

Learning Spacial Domain Infectious Disease Control Model for Vaccine Distribution by Physics-Constraint Machine Learning

Chun Li (✉ lichun2020@hit.edu.cn)

Harbin Institute of Technology Shenzhen <https://orcid.org/0000-0002-2021-751X>

Yunyun Yang

Harbin Institute of Technology Shenzhen <https://orcid.org/0000-0002-0488-7652>

Xin Jiang

Harbin Institute of Technology Shenzhen

Boying Wu

Harbin Institute of Technology

Research Article

Keywords: Uncertainty learning, epidemic control, dynamic system, physics-constraint machine learning.

Posted Date: January 27th, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-1248214/v1>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Learning Spacial Domain Infectious Disease Control Model for Vaccine Distribution by Physics-Constraint Machine Learning

Chun Li¹ · Yunyun Yang^{1,*} · Xin Jiang¹ · Boying Wu²

Received: date / Accepted: date

Abstract The coronavirus disease, COVID-19, has become a global challenging pandemic. Causing significant loss of life, property, and economics worldwide. The study of infectious disease models allows us to have a deeper understand of spreading trend of highly infectious diseases. Therefore, studying infectious disease models is imminent. In this work, we propose learning spacial domain infectious disease control model for vaccine distribution by physics-constraint machine learning. Usually, these dynamical systems are utilized in epidemiology models to predict the time evolution and development trend of highly infectious diseases such as COVID-19. We reformulate the SIR models and give corresponding policies via dynamical systems. More importantly, we obtain the approximating numerical solution of the systems of dynamical PDEs via the PINNs algorithm, within the acceptable range of approximation error. Additionally, we present several numerical solutions of the PDEs under a variety of scenarios.

Keywords Uncertainty learning · epidemic control · dynamic system · physics-constraint machine learning.

1 Introduction

The coronavirus disease pandemic, named COVID-19, has been spread fast worldwide since the end of 2019, becoming the largest global public health crisis and global challenging pandemic. Until Dec 9, 2021, the

coronavirus COVID-19 had been affecting 213 countries and territories around the world reported by <https://www.worldometers.info/coronavirus/>. Moreover, COVID-19 was caused by a kind of savagely contagious virus and could lead to acute respiratory distress and multiple organ failure [1]. World Health Organization (WHO) has declared the COVID-19 as a global health emergency on January 30, 2020 [2], which poses a great threat to international human health. To prevent the spread of the epidemic, some strategies such as social distancing, citywide lockdown, propaganda of health education knowledge, traffic halt, and community management have been carried out by the governments of some countries in the world. Such as the disease, COVID-19, which can infect a large portion of the population, causes great pressure on medical system. Therefore, the prediction of infectious diseases is an important undertaking in health work. Through the prediction of infectious diseases, we can detect the development trend of the disease in time, thereby improving the epidemic prevention work, and strangling infectious diseases as early as possible. Subsequently, the analysis and prediction of infectious diseases are essential to prevent the occurrence of diseases, formulate disease control strategies, treatment, and make health decision.

Eventually, every epidemic comes to a natural halt, but its impact on the population may be very serious. The goal of epidemic control is to prevent and slow down the spread of the virus as much as possible so that it does not affect the health care system under the premise of economic operation. All of this should be done within the available resources. Usually, the data available for analysis is extremely limited, in early outbreak of the disease. For the lack of large-scale testing and investigation analysis, we almost know nothing

Chun Li, Yunyun Yang (*Corresponding author), Xin Jiang
E-mail: (lichun2020,yangyunyun@hit.edu.cn)
· Boying Wu E-mail: (mathwby@hit.edu.cn)

¹ School of Science, Harbin Institute of Technology, Shenzhen, 518055, China

² School of Mathematics, Harbin Institute of Technology, Harbin, 150001, China

ing about new epidemics. The public health response is to try to isolate everyone, whether susceptible, infected, or immune. This may lead to panic reactions among the public. Such reactions include the chaotic and economically devastating public health response to the epidemic.

During the spread of infectious diseases, there are many uncertainties. Based on the considerations above, it is necessary to integrate the knowledge of statistical, machine learning, probability theory, and differential equation to explore and solve the challenging problems that people are facing. About the stochastic differential equation theory, the predecessors did a lot of valuable research, for more details see for instance [3–5]. Modern society is an era of information fusion, multi-disciplinary integration, so all disciplines (such as mathematical biology, machine learning, biostatistics, and bioengineering) have developed rapidly.

To have a deeper understanding of the epidemic, plenty of machine learning-based studies have been conducted [6, 7]. Several theoretical studies of the evolution of infectious diseases of the COVID-19 are recently proposed in [8–13]. However, there are still two major challenges: 1). Some scholars do a lot of theoretical analysis, but lack real data simulation, resulting in a certain gap between the model and reality. 2). Most of the methods based on machine learning are only demonstrated through experiments, lacking the interpretability of the model. Therefore, how to handle these challenges is the key to the successful identification of COVID-19.

Consequently, in this work, we propose an uncertainty-based learning framework, called learning spacial domain infectious disease control model for vaccine distribution by physics-constraint machine learning, to analyze and predict the development trend of highly infectious diseases. Also, we study the problem of optimal control of an epidemic model employing a partial differential equations (PDEs), which plays a crucial role in the prediction of the studied system and even in the decision-making policies. It allows policy-makers to make more reasonable decisions, avoids economically devastating and public panic as much as possible.

The main contributions of this paper are summarized as follows,

1. We propose to formulate epidemic predicting-controlling by optimal control using PINNs algorithm to solve PDEs.
2. We deepen the analysis of such phenomena in uncertainty situations research from the perspective of numerical analysis.
3. Also, we have demonstrated better performance in the task of epidemic control and given the relevant

departments the basis to make more reasonable decisions.

The rest of the paper is organized as follows, in Section 2, we first present some related works about traditional infectious disease model for the simulation of highly infectious diseases and deep-learning-based approach to solve PDEs. We introduce the proposed algorithm about learning spacial domain infectious disease control model for vaccine distribution by physics-constraint machine learning (LSPCML) in Section 3. After that, in Section 4, we describe the experimental settings and experimental results. Finally, we summarize this paper and present the limitations of the current study, and present several future research directions in Section 5.

2 related work

In this section, we briefly review the most relevant studies from the following aspects: 1) The traditional infectious disease model for simulation of highly infectious diseases, 2) Machine learning-based studies for identification of highly infectious diseases.

Traditional infectious disease model for the simulation of highly infectious diseases. The SIR epidemic model was first proposed by Kermack and McKendrick in 1927 [14]. An infectious disease epidemic is described via an ordinary differential equation system. It is regarded as the gold standard tool of mathematical epidemiology for quantitative analysis of the spread of an epidemic,

$$\begin{cases} \frac{dS}{dt} = -\beta IS, \\ \frac{dI}{dt} = \beta IS - \gamma I, \\ \frac{dR}{dt} = \gamma I, \end{cases} \quad (1)$$

where S , I , and R : $[0, T] \rightarrow [0, 1]$ refer to the fraction of individuals who are susceptible, infectious, and recovered. And the constant parameters $\beta > 0$, $\gamma > 0$ refer to the infectious and recovery rate, respectively. For example, Bliman et al. [15] used the SIR epidemic model to understand how to apply partial or total containment to SIR epidemic model during a given finite time interval to minimize the epidemic final size. Also, Chen et al. [5] proposed a time-dependent SIR model for COVID-19 with undetectable infected persons. And Kitsos et al. [11] used the quasi-linear integro-differential hyperbolic systems-application to an epidemic model. Moreover, Chen et al. [16] introduced the population migration to SEIAR for COVID-19 epidemic modeling with an efficient intervention strategy. Zheng et al. [17] proposed a hybrid artificial-intelligence (AI) model on

COVID-19 prediction. Besides, Cui et al. [18] developed concentration behavior of endemic equilibrium for a reaction-diffusion SIS epidemic model with mass action infection mechanism. More importantly, Fabricius et al. [19] explored the threshold of epidemic spreading for a stochastic SIR model with local and global contacts. And Nishiura et al. [20] proposed a model to estimate the serial interval of novel coronavirus (COVID-19).

Mathematically, According to [21, 22] the spatial SIRV model can be viewed as the following variational optimization problem. We consider the following functions $S, I, R, V : [0, T] \times \Omega \rightarrow R_+$, and Ω refers to a bounded domain. Also, we assume S, I, R , and V move on a spatial domain with velocities v_S, v_I, v_R and v_V , respectively. Therefore, the movements can be described as the following continuity PDEs,

$$\begin{cases} \frac{\partial S}{\partial t} + \operatorname{div}(v_S S) + \beta SKI - \frac{\eta_S^2}{2} \Delta S + \theta_1 VS = 0, \\ \frac{\partial I}{\partial t} + \operatorname{div}(v_I I) + \beta SKI + \gamma I - \frac{\eta_I^2}{2} \Delta I = 0, \\ \frac{\partial R}{\partial t} + \operatorname{div}(v_R R) - \gamma I - \frac{\eta_R^2}{2} \Delta R - \theta_1 VS = 0, \\ \frac{\partial V}{\partial t} - f(x, t) + \theta_2 VS = 0, \frac{\partial V}{\partial t} + \operatorname{div}(v_V V) + \theta_2 VS = 0, \\ S(0, \cdot), E(0, \cdot), I(0, \cdot), R(0, \cdot) \text{ are given,} \end{cases} \quad (2)$$

where θ_1, θ_2, v_V , and K refer to the vaccine efficiency, utilization rate of vaccines, and Gaussian kernel function, respectively. And the control variable $f(t, x)$ describes how many vaccines are produced at a specific time and location.

Consider the following variational optimization control problem,

$$\inf_{S, I, R, V} \left(\int_0^T \int_{\Omega} \left(\frac{\alpha_S}{2} S \|v_S\|^2 + \frac{\alpha_I}{2} I \|v_I\|^2 + \frac{\alpha_R}{2} R \|v_R\|^2 + \frac{\alpha_V}{2} \|v_V\|^2 + \frac{\epsilon}{2} (S + I + R + V)^2 \right) dxdt \right) + L(I(T, \cdot)), \quad (3)$$

s.t. (2) with fixed initial densities.

Deep learning-based approach to solve PDEs.

Recently, with the rapid development of DL, the numerical methods of PDEs have made significant progress. Advantageously, compared with the traditional mesh-based numerical schemes, DNNs as mesh-free approximators. So they can not only avoid the curse of dimensionality but also approximate the solutions of PDEs on complex geometries effectively. Using neural networks (NNs) to solve partial differential equations (PDEs) is not a new thing. For example, Lagaris et al. [23] used NNs to solve ordinary equations and PDEs. And Raissi et al. [24] developed physics-informed neural networks (PINNs) to solve both forward and inverse problems.

Also, Yang et al. [25] proposed the Bayesian PINNs to learn inverse problems from noisy data. Moreover, Eliasof et al. [26] developed a family of architectures that are motivated by numerical methods to solve PDEs on manifolds. Besides, Krishnapriyan et al. [27] proposed the characterizing possible failure modes in PINNs from the opposite perspective. More recently, Both et al. [28] introduced the DeepMoD for model discovery in noisy data. Also, Lu et al. [29] developed a DL library to solve differential equations and dynamic systems. And Sirignano et al. [30] introduced a new DL PDEs augmentation method with application to large-eddy simulation. Moreover, Lu et al. [31] used the universal approximation theorem of operators to learn the nonlinear operators. Furthermore, Wang et al. [32] developed the physics-informed DeepONets to learn the solution operator of parametric PDEs. Besides, Karniadakis et al. [33] developed physics-informed machine learning.

3 methodology

In this section, we develop a novel spatial-type diffusion-reaction SIRV system, to not only control the propagation of pandemics by the optimal transportation strategy of vaccine distribution but also to understand the spread of epidemic diseases. After that, the PINNs are used to solve the variational optimization control problems.

3.1 Problem Description

Motivated by the variational problem in image reconstruction [34], we take the following objective function as an example,

$$E(u) = \frac{1}{2} \int_{\Omega} |u_0 - Au|^2 dx + \lambda \int_{\Omega} \varphi(|\nabla u|) dx, \quad (4)$$

where u, u_0 refer to a reference and given image, and A is a linear operator. We assume that the objective function has a minimum point u , then it satisfies the following Euler-Lagrange equation,

$$A^* Au - \lambda \nabla \cdot \left(\frac{\varphi'(|\nabla u|)}{|\nabla u|} \nabla u \right) = A^* u_0, \quad (5)$$

where A^* is the conjugate operator of A , φ is a diffusion function.

Table 1: Summary of existing surrogate curvature functions

Function Name	EE [36]	TSC [37]	TRV [37]
$g(x, \alpha)$	$\alpha + \beta x^2, \alpha, \beta > 0$	$1 + \alpha x ^2$	$\sqrt{1 + \alpha x ^2}$
Function Name	Geman [38]	log-det [39]	Laplace [40]
$g(x, \alpha)$	$\frac{(1+x)x}{\alpha+x}$	$\log(1+x^2)$	$1 - \exp(-\frac{x}{\alpha})$

According to the Proposition 1 in [21], the following system can be obtained,

$$\begin{cases} \frac{\partial S}{\partial t} + \operatorname{div}(\varphi(\nabla v_S)S) + \beta SI - \frac{\eta_S^2}{2}\Delta S + \theta_1 VS = 0, \\ \frac{\partial I}{\partial t} + \operatorname{div}(\varphi(\nabla v_I)I) + \beta SI + \gamma I - \frac{\eta_I^2}{2}\Delta I = 0, \\ \frac{\partial R}{\partial t} + \operatorname{div}(\varphi(\nabla v_R)R) - \gamma I - \frac{\eta_R^2}{2}\Delta R - \theta_1 VS = 0, \\ \frac{\partial V}{\partial t} - f(x, t) + \theta_2 VS = 0, \\ \frac{\partial V}{\partial t} + \operatorname{div}(\varphi(\nabla v_V)V) + \theta_2 VS = 0, \\ S(0, \cdot), E(0, \cdot), I(0, \cdot), R(0, \cdot), \text{ are given,} \end{cases} \quad (6)$$

where φ is also the diffusion function.

Proposition 1 *Given the initial data $S_0, I_0, R_0,$ and $V_0,$ the unique solution (S, I, R, V) exists for all $t > 0.$ Moreover, there exists a positive constant $C_1,$ which depends on initial data $S_0, I_0, R_0,$ and $V_0,$ such that (S, I, R, V) satisfies,*

$$\begin{aligned} & \|S(\cdot, t)\|_{L^\infty(\Omega)} + \|I(\cdot, t)\|_{L^\infty(\Omega)} + \|R(\cdot, t)\|_{L^\infty(\Omega)} \\ & + \|V(\cdot, t)\|_{L^\infty(\Omega)} \leq C_1, \forall t \geq 0. \end{aligned} \quad (7)$$

It can be proved by [35] similar means. Proposition 1 shows that the system (6) is bounded. And the diffusion functions are shown in Table 1.

3.2 Optimization

Definition 1 [41] (**Feedback neural network**). Let $U_l = \{1, 2, \dots, d^{(l)}\}, 0 \leq l \leq L,$ and consider the sequence of affine functions $w_1, w_2, \dots, w_L, w_l : F(U_{l-1}) \rightarrow F(U_l)$ and the sequence of activation functions $\sigma^{(l)} : R \rightarrow R.$ Then the corresponding feedforward NN is the sequence of maps $F_{\Theta_1}, F_{\Theta_2}, \dots, F_{\Theta_L},$ where $F_{\Theta_l} = \sigma^{(l)} \odot w_l \odot F_{\Theta_{l-1}}, 1 \leq l \leq L,$ with F_{Θ_0} given.

Definition 2 (**Neural Network Space**) [41]. A neural network space with L hidden layers and m nodes on each layer can be described as:

$$\begin{cases} \mathcal{L}_n^m(\sigma) = \{F : R^{n+1} \rightarrow R, \\ F_0(x) = \sigma(w_0x + b_0), w_0 \in R^{m \times (n+1)}, b_0 \in R^m, \\ F_l(x) = \sigma(w_l F_{l-1}(x) + b_l), l = 1, \dots, L, \\ w_l \in R^{m \times m}, b_l \in R^m, \\ F(x) = \beta^T F_l(x) + c, \beta \in R^m, c \in R\} \end{cases}$$

(8)

To numerically solve the system Eq. (6), a general PDE as the following form is taken into consideration,

$$\begin{cases} u_t = \mathcal{L}u + \mathcal{N}u, x, y \in \Omega, \\ u(x, y, t) = g(t), x, y \in \partial\Omega, \\ u(x, y, 0) = u_0(x, y), x, y \in \Omega, \end{cases} \quad (9)$$

where \mathcal{L} refers to the differential operator, \mathcal{N} is the nonlinear operator, $g(t)$ is the boundary condition, and $u_0(x)$ refers to the initial value condition. More specifically, a process of numerically solving PDEs, can be viewed as a PDEs-constraint problem. So, according to Definition 1, if F refers to NN, the system (6) can be viewed as the following PDEs-constraint optimization problem,

$$\hat{\mathbf{u}}(\mathbf{x}) = F_\Theta(\mathbf{x}, \Theta) \text{ s.t. } \Theta^* = \arg \min_{\Theta} L(\Theta). \quad (10)$$

According to Definition 2, Eq. (9) can be embedded in the NN space as follow,

$$\begin{cases} NN_t = \mathcal{L}NN(x, y, t; \Theta) + \mathcal{N}NN(x, y, t; \Theta), x, y \in \Omega, \\ NN(x, y, t; \Theta) = g(t), x, y \in \partial\Omega, \\ u(x, y, 0) = u_0(x), x, y \in \Omega. \end{cases} \quad (11)$$

To facilitate the use of NN to learn the above system, some auxiliary variables are introduced,

$$\begin{cases} \frac{\partial S}{\partial t} + \operatorname{div}(\varphi(\nabla v_S)S) + \beta SI - \frac{\eta_S^2}{2}\Delta S + \theta_1 VS = e_1, \\ \frac{\partial I}{\partial t} + \operatorname{div}(\varphi(\nabla v_I)I) + \beta SI + \gamma I - \frac{\eta_I^2}{2}\Delta I = e_2, \\ \frac{\partial R}{\partial t} + \operatorname{div}(\varphi(\nabla v_R)R) - \gamma I - \frac{\eta_R^2}{2}\Delta R - \theta_1 VS = e_3, \\ \frac{\partial V}{\partial t} - f(x, t) + \theta_2 VS = e_4, \\ \frac{\partial V}{\partial t} + \operatorname{div}(\varphi(\nabla v_V)V) + \theta_2 VS = e_5. \end{cases} \quad (12)$$

Definition 3 (**Loss Function of PCL**). A NN space with L hidden layers and m nodes on each layer can be described as:

$$Loss(\Theta) = \lambda_1 \|PDE(\Theta)\|^2 + \lambda_2 \|IC(\Theta)\|^2 + \lambda_3 \|BC(\Theta)\|^2, \quad (13)$$

where BC is the boundary condition, IC is the initial value condition, $\lambda_i (i = 1, 2, 3)$ are the training weights of the equation.

According to Definition 3 and PINNs [24, 25, 33], we can get the loss function as,

$$L(\Theta) = \|e_1\|_2^2 + \|e_2\|_2^2 + \|e_3\|_2^2 + \|e_4\|_2^2 + \|e_5\|_2^2. \quad (14)$$

After that, different optimizers including AdaBelief [42], AdaBound [43], AdamP [44], Apollo [45], PID [46], SGDP [44], and AdamX [47] are used to train the models for several epochs. And the algorithm is summarized in Algorithm 1.

Algorithm 1: Learning Spacial Domain Infectious Disease Control Model for Vaccine Distribution by Physics-Constraint Machine Learning.

Initialization:

- (1) Initialize parameter Θ , and learning rate α_n ;
- (2) Initialize u^0 ;

for $n = 1, 2, 3, \dots, N$ **do**

- (1) Read current $L(\Theta)$ via Eq. (14);
- (2) Update parameter Θ via gradient descent
 $\Theta_{n+1} = \Theta_n - \alpha_n \nabla_{\Theta} L_r(\Theta_n)$;

end

output:

Output the learned solution;

4 Experimental results

In this section, more details about the experimental setup and results are described. To verify the effectiveness and efficiency of our learning diffusion SIRV based on physics-constrained machine learning from noisy data, comprehensive experiments on several initial conditions are implemented, including different initial conditions, comparison methods, evaluation metrics, and implementation details. Also, the comparison results and some related discussions are present.

4.1 Different Initial Conditions, Evaluation Metrics, and Comparison Methods

In this subsection, three different types of initial conditions are set to test the diffusion SIRV system. The purpose of our experiments is to show that the PINNs successfully search the solution of the system by setting up spatial barriers of the vaccine production and distribution. More specifically, we can test the influence of this factor on the system solution of V -related parameters by controlling f (vaccine productivity) in initial condition 1. Moreover, initial condition 2 represents the situation of a single factory. Furthermore, initial condition 3 represents the situation of multiple factories. And all initial conditions are summarized in Table 1.

To quantitatively evaluate the learning solution performance, three popular metrics (Peak Signal-to-Noise Ratio (PSNR) [50], Structural Similarity Index Measurement (SSIM) [50], and Normalized Root Mean Square Error (NRMSE)) [51] are adopted in this study. Given a ground-truth solution image $g(x)$ and a learning solution $y(x)$, PSNR is defined as $PSNR = 10 \log_{10} \frac{\max^2(G(x), y(x))}{\frac{1}{N} \sum \|G(x) - y(x)\|^2}$, where N refers to the total number of voxels in each solution image, and $\max^2(G(x), y(x))$ is the maximal intensity value of the ground-truth solution $G(x)$ and the learning solution $y(x)$. Structural Similarity Index

Measurement (SSIM), which is defined as $SSIM = \frac{(2\mu_{G(x)}\mu_{y(x)} + C_1)(2\sigma_{G(x)y(x)} + C_2)}{(\mu_{G(x)}^2 + \mu_{y(x)}^2 + C_1)(\sigma_{G(x)}^2 + \sigma_{y(x)}^2 + C_2)}$, where $\mu_{G(x)}$, $\mu_{y(x)}$, $\sigma_{G(x)}$, $\sigma_{y(x)}$ refer to the means and variances of solution image $G(x)$ and $y(x)$, and $\sigma_{G(x)y(x)}$ is the covariance of $y(x)$ and $G(x)$. And C_1, C_2 refer to the positive constants, which are used to avoid a null denominator. Normalized Root Mean Square Error (NRMSE), which can be defined as $NRMSE = \sqrt{\frac{G(x) - y(x)}{\|G(x)\|_2^2}}$. It should be noted that the higher PSNR and SSIM values and lower NRMSE values show higher quality in the learned solution.

To verify the effectiveness of the proposed PINNs-based SIRV model, we compare it with two different NNs, DnCNN [48] and VNet [49]. These NNs can be summarized as follows: 1) DnCNN [48]. This method constructs the feed-forward denoising convolutional neural networks (DnCNNs). 2) VNet [49] proposed a new variational inference method, which integrates both noise estimation and image denoising into a unique Bayesian framework, for blind image denoising.

4.2 Implementation Details

The AdamX [47] solver is used to train the NNs. Moreover, to train the proposed model, 5000 epochs are conducted. The 14 hidden layers (each layer contains 20 neurons) are used for our networks. Besides, we take $\Omega \times [0, T] = [0, 1]^2 \times [0, 1]$, with a uniform space-grid $N_x = N_y = 100$ and a uniform time-grid $N_t = 100$ in all experiments. And the 50 boundary, 50000 inner, and 5000 initial sampling points are chosen to train NNs. And the learning rate is set to 0.001. Furthermore, Our equipment has a GeForce RTX 1080 super GPU. And all numerical experiments are performed utilizing the high-performance DL library (PyTorch) [52].

4.3 Different Diffusion Methods

To quantitatively evaluate the learning solution performance, different diffusion approaches are used, including Euler Elastica (EE) [36], Laplace [40], log-determinant (log-det) [39], total roto-translational variation (TRV) [37], and total square curvature (TSC) [37]. 1) EE [36] has a broad application in computer vision and image processing. The definition came from the algebraic geometry and applications, which is first introduced into computer vision by Mumford from the differential geometry aspect. 2) Laplace [40], is used as a regularizer to approximate the traditional nuclear norm for image denoising tasks. 3) Log-det [39], in several machine learning and computer vision problems we of-

Table 2: Summary of some initial conditions

Initial condition	$S_{init}, I_{init}, R_{init}, V_{init}$
Condition-1	$\begin{cases} S_0 = (2 \exp(-5 [(x_1 - 0.7)^2 + (x_2 - 0.7)^2]) - 1.5)_+, \\ I_0 = (2 \exp(-5 [(x_1 - 0.7)^2 + (x_2 - 0.7)^2]) - 1.8)_+, \\ R_0 = V_0 = 0, (x)_+ = \max(x, 0). \end{cases}$
Condition-2	$\begin{cases} S_0 = (2 \exp(-15 [(x_1 - 0.2)^2 + (x_2 - 0.5)^2]) - 1.6)_+ \\ + (2 \exp(-15 [(x_1 - 0.8)^2 + (x_2 - 0.5)^2]) - 1.6)_+, \\ I_0 = (2 \exp(-15 [(x_1 - 0.2)^2 + (x_2 - 0.5)^2]) - 1.8)_+ \\ R_0 = V_0 = 0. \end{cases}$
Condition-3	$\begin{cases} S_0 = (2 \exp(-15 [(x_1 - 0.8)^2 + (x_2 - 0.8)^2]) - 1.6)_+ \\ + (2 \exp(-15 [(x_1 - 0.2)^2 + (x_2 - 0.7)^2]) - 1.6)_+ \\ + (2 \exp(-15 [(x_1 - 0.8)^2 + (x_2 - 0.3)^2]) - 1.6)_+ \\ + (2 \exp(-15 [(x_1 - 0.2)^2 + (x_2 - 0.2)^2]) - 1.6)_+, \\ I_0 = (2 \exp(-15 [(x_1 - 0.2)^2 + (x_2 - 0.7)^2]) - 1.8)_+ \\ + (2 \exp(-15 [(x_1 - 0.2)^2 + (x_2 - 0.2)^2]) - 1.8)_+, \\ R_0 = V_0 = 0. \end{cases}$

Table 3: Comparison results (SSIM/PSNR/NRMSE) of different diffusion functions under initial condition 3

Diffusion function	Noise level	S	I	R	V
EE [36]	noise level = 0.1	0.9792/23.2155/0.0736	0.9568/25.20771/0.0635	0.9576/25.0402/0.0645	0.9750/26.4540/0.0547
	noise level = 0.5	0.97152/21.6920/0.087	0.9316/24.4224/0.0695	0.9308/23.0240/0.0814	0.9714/26.3336/0.0555
	noise level = 0.9	0.9678/23.2172/0.0736	0.9222/23.8893/0.0739	0.9265/23.5252/0.0768	0.9639/25.7803/0.0591
Laplace [40]	noise level = 0.1	0.8974/15.4973/0.1825	0.9306/18.7378/0.1294	0.8985/17.7066/0.1429	0.9580/25.1540/0.0618
	noise level = 0.5	0.8727/15.9829/0.1725	0.9149/18.1939/0.1378	0.8664/14.8776/0.1979	0.9226/18.5302/0.1326
	noise level = 0.9	0.8688/15.9344/0.1735	0.8920/16.2658/0.1720	0.8870/15.6995/0.1800	0.9225/19.1044/0.1241
log-det [39]	noise level = 0.1	0.8974/15.4973/0.1825	0.9306/18.7378/0.1294	0.8985/17.7066/0.1429	0.9580/25.1540/0.0618
	noise level = 0.5	0.8727/15.9829/0.1725	0.9149/18.1939/0.1378	0.8664/14.8776/0.1979	0.9226/18.5302/0.1326
	noise level = 0.9	0.8688/15.9344/0.1735	0.8920/16.2658/0.1720	0.8870/15.6995/0.1800	0.9225/19.1044/0.1241
TRV [37]	noise level = 0.1	0.8735/18.1702/0.1359	0.9413/21.5758/0.095	0.8797/15.3008/0.1949	0.92076/18.1248/0.1404
	noise level = 0.5	0.8603/18.1095/0.1368	0.9403/21.5916/0.0951	0.8377/11.9392/0.2871	0.9370/22.5005/0.0848
	noise level = 0.9	0.8596/17.5988/0.1451	0.8823/15.1992/0.1972	0.8823/15.1992/0.1972	0.9150/17.1044/0.1579
TSC [37]	noise level = 0.1	0.8735/18.1702/0.1359	0.9413/21.5758/0.0953	0.8797/15.3008/0.1949	0.9207/18.1248/0.1404
	noise level = 0.5	0.8603/18.1095/0.1368	0.9403/21.5916/0.0951	0.8377/11.9392/0.2871	0.9370/22.5005/0.0848
	noise level = 0.9	0.8596/17.5988/0.1451	0.9525/23.7664/0.0740	0.8823/15.1992/0.1972	0.9150/17.1044/0.1579

Table 4: Comparison results (SSIM/PSNR/NRMSE) of different neural networks under initial condition 3 (TSC)

Neural networks	Noise level	S	I	R	V
DnCNN [48]	noise level = 0.1	0.9030/17.7154/0.1426	0.9191/17.1170/0.1507	0.9398/18.7545/0.1258	0.8946/13.2686/0.2380
	noise level = 0.5	0.9072/18.3638/0.1324	0.9186/17.4050/0.1458	0.9405/18.7239/0.1263	0.8946/13.2417/0.2387
	noise level = 0.9	0.9073/18.2077/0.1348	0.91406/16.6557/0.159	0.9390/18.4518/0.1303	0.8878/12.9143/0.2479
VDN [49]	noise level = 0.1	0.7494/16.9920/0.1525	0.7732/16.8984/0.1525	0.7315/17.0165/0.1517	0.9298/24.3740/0.065
	noise level = 0.5	0.7189/16.5743/0.1600	0.7374/16.5707/0.1584	0.7446/15.8900/0.1727	0.8726/19.5101/0.1146
	noise level = 0.9	0.7439/16.2941/0.1653	0.7455/17.1106/0.1488	0.7318/16.9459/0.1529	0.8244/19.2187/0.1186
Our	noise level = 0.1	0.8735/18.1702/0.1359	0.9413/21.5758/0.095	0.8797/15.3008/0.1949	0.92076/18.1248/0.1404
	noise level = 0.5	0.8603/18.1095/0.1368	0.9403/21.5916/0.0951	0.8377/11.9392/0.2871	0.9370/22.5005/0.0848
	noise level = 0.9	0.8596/17.5988/0.1451	0.8823/15.1992/0.1972	0.8823/15.1992/0.1972	0.9150/17.1044/0.1579

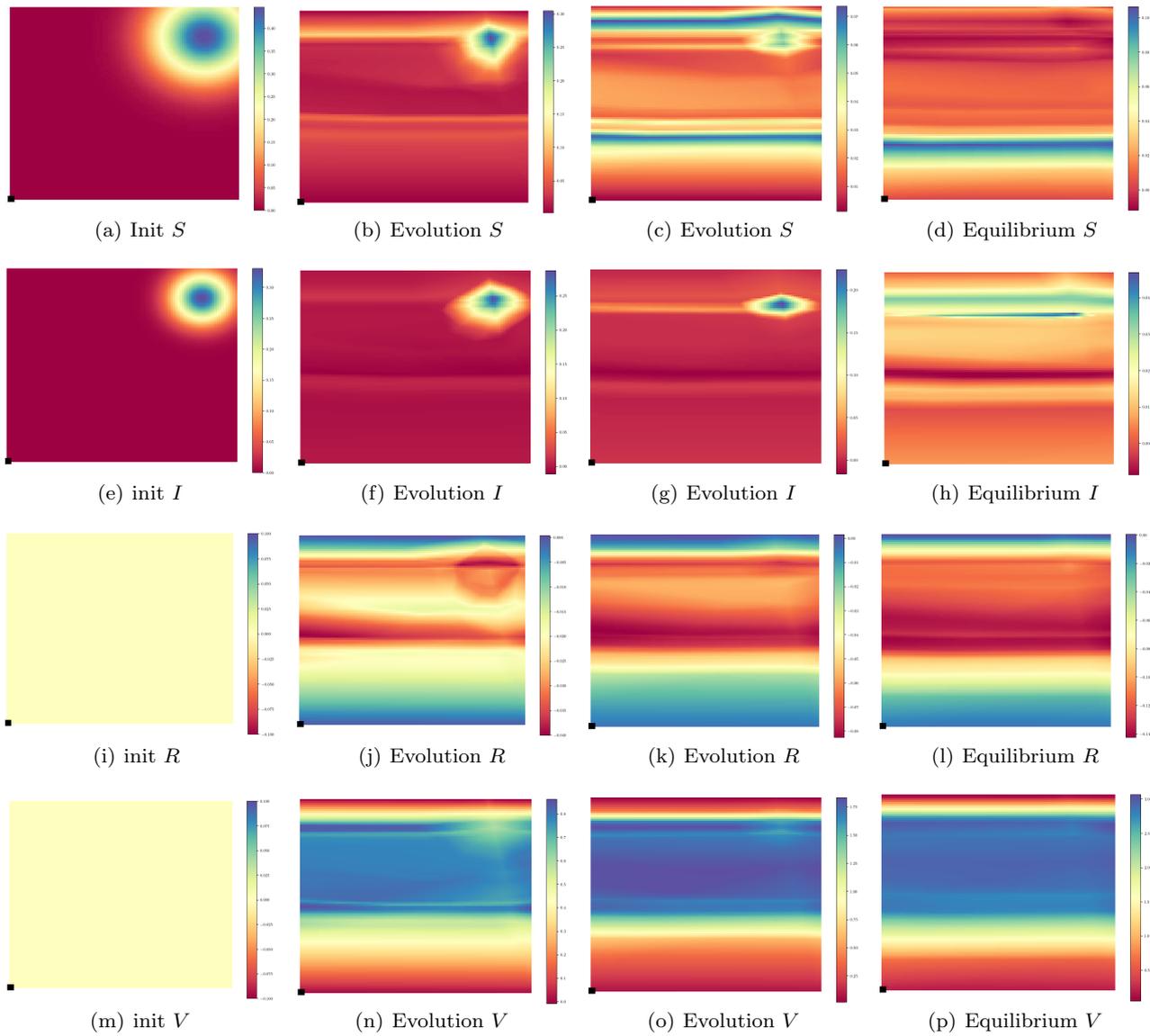


Fig. 1: The visual images under initial condition 1 (TSC).

Table 5: Comparison results (SSIM/PSNR/NRMSE) of different optimizers under initial condition 3 (noise level=0.1 (TSC))

Optimizers	S	I	R	V
AdaBelief [42]	0.8770/15.0715/0.1941	0.8821/12.4692/0.2720	0.8603/12.3503/0.2738	0.8716/12.1799/0.2785
AdaBound [43]	0.8506/14.7798/0.2008	0.8763/13.6364/0.2378	0.8886/16.2819/0.1741	0.8721/14.0787/0.2238
AdamP [44]	0.8749/18.2209/0.1351	0.9310/20.9773/0.1021	0.8758/14.8918/0.2043	0.9144/18.4511/0.1353
Apollo [45]	0.8511/14.6005/0.2050	0.8831/13.6749/0.2367	0.8774/14.0421/0.2253	0.8936/14.5193/0.2127
PID [46]	0.8706/14.5634/0.2058	0.8773/12.2946/0.2775	0.8673/12.4763/0.2699	0.8877/13.0355/0.2523
SGDP [44]	0.8706/14.5643/0.2058	0.8770/12.2942/0.2775	0.8686/12.4787/0.2698	0.8876/13.0345/0.2524
Adam [47]	0.8735/18.1702/0.1359	0.9413/21.5758/0.095	0.8797/15.3008/0.1949	0.9207/18.1248/0.1404
AdamX [47]	0.8786/17.9735/0.1390	0.9400/22.7719/0.0830	0.8711/15.0540/0.2006	0.9247/17.5701/0.1497

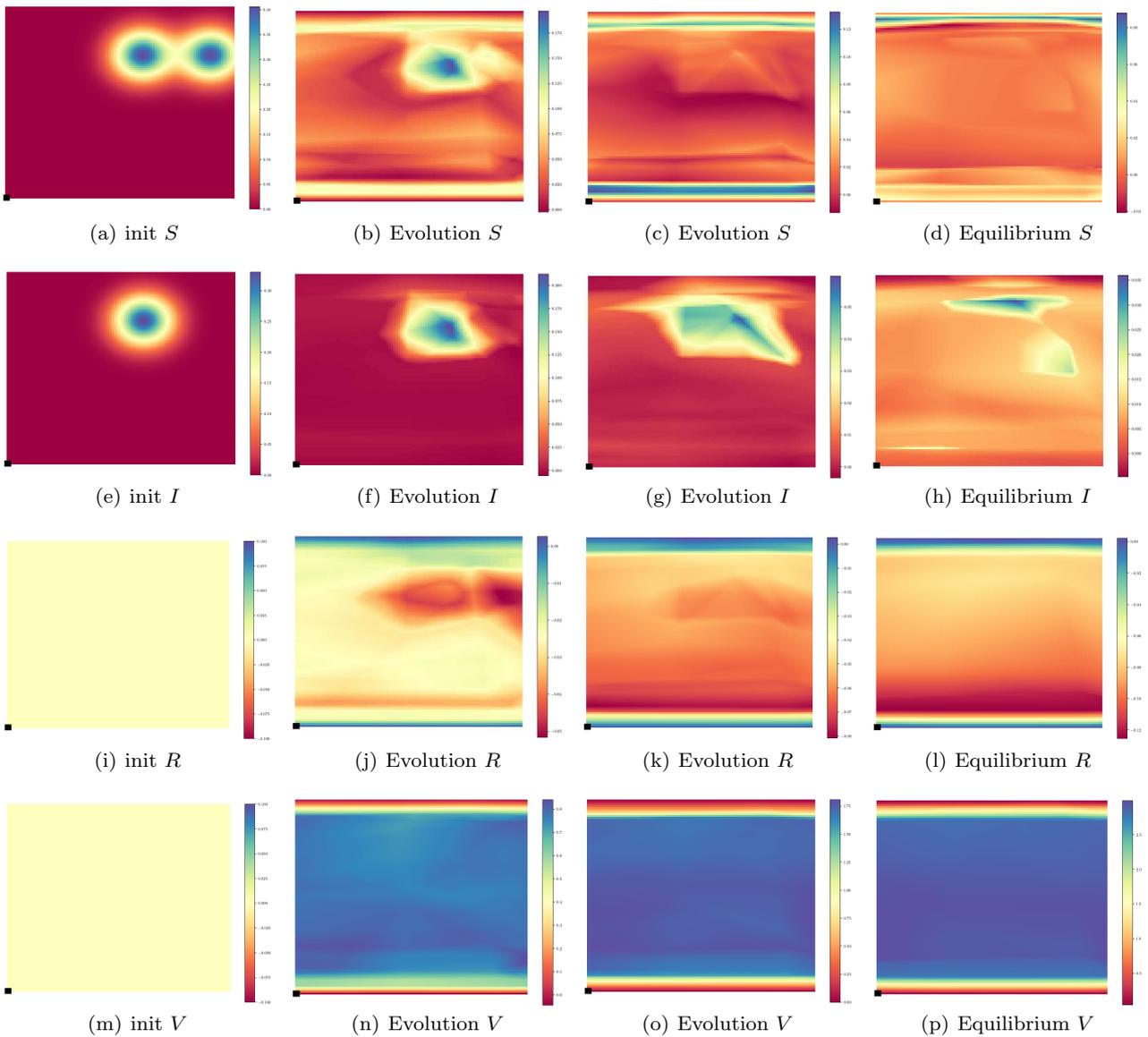


Fig. 2: The visual images under initial condition 2 (TSC).

ten encounter low-rank matrix. To obtain a low-rank representation, a log-determinant (LogDet) function is used to approximate the low-rank matrix. 4) TRV and TSC [37], they are used as the regularizers in the task of image denoising, TRV measures the surface area and curvature through a Euclidean metric. And the TSC penalizes the surface area and the squared curvature.

4.4 Different Optimization Solvers

To explore which optimization solver is more suitable to solve the system, the following optimization solvers are tested. They are AdaBelief [42], AdaBound [43], AdamP [44], Apollo [45], PID [46], SGDP [44], and

AdamX [47]. 1) AdaBelief [42], is introduced to achieve fast convergence as in adaptive methods and training stability, respectively. 2) AdaBound [43] is a variant of Adam, which used the dynamic bounds on learning rates to achieve a gradual and smooth transition. 3) AdamP and SGDP [44] is used to get rid of the radial component, or the norm-increasing direction, at each optimizer step. They were proposed by Heo et al. [44]. This method not only keeps the original convergence of the GD optimizer but also avoids changing the effective update direction. 4) Apollo was introduced by Ma [45], which is a quasi-Newton method for nonconvex stochastic optimization and the curvature is incorporated into the loss function to approximate Hessian

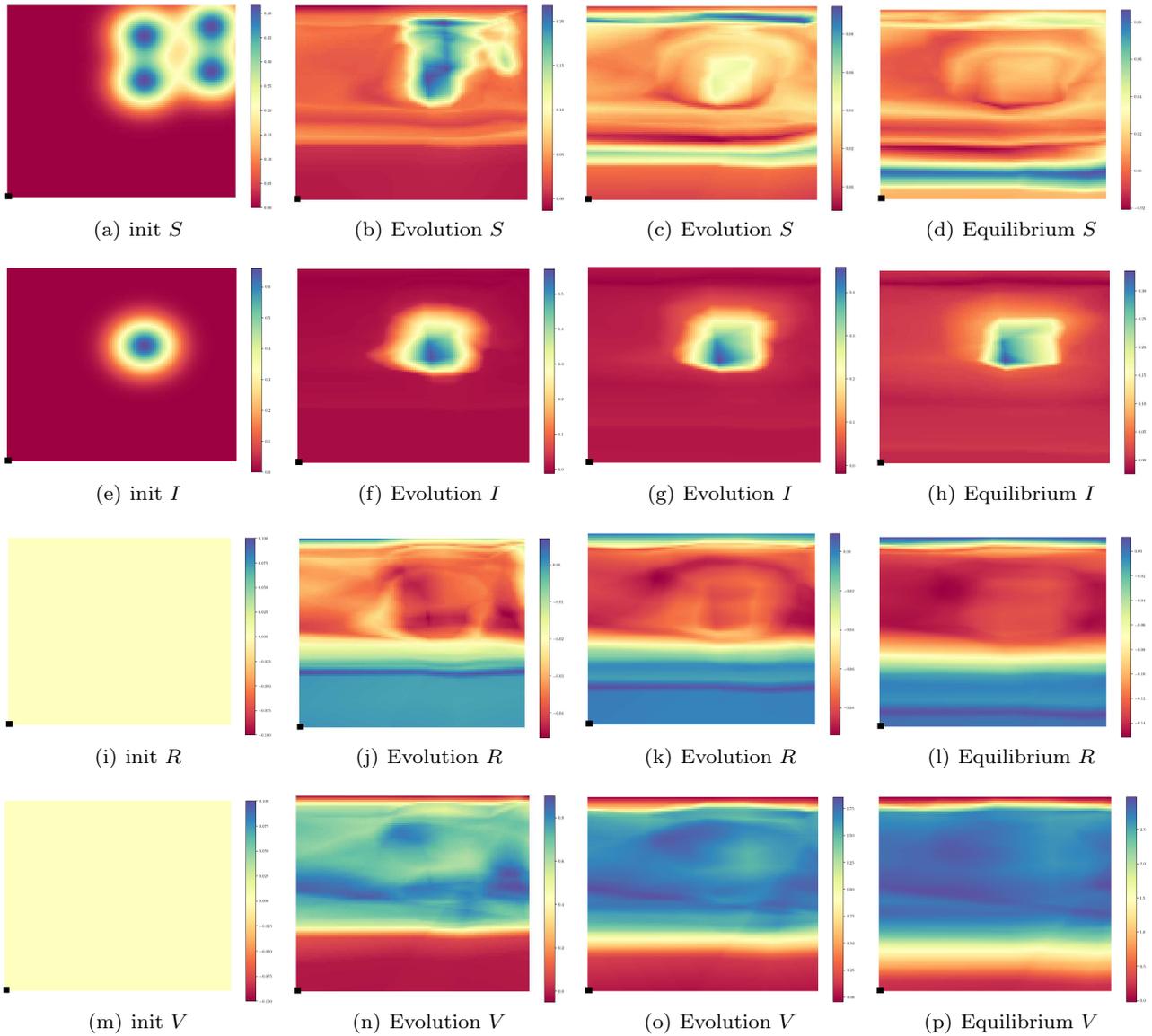


Fig. 3: The visual images under initial condition 3 (TSC).

matrix. 5) PID [46] is a method to accelerate NNs optimization, which reduces much of the overshoot phenomena of SGD-Momentum. AdamX [47] is an algorithm for first-order gradient-based optimization, which is computationally efficient and has few memory requirements.

4.5 Results Comparison

Three initial conditions are experimented to evaluate our model. With the vaccine has been successfully developed and starting to be used in the market, more and more people are vaccinated in society. The vaccination policy is adopted by the governments and coun-

tries as the best choice. To study the effect of vaccines on disease control, we select a vaccine coverage rate for simulation experiments. We also do simulation experiments for the models in corresponding noise level. In all experiments, the parameters are set as follows, $\beta = 0.8$, $\gamma = 0.1$, $\eta_S = 0.01$, $\eta_I = 0.01$, $\eta_R = 0.01$, $f = 10$, and $\theta_1 = \theta_2 = 0.9$. We first evaluate the performance under initial condition 1. From Fig. 1, it can be seen that the susceptible is gradually decreasing. And the infected is gradually controlled. Moreover, the number of infected is on the decline. The number of recovered continues to increase, and the number of vaccinated people continues to increase, thereby increasing the immunity of group. For the initial condition 2, from Fig. 2 we can

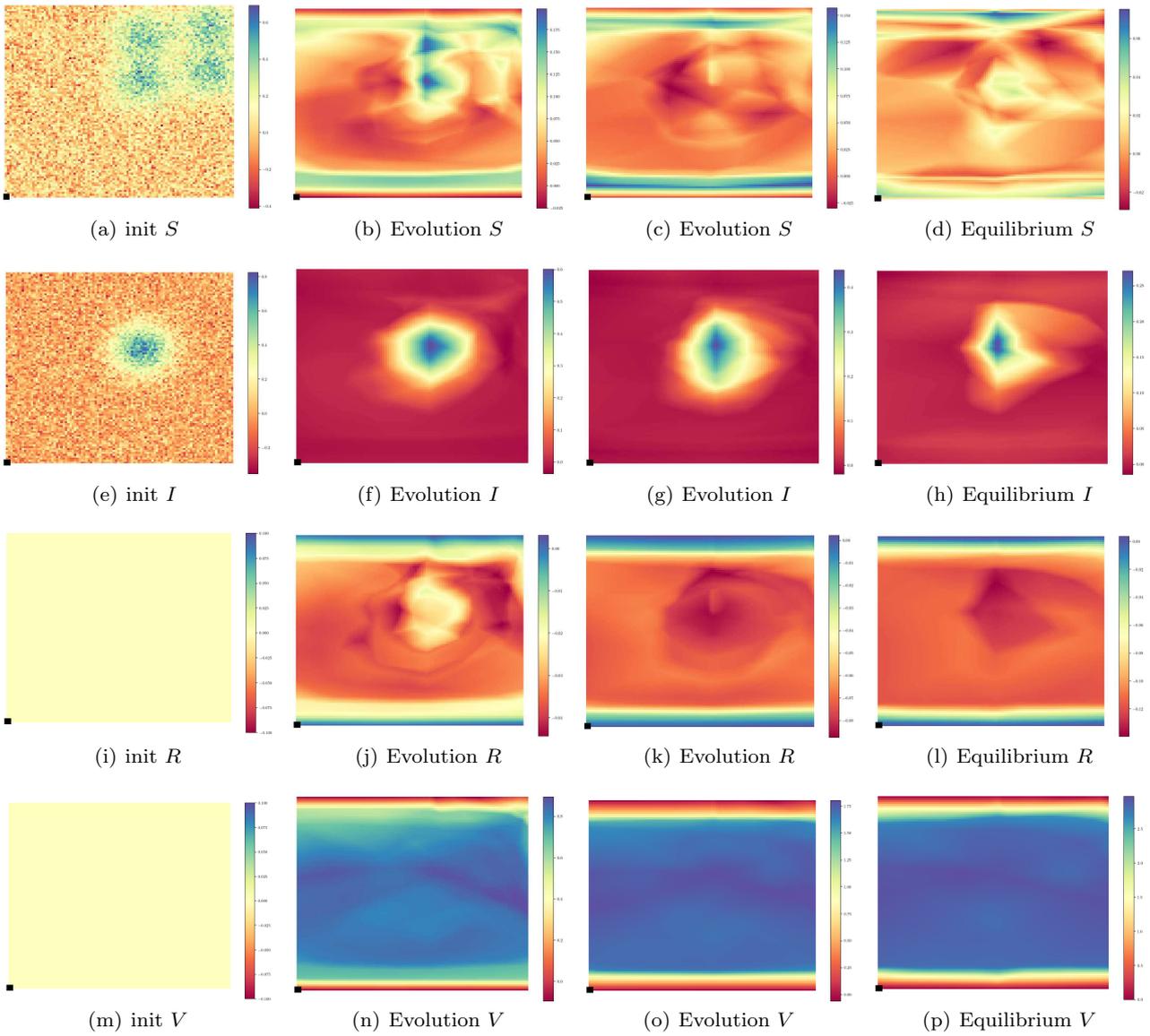


Fig. 4: The visual images under initial condition 3 (TSC noise level is 0.9).

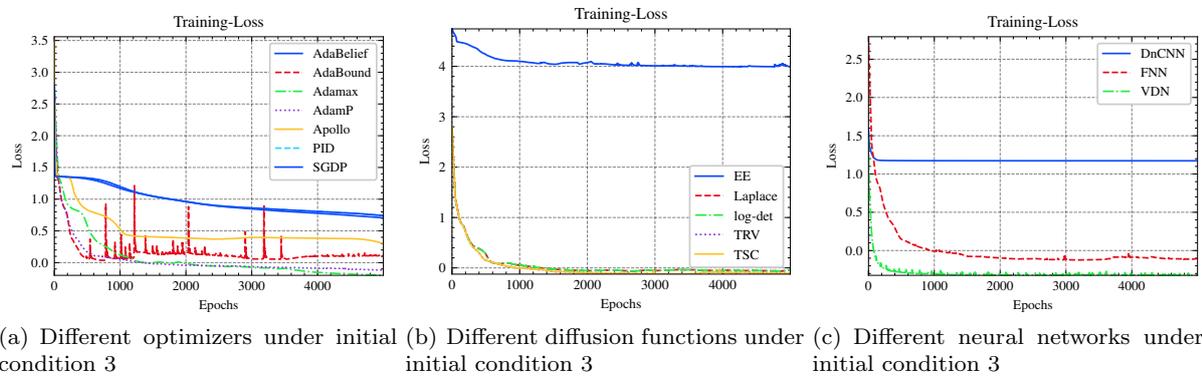


Fig. 5: Training loss curves in different situations.

see that although the susceptible population is gradually decreasing, the number of infected tends to increase during the evolution of the virus. However, it is effectively controlled in the end. At the same time, the number of recovered and vaccinated are also increasing rapidly. For initial condition 3, from Fig. 3 we can find that in addition to the substantial increase in vaccine production coverage, the susceptible population is gradually reduced, and the number of infected is quickly controlled. Meanwhile, the number of recovered and vaccinated have increased significantly.

To test the robustness of the system we designed, we take the TSC and initial condition 3 as examples. We add different levels of Gaussian white noise (0.1, 0.5, and 0.9) to the initial condition to perform different diffusion methods of the system from the noisy data. In the test, it can be seen from Table 3, Figs. 4-5 that TSC and TRV are more suitable for the diffusion methods of this system. Meanwhile, since the system involves learning the solutions of PDEs from noise data, we use two denoising NNs as the baseline. From (c) in Fig. 4, we can see that the computational cost of DnCNN and VDN are relatively high. Through experiments, it can be known that the feed-forward NN is more suitable for this system. At the same time, exploring different optimization solvers is more suitable for solving the system. We have tested the following optimization methods, including AdaBelief [42], AdaBound [43], AdamP [44], Apollo [45], PID [46], SGDP [44], and AdamX [47]. From (a) in Fig 4 and Table 5 we can see that except for the high computational cost of AdaBelief [42], AdaBound [43], and Apollo [45], several other optimization solvers are suitable for solving the system.

5 Conclusion

In this work, data-driven and optimal control methods are used for investigating the dynamics of infection spreading via a population. In the modeling stage, we give different optimal strategies according to different situations and give corresponding numerical simulations for each model. We use the PINNs algorithm that solves the spatial domain dynamical systems for highly infectious diseases. Additionally, we obtain different numerical approximation solutions from the perspective of numerical analysis and carried out simulation experiments. In the early stage of successful vaccine development, when only a small amount of vaccines were put on the market, the relevant governments should adopt a mixed policy, namely, some susceptible people are injected with vaccines and a small number of patients are isolated-controlled. This makes the disease effectively controlled quickly.

At the beginning of the outbreak of infectious diseases, it is necessary to control those few patients. Extreme policies will force companies and factories to halt production, which impacts all sectors of the economy and causes social panic. Therefore, it is very crucial to find a balance between disease prevention and ensuring normal social function. Government departments need to propose the so-called optimal control policies when formulating strategies.

Acknowledgements Thank you for your valuable comments from anonymous reviewers.

Funding This Project is funded by China Postdoctoral Science Foundation (No.2021M690837).

Compliance with ethical standards

Conflict of interest The authors declare that there is no conflict of interest regarding the publication of this paper.

Open Access This article is permitted to use, share, adapt, distribute, and reproduce in any medium or format, as long as you give appropriate credit to the original author(s) and the source. The images and other third-party material in this article are included in the papers.)

Data Availability Statement All materials are available from GitHub (<https://github.com/lichun0503/LSDML>).

References

1. D. Wang, B. Hu, C. Hu, F. Zhu, X. Liu, J. Zhang, B. Wang, H. Xiang, Z. Cheng, Y. Xiong, *et al.*, "Clinical Characteristics of 138 Hospitalized Patients with 2019 Novel Coronavirus-Infected Pneumonia in Wuhan, China," *Jama*, vol. 323, no. 11, pp. 1061–1069, 2020.
2. C. Sohrabi, Z. Alsafi, N. O'Neill, M. Khan, A. Kerwan, A. Al-Jabir, C. Iosifidis, and R. Agha, "World Health Organization Declares Global Emergency: A Review of the 2019 Novel Coronavirus (COVID-19)," *International Journal of Surgery*, vol. 76, pp. 71–76, 2020.
3. Y. Ding, Z. Wang, and H. Ye, "Optimal Control of a Fractional-Order HIV-Immune System with Memory," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 3, pp. 763–769, 2011.
4. C. Li, Y. Yang, H. Liang, and B. Wu, "Robust PCL Discovery of Data-Driven Mean-Field Game Systems and Control Problems," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 6, pp. 2651–2664, 2021.
5. Y.-C. Chen, P.-E. Lu, C.-S. Chang, and T.-H. Liu, "A Time-Dependent SIR Model for COVID-19 with Undetectable Infected Persons," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 3279–3294, 2020.
6. N. Zheng, S. Du, J. Wang, H. Zhang, W. Cui, Z. Kang, T. Yang, B. Lou, Y. Chi, H. Long, *et al.*, "Predicting COVID-19 in China Using Hybrid AI Model," *IEEE*

- Transactions on Cybernetics*, vol. 50, no. 7, pp. 2891–2904, 2020.
7. L. Wang, J. Chen, and M. Marathe, “DEFISI: Deep Learning based Epidemic Forecasting with Synthetic Information,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 9607–9612, 2019.
 8. H. Nishiura, N. M. Linton, and A. R. Akhmetzhanov, “Serial Interval of Novel Coronavirus (COVID-19) Infections,” *International Journal of Infectious Diseases*, vol. 93, pp. 284–286, 2020.
 9. D. H. Morris, F. W. Rossine, J. B. Plotkin, and S. A. Levin, “Optimal, Near-Optimal, and Robust Epidemic Control,” *Communications Physics*, vol. 4, no. 1, pp. 1–8, 2021.
 10. Z. Gu, L. Wang, X. Chen, Y. Tang, X. Wang, X. Du, M. Guizani, and Z. Tian, “Epidemic Risk Assessment by a Novel Communication Station based Method,” *IEEE Transactions on Network Science and Engineering*, vol. 8, pp. 1–1, 2021.
 11. C. Kitsos, G. Besancon, and C. Prieur, “High-Gain Observer Design for a Class of Quasi-Linear Integro-Differential Hyperbolic Systems-Application to an Epidemic Model,” *IEEE Transactions on Automatic Control*, vol. 66, pp. 1–1, 2021.
 12. J. Chen, Y. Huang, R. Zhang, and Q. Zhu, “Optimal Curing Strategy for Competing Ppidemics Spreading over Complex Networks,” *IEEE Transactions on Signal and Information Processing over Networks*, vol. 7, pp. 294–308, 2021.
 13. M. Ye, J. Liu, B. D. Anderson, and M. Cao, “Applications of the Poincare-Hopf Theorem: Epidemic Models and Lotka-Volterra Systems,” *IEEE Transactions on Automatic Control*, vol. 66, pp. 1–1, 2021.
 14. W. O. Kermack and A. G. McKendrick, “A Contribution to the Mathematical Theory of Epidemics,” *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, vol. 115, no. 772, pp. 700–721, 1927.
 15. P.-A. Bliman, M. Duprez, Y. Privat, and N. Vauchelet, “Optimal Immunity Control and Final Size Minimization by Social Distancing for the SIR Epidemic Model,” *Journal of Optimization Theory and Applications*, vol. 189, no. 2, pp. 408–436, 2021.
 16. M. Chen, M. Li, Y. Hao, Z. Liu, L. Hu, and L. Wang, “The Introduction of Population Migration to SEIAR for COVID-19 Epidemic Modeling with an Efficient Intervention Strategy,” *Information Fusion*, vol. 64, pp. 252–258, 2020.
 17. N. Zheng, S. Du, J. Wang, H. Zhang, W. Cui, Z. Kang, T. Yang, B. Lou, Y. Chi, H. Long, M. Ma, Q. Yuan, S. Zhang, D. Zhang, F. Ye, and J. Xin, “Predicting COVID-19 in China Using Hybrid AI Model,” *IEEE Transactions on Cybernetics*, vol. 50, pp. 2891–2904, 2020.
 18. R. Cui, H. Li, R. Peng, and M. Zhou, “Concentration Behavior of Endemic Equilibrium for a Reaction–Diffusion–Advection SIS Epidemic Model with Mass Action Infection Mechanism,” *Calculus of Variations and Partial Differential Equations*, vol. 60, no. 5, pp. 1–38, 2021.
 19. G. Fabricius and A. Maltz, “Exploring the Threshold of Epidemic Spreading for a Stochastic SIR Model with Local and Global Contacts,” *Physica A: Statistical Mechanics and its Applications*, vol. 540, p. 123208, 2020.
 20. H. Nishiura, N. M. Linton, and A. R. Akhmetzhanov, “Serial Interval of Novel Coronavirus (COVID-19) Infections,” *International Journal of Infectious Diseases*, vol. 93, pp. 284–286, 2020.
 21. W. Lee, S. Liu, H. Tembine, W. Li, and S. Osher, “Controlling Propagation of Epidemics via Mean-Field Control,” *SIAM Journal on Applied Mathematics*, vol. 81, no. 1, pp. 190–207, 2021.
 22. W. Lee, S. Liu, W. Li, and S. Osher, “Mean Field Control Problems for Vaccine Distribution,” *arXiv preprint arXiv:2104.11887*, 2021.
 23. I. E. Lagaris, A. Likas, and D. I. Fotiadis, “Artificial Neural Networks for Solving Ordinary and Partial Differential Equations,” *IEEE Transactions on Neural Networks*, vol. 9, no. 5, pp. 987–1000, 1998.
 24. M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations,” *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.
 25. L. Yang, X. Meng, and G. E. Karniadakis, “B-PINNs: Bayesian Physics-Informed Neural Networks for Forward and Inverse PDE Problems with Noisy Data,” *Journal of Computational Physics*, vol. 425, p. 109913, 2021.
 26. M. Eliasof, E. Haber, and E. Treister, “PDE-GCN: Novel Architectures for Graph Neural Networks Motivated by Partial Differential Equations,” *Advances in Neural Information Processing Systems*, vol. 34, 2021.
 27. A. Krishnapriyan, A. Gholami, S. Zhe, R. Kirby, and M. W. Mahoney, “Characterizing Possible Failure Modes in Physics-Informed Neural Networks,” *Advances in Neural Information Processing Systems*, vol. 34, 2021.
 28. G.-J. Both, S. Choudhury, P. Sens, and R. Kusters, “DeepMoD: Deep Learning for Model Discovery in Noisy Data,” *Journal of Computational Physics*, vol. 428, p. 109985, 2021.
 29. L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis, “DeepXDE: A Deep Learning Library for Solving Differential Equations,” *SIAM Review*, vol. 63, no. 1, pp. 208–228, 2021.
 30. J. Sirignano, J. F. MacArt, and J. B. Freund, “DPM: A Deep Learning PDE Augmentation Method with Application to Large-Eddy Simulation,” *Journal of Computational Physics*, vol. 423, p. 109811, 2020.
 31. L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, “Learning Nonlinear Operators via DeepONet based on the Universal Approximation Theorem of Operators,” *Nature Machine Intelligence*, vol. 3, no. 3, pp. 218–229, 2021.
 32. S. Wang, H. Wang, and P. Perdikaris, “Learning the Solution Operator of Parametric Partial Differential Equations with Physics-Informed DeepONets,” *Science Advances*, vol. 7, 2021.
 33. G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, “Physics-Informed Machine Learning,” *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, 2021.
 34. G. Aubert and L. Vese, “A Variational Method in Image Recovery,” *SIAM Journal on Numerical Analysis*, vol. 34, no. 5, pp. 1948–1979, 1997.
 35. B. Li and Q. Bie, “Long-Time Dynamics of an SIRS Reaction-Diffusion Epidemic Model,” *Journal of Mathematical Analysis and Applications*, vol. 475, no. 2, pp. 1910–1926, 2019.
 36. D. Mumford, “Elastica and Computer Vision,” in *Algebraic Geometry and Its Applications*, pp. 491–506, Springer, 1994.
 37. A. Chambolle and T. Pock, “Total Roto-Translational Variation,” *Numerische Mathematik*, vol. 142, no. 3, pp. 611–666, 2019.

38. Z. Kang, C. Peng, and Q. Cheng, "Robust PCA via Non-convex Rank Approximation," in *2015 IEEE International Conference on Data Mining*, pp. 211–220, IEEE, 2015.
39. Z. Kang, C. Peng, J. Cheng, and Q. Cheng, "Logdet Rank Minimization with Application to Subspace Clustering," *Computational Intelligence and Neuroscience*, vol. 2015, 2015.
40. Y. Chen, Y. Guo, Y. Wang, D. Wang, C. Peng, and G. He, "Denoising of Hyperspectral Images Using Non-convex Low Rank Matrix Approximation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 9, pp. 5366–5380, 2017.
41. O. Calin, *Deep Learning Architectures*. Springer, 2020.
42. J. Zhuang, T. Tang, Y. Ding, S. Tatikonda, N. Dvornik, X. Papademetris, and J. S. Duncan, "Adabelief Optimizer: Adapting Stepsizes by the Belief in Observed Gradients," *Advances in Neural Information Processing Systems*, 2020.
43. L. Luo, Y. Xiong, Y. Liu, and X. Sun, "Adaptive Gradient Methods with Dynamic Bound of Learning Rate," *7th International Conference on Learning Representations, ICLR*, 2019.
44. B. Heo, S. Chun, S. J. Oh, D. Han, S. Yun, G. Kim, Y. Uh, and J.-W. Ha, "AdamP: Slowing Down the Slowdown for Momentum Optimizers on Scale-Invariant Weights," *9th International Conference on Learning Representations, ICLR*, 2021.
45. X. Ma, "Apollo: An Adaptive Parameter-Wise Diagonal Quasi-Newton Method for Nonconvex Stochastic Optimization," *arXiv preprint arXiv:2009.13586*, 2020.
46. W. An, H. Wang, Q. Sun, J. Xu, Q. Dai, and L. Zhang, "A PID Controller Approach for Stochastic Optimization of Deep Networks," pp. 8522–8531, 2018.
47. D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *3rd International Conference on Learning Representations, ICLR*, 2015.
48. K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.
49. Z. Yue, H. Yong, Q. Zhao, L. Zhang, and D. Meng, "Variational Denoising Network: Toward Blind Noise Modeling and Removal," *Advances in Neural Information Processing Systems*, 2019.
50. B. Yu, L. Zhou, L. Wang, Y. Shi, J. Fripp, and P. Bourgeat, "Ea-GANs: Edge-aware Generative Adversarial Networks for Cross-Modality MR Image Synthesis," *IEEE Transactions on Medical Imaging*, vol. 38, no. 7, pp. 1750–1762, 2019.
51. T. Chai and R. R. Draxler, "Root Mean Square Error (RMSE) or Mean Absolute Error (MAE)?—Arguments Against Avoiding RMSE in the Literature," *Geoscientific Model Development*, vol. 7, no. 3, pp. 1247–1250, 2014.
52. A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., "Pytorch: An Imperative Style, High-Performance Deep Learning Library," *Advances in Neural Information Processing Systems*, vol. 32, pp. 8026–8037, 2019.