\beta$  and  $\alpha$ ), this can be extended to multi-dimensional parameter scans in the future.



**Figure 10.** Left: Fix the SIR models  $\{\beta, \alpha\} = 0.5$  separately and calculate the MSE of another parameter within the range of 0.1–0.9. Right: The MSE of two parameters in the fixed SEIR model  $\{\beta, \sigma, \alpha\} = 1.0$  and the other parameter within the range of 0.6–1.4.

#### 4. Method validation

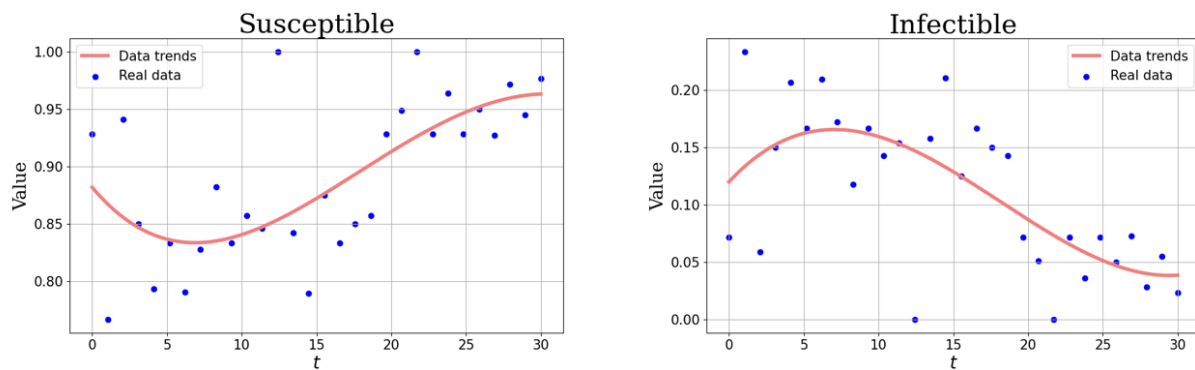
To validate the model in a public health outbreak, we use data on new crown infections in five cities in mid-Michigan (Allegan, Barry, Eaton, Ingham, and Livings-Ton) in April 2020. We view the properties of a consistent and contiguous spatial distribution of city latitudes as a condition on the spatial  $x$  in Figure 11, and we determine whether an infected person is a member of the public through daily nucleic acid tests reported to the public to determine whether an individual is infected. We use population density to represent the current status of each category.



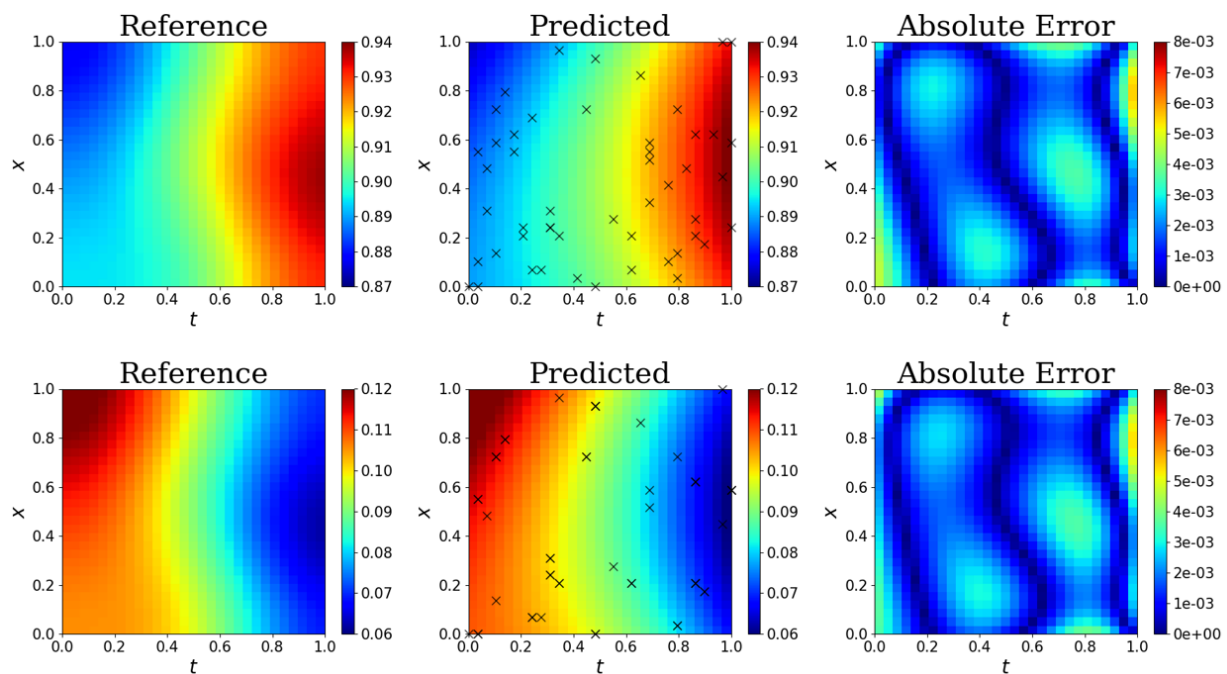
**Figure 11.** Map of five cities in the state of Michigan that are spatially aligned at latitude and in close proximity to each other.

We construct the space of disease for the four cities by performing spatial penalization, considering only transmission at latitude (a one-dimensional spatial problem only). Due to the high degree of dispersion in the data, some degree of data smoothing was required; we compare the impact of the smoothing methods on the data, applying a moving average filter using 3, 4, and 5-day windows, and choosing to apply the 4-day window filter twice; smoothing had little impact on the trend of the data and did not negatively affect the data.

We can see in Figure 12 that the city of Livings-Ton shows an increasing trend in the density of susceptible people and a gradual decreasing trend in the density of infected people in April. Since we do not know this city's birth and death rates, the classical PDE-SIR model with a diffusion term is used to validate this set of accurate data.



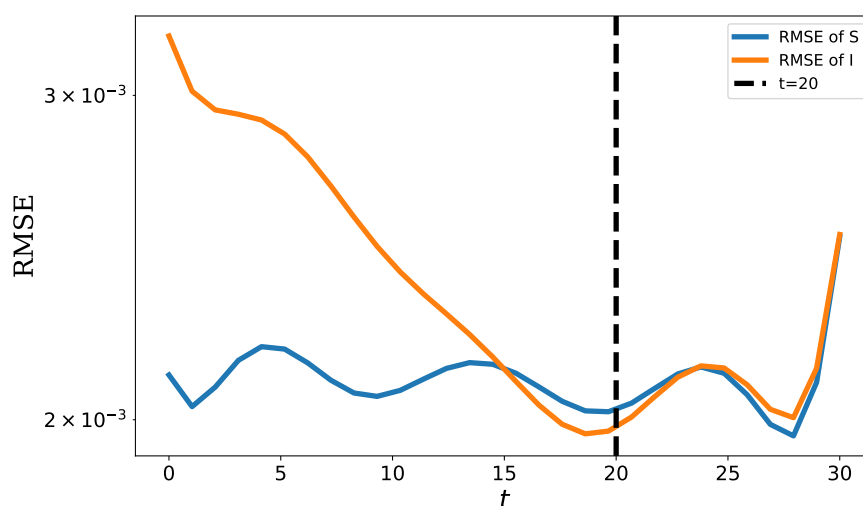
**Figure 12.** Population Density of Susceptible and Infected Persons in Livings-Ton Cities, Michigan, in April.



**Figure 13.** RD-DINN fits real data in an inverse problem such that the Adam optimizer with a learning rate of  $1e - 03$  iterates 3000 epochs.

By viewing the five cities with the same latitude and close to each other as a one-dimensional spatial distribution, we can see in Figure13 that RD-DINN can reach an error of  $5e - 03$  in predicting the spatial distribution. The infection rate is  $\beta$  of 0.037 with a float of 10% through the RD-DINN inversion parameter. The recovery rate is 0.070 with a float of 8%. By calculating the mostly regeneration number  $R_0 = \frac{\beta}{\alpha} < 1$ , it can be concluded that the disease receded briefly in April, which is basically in line with the actual data we selected for the state of Michigan. In Figure 14, we verify the model's extrapolation ability, using the first 20 days of real data to train a pre-training model to reason about the growth of various types of population densities in the latter 10 days. It is not difficult to see that the inference

stage and the pre-training stage of the RMSE basically maintains the same order of magnitude, which indicates that the model has the ability to reason about the future state of the disease. Here, the inference phase error has some growth because the model also has some empirical error.



**Figure 14.** RMSE of predicted vs true values of density of infected and susceptible persons from April 20 to April 30.

## 5. Conclusions

In this paper, we introduce a data-driven deep learning approach based on PINN, which is RD-DINN. RD-DINN is a deep neural network method capable of learning various diseases and how they spread, predicting disease progression, and identifying parameters of disease spread. In the forward problem, where the control equation, initial conditions, and boundary conditions are known, they are incorporated into the neural network loss function. The proposed method achieves an absolute error with finite difference methods of  $1e - 03$ . In the inverse problem, where the inversion of model parameters is crucial, and due to the lack of exact solutions for partial differential equations, we extract a random subset of sampling points from numerical solutions as mini-batch data to minimize the equation and mini-batch data losses. The absolute error between the predicted results and numerical solutions reach  $1e - 03$ . Although RD-DINN demonstrates effective predictions in the forward problem, there is still some error compared to high-order finite differences. We are actively working on more advanced methods to encode disease information into deep neural networks in a more advanced way to improve the errors that occur in positive problems. The inverse problem for time-dependent parameters (e.g., a time-varying transmission rate due to interventions or seasonal effects) is also challenging.

This is the first application of the DINN method to diffusion-based infectious disease models, and the results from two sets of benchmark numerical experiments are promising. Traditional methods such as finite difference and finite element have proven effective in deducing model dynamics. However, RD-DINN not only demonstrates satisfactory performance in predicting model dynamics in the forward problem, but also excels in using mini-batch data for the inverse problem, where it successfully inverts the dynamics and unknown parameters of the model. It is essential to acknowledge that, given the lack

of exact solutions for the discussed partial differential equations, the mini-batch data are sampled from the numerical solutions obtained through high-order finite differences.

In the two benchmark disease experiments, we investigate the forward problem of predicting the dynamics of diffusion diseases and the inverse problem of estimating unknown parameters. We also assess the robustness of the RD-DINN method under different neural network architectures and noise levels. We validate the model's effectiveness under new crown data from five cities in Michigan. In conclusion, our proposed Reaction-Diffusion Disease Information Neural Network shows promising prospects as a robust and reliable candidate for learning the dynamics and unknown parameters of compartmental models in infectious diseases, particularly in the context of inverse problems.

### Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

### Acknowledgments

The authors are grateful to the National Natural Science Foundation of China (12471505, 12101576).

### Conflict of interest

The authors declare there are no conflicts of interest.

### References

1. W. O. Kermack, A. G. McKendrick, A contribution to the mathematical theory of epidemics, *Proc. R. Soc. Lond. A*, **115** (1927), 700–721. <http://doi.org/10.1098/rspa.1927.0118>
2. T. Sigler, S. Mahmuda, A. Kimpton, J. Loginova, P. Wohland, E. Charles-Edwards, et al., The socio-spatial determinants of COVID-19 diffusion: The impact of globalisation, settlement characteristics and population, *Global. Health*, **17** (2021), 56. <https://doi.org/10.1186/s12992-021-00707-2>
3. N. Sene, SIR epidemic model with Mittag–Leffler fractional derivative, *Chaos, Solitons Fractals*, **137** (2020), 109833. <https://doi.org/10.1016/j.chaos.2020.109833>
4. J. A. Backer, A. Berto, C. McCreary, F. Martelli, W. H. M. van der Poel, Transmission dynamics of hepatitis E virus in pigs: Estimation from field data and effect of vaccination, *Epidemics*, **4** (2012), 86–92. <https://doi.org/10.1016/j.epidem.2012.02.002>
5. R. Anderson, R. May, *Infectious Diseases of Humans: Dynamics and Control*, Oxford Academic, 1991. <https://doi.org/10.1093/oso/9780198545996.001.0001>
6. A. Nauman, M. Rafiq, M. A. Rehman, M. Ali, M. O. Ahmad, Numerical modeling of SEIR measles dynamics with diffusion, *Commun. Math. Appl.*, **9** (2018), 315–326.
7. T. N. Sindhu, A. Shafiq, Z. Huassian, Generalized exponentiated unit Gompertz distribution for modeling arthritic pain relief times data: Classical approach to statistical inference, *J. Biopharm. Stat.*, **34** (2024), 323–348. <https://doi.org/10.1080/10543406.2023.2210681>

8. T. N. Sindhu, A. Shafiq, M. B. Riaz, T. A. Abushal, H. Ahmad, E. M. Almetwally, et al., Introducing the new arcsine-generator distribution family: An in-depth exploration with an illustrative example of the inverse weibull distribution for analyzing healthcare industry data, *J. Radiat. Res. Appl. Sci.*, **17** (2024), 100879. <https://doi.org/10.1016/j.jrras.2024.100879>
9. T. N. Sindhu, A. Shafiq, Q. M. Al-Mdallal, Exponentiated transformation of Gumbel Type-II distribution for modeling COVID-19 data, *Alexandria Eng. J.*, **60** (2021), 671–689. <https://doi.org/10.1016/j.aej.2020.09.060>
10. A. Shafiq, A. B. Çolak, T. N. Sindhu, S. A. Lone, A. Alsubie, F. Jarad, Comparative study of artificial neural network versus parametric method in COVID-19 data analysis, *Results Phys.*, **38** (2022), 105613. <https://doi.org/10.1016/j.rinp.2022.105613>
11. A. Shafiq, A. B. Çolak, T. N. Sindhu, S. A. Lone, T. A. Abushal, Modeling and survival exploration of breast carcinoma: A statistical, maximum likelihood estimation, and artificial neural network perspective, *Artif. Intell. Life Sci.*, **4** (2023), 100082. <https://doi.org/10.1016/j.ailsci.2023.100082>
12. G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, *Nat. Rev. Phys.*, **3** (2021), 422–440. <https://doi.org/10.1038/s42254-021-00314-5>
13. M. Raissi, P. Perdikaris, G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.*, **378** (2019), 686–707. <https://doi.org/10.1016/j.jcp.2018.10.045>
14. S. Cai, Z. Mao, Z. Wang, M. Yin, G. E. Karniadakis, Physics-informed neural networks (PINNs) for fluid mechanics: A review, *Acta Mech. Sin.*, **37** (2021), 1727–1738. <https://doi.org/10.1007/s10409-021-01148-1>
15. A. Yazdani, L. Lu, M. Raissi, G. E. Karniadakis, Systems biology informed deep learning for inferring parameters and hidden dynamics, *PLOS Comput. Biol.*, **16** (2020), e1007575. <https://doi.org/10.1371/journal.pcbi.1007575>
16. M. Raissi, N. Ramezani, P. N. Seshaiyer, On parameter estimation approaches for predicting disease transmission through optimization, deep learning and statistical inference methods, *Lett. Biomath.*, **6** (2019), 1–26.
17. S. Shaier, M. Raissi, P. Seshaiyer, Data-driven approaches for predicting spread of infectious diseases through DINNs: Disease informed neural networks, *Lett. Biomath.*, **9** (2022), 71–105.
18. S. Yin, J. Wu, P. Song, Optimal control by deep learning techniques and its applications on epidemic models, *J. Math. Biol.*, **86** (2023), 36. <https://doi.org/10.1007/s00285-023-01873-0>
19. F. V. Difonzo, L. Lopez, S. F. Pellegrino, Physics informed neural networks for learning the horizon size in bond-based peridynamic models, *Comput. Methods Appl. Mech. Eng.*, **436** (2025), 117727. <https://doi.org/10.1016/j.cma.2024.117727>
20. F. V. Difonzo, L. Lopez, S. F. Pellegrino, Physics informed neural networks for an inverse problem in peridynamic models, *Eng. Comput.*, **2024** (2024). <https://doi.org/10.1007/s00366-024-01957-5>
21. D. Ouedraogo, I. Ibrango, A. Guiro, Global stability for reaction-diffusion SIR model with general incidence function, *Malaya J. Mat.*, **10** (2022), 139–150. <https://doi.org/10.26637/mjm1002/004>

22. J. P. C. Santos, E. Monteiro, J. C. Ferreira, N. H. T. Lemes, D. S. Rodrigues, Well-posedness and qualitative analysis of a SEIR model with spatial diffusion for COVID-19 spreading, *Biomath*, **12** (2023), 2307207. <https://doi.org/10.55630/j.biomath.2023.07.207>
23. S. Chinviriyasit, W. Chinviriyasit, Numerical modelling of an SIR epidemic model with diffusion, *Appl. Math. Comput.*, **216** (2010), 395–409. <https://doi.org/10.1016/j.amc.2010.01.028>
24. P. van den Driessche, J. Watmough, Reproduction numbers and sub-threshold endemic equilibria for compartmental models of disease transmission, *Math. Biosci.*, **180** (2002), 29–48. [https://doi.org/10.1016/S0025-5564\(02\)00108-6](https://doi.org/10.1016/S0025-5564(02)00108-6)
25. Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *Nature*, **521** (2015), 436–444. <https://doi.org/10.1038/nature14539>
26. A. G. Baydin, B. A. Pearlmutter, A. A. Radul, J. M. Siskind, Automatic differentiation in machine learning: A survey, *J. Mach. Learn. Res.*, **18** (2018), 1–43.
27. M. Raissi, G. E. Karniadakis, Hidden physics models: Machine learning of nonlinear partial differential equations, *J. Comput. Phys.*, **357** (2018), 125–141. <https://doi.org/10.1016/j.jcp.2017.11.039>
28. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, (2016), 770–778. <https://doi.org/10.1109/CVPR.2016.90>
29. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, et al., Attention is all you need, preprint, arXiv:1706.03762.
30. M. D. McKay, R. J. Beckman, W. J. Conover, A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, *Technometrics*, **21** (1979), 239–245. <https://doi.org/10.2307/1268522>
31. E. Massad, M. N. Burattini, F. B. A. Coutinho, L. F. Lopez, The 1918 influenza A epidemic in the city of São Paulo Brazil, *Med. Hypotheses*, **68** (2007), 442–445. <https://doi.org/10.1016/j.mehy.2006.07.041>
32. N. Sene, 2-Numerical methods applied to a class of SEIR epidemic models described by the Caputo derivative, *Methods Math. Model.*, **2022** (2022), 23–40. <https://doi.org/10.1016/B978-0-323-99888-8.00003-6>
33. S. L. Brunton, J. L. Proctor, J. N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, *Proc. Natl. Acad. Sci. U.S.A.*, **113** (2016), 3932–3937. <https://doi.org/10.1073/pnas.1517384113>

## Appendix

Figure A1 illustrates the training process and convergence of neural networks in both direct and inverse problems. In the direct problem, the neural network solves for the system state variables using known partial differential equations, initial conditions, and boundary conditions. The total loss and its components, such as the initial condition loss, boundary condition loss, and function loss, gradually decrease as the number of training epochs increases. This trend indicates that the neural

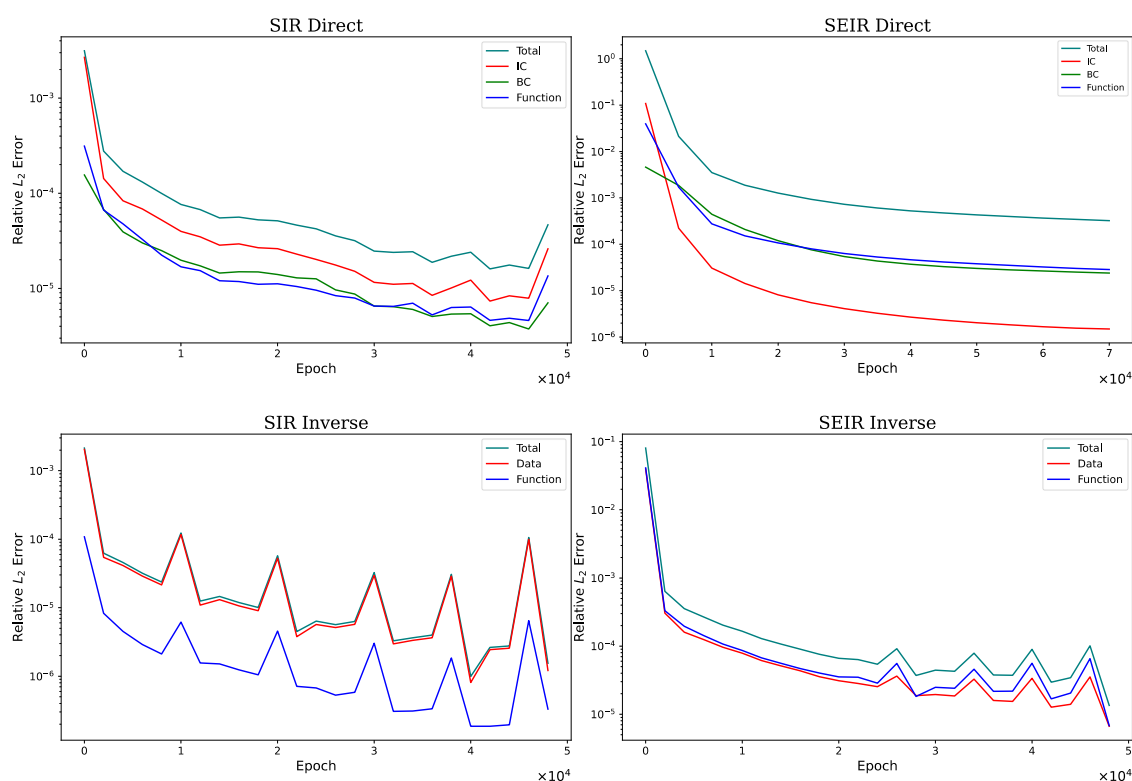
network progressively approximates the true solution. In the inverse problem, the neural network infers the governing equations, initial conditions, or boundary conditions from observed data. Similarly, the total loss and its components decrease with increasing training epochs, demonstrating that the neural network not only fits the observed data but also accurately infers unknown system parameters or dynamic behaviors. These results validate the effectiveness of neural networks in both direct and inverse problems, highlighting their powerful capabilities in modeling complex systems and performing parameter inversion. The total loss in a direct problem consists of three components

$$\mathcal{L}_{Total} = \mathcal{L}_{IC} + \mathcal{L}_{BC} + \mathcal{L}_{PDE},$$

where  $\mathcal{L}_{IC}$  is mse for initial conditions,  $\mathcal{L}_{BC}$  is mse for boundary conditions, and  $\mathcal{L}_{PDE}$  is mse for PDE residuals. The total loss in a inverse problem consists of two components

$$\mathcal{L}_{Total} = \mathcal{L}_{Data} + \mathcal{L}_{PDE},$$

where  $\mathcal{L}_{Data}$  is mse for truth and predict data,  $\mathcal{L}_{PDE}$  mse for PDE residuals.



**Figure A1.** RD-DINN loss during training using Adam optimizer with initial learning rate of  $1e - 03$ , using cyclic learning rate mechanism with step size = 500.