DDE: Deep Dynamic Epidemiological Modelling for Infectious Illness Development Forecasting in Multi-level Geographic Entities

Ruhan Liu
Central South University
Jiajia Li
lijiajia@sjtu.edu.cn
Shanghai Jiao Tong University
Yang Wen
Shenzhen University
Huating Li
Shanghai Sixth People's Hospital
Ping Zhang
The Ohio State University
Bin Sheng
Shanghai Jiao Tong University
David Dagan Feng
The University of Sydney

Research Article

Keywords: COVID-19, SEIR/SIR model, neural ODE, epidemiological equations

Posted Date: October 9th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3407592/v1

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Journal of Healthcare Informatics Research on May 28th, 2024. See the published version at https://doi.org/10.1007/s41666-024-00167-4.

DDE: Deep Dynamic Epidemiological Modelling for Infectious Illness Development Forecasting in Multi-level Geographic Entities

Ruhan Liu[1,2,3], Jiajia Li[4*], Yang Wen[5], Huating Li[6], Ping Zhang[7,8], Bin Sheng[9], David Dagan Feng[10]

[1]National Engineering Research Center of Personalized Diagnostic and Therapeutic Technology, Furong Laboratory, Central South University, Changsha, 410012, Hunan, China.
[2]Department of Dermatology, Xiangya Hospital, Central South University, Changsha, 410008, Hunan, China.
[3]Hunan Key Laboratory of Skin Cancer and Psoriasis, Hunan Engineering Research Center of Skin Health and Disease, Xiangya Hospital, Changsha, 410008, Hunan, China.
[4]School of Chemistry and Chemical Engineering and National Center for Translational Medicine, Shanghai Jiao Tong University, Shanghai, 200240, Shanghai, China.
[5]College of Electronics and Information Engineering, Shenzhen University, Shenzhen, 518060, Guangdong, China.
[6]Department of Endocrinology and Metabolism, Shanghai Jiao Tong University Affiliated Sixth People's Hospital, Shanghai, 200233, Shanghai, China.
[7]Department of Computer Science and Engineering, The Ohio State University, Columbus, 43210, Ohio, USA.
[8]Department of Biomedical Informatics, The Ohio State University, Columbus, 43210, Ohio, USA.
[9]Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, 200240, Shanghai, China.
[10]School of Computer Science, The University of Sydney, Sydney, 410008, New South Wales, Australia.

*Corresponding author(s). E-mail(s): lijiajia@sjtu.edu.cn;

Contributing authors: 223101@csu.edu.cn; wen_yang@szu.edu.cn;
huarting99@sjtu.edu.cn; zhang.10631@osu.edu; shengbin@sjtu.edu.cn;
dagan.feng@sydney.edu.au;

Abstract

Understanding and effectively addressing the dynamics of infectious diseases, including global diseases like COVID-19, is crucial for managing the current situation and developing effective intervention strategies. Epidemiologists commonly use epidemiological equations (EE) to model disease progression. Nevertheless, the traditional approach to parameter estimation in EE frequently faces challenges in accurately fitting real-world data, primarily due to factors like the different implementation of social distancing policies and intervention strategies. However, developing high-quality but complex EE models can be time-intensive for epidemiologists. Hence, we introduce a novel method known as the deep dynamic epidemiological (DDE) approach, which integrates the strengths of EE with the capabilities of deep neural networks (DNN) to enhance accuracy. The DDE method incorporates DNNs to model dynamic effects and adapt to evolving situations. It employs the neural ordinary differential equation (ODE) method to solve variant-specific equations, ensuring a precise fit to disease progression in diverse geographic regions. In this study, we introduce four variants of EE customized to address specific scenarios in different countries and regions. We assess the performance of our DDE method using real-world data from five diverse geographic entities (countries: the USA, Colombia, and South Africa; regions: Wuhan in China and Piedmont in Italy). We show that the DDE method outperforms alternative approaches, achieving the highest predictive accuracy in modeling disease progression across all five geographic entities. This paves the way for constructing a simplified EE model for various geographic levels.

Keywords: COVID-19, SEIR/SIR model, neural ODE, epidemiological equations

1 Introduction

Coronavirus Respiratory Disease 2019 (COVID-19) from the virus "SARS-CoV-2" has a catastrophic spread and influenced at least 214 countries and territories over seven continents, resulting in more than 5 million people infected and over 3456 thousand deaths. The disease has brought unprecedented impact on people's health and safety around the world and has influenced economic and social development. Due to different conditions such as quarantine measurement, social distance, population density, and medical conditions, the development of COVID-19 in different places is also diverse [1, 2]. Because of the various situations in different regions, constructing a uniform framework to analyze the disease's condition and estimate future development is crucial for analyzing the spread condition and helping control the outbreak of coronavirus disease in different level regions.

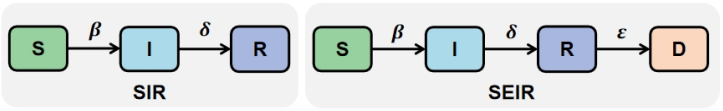

Fig. 1 The architectures of Susceptible-Infected-Removed (SIR) and Susceptible-Exposed-Infected-Removed (SEIR) models. The left figure and right figure show the SIR model and the SEIR model, respectively. β in the figures represents infectious rate, γ is the exposed rate, and δ shows the recovery rate.

Many studies propose models to simulate the development of diseases: The Susceptible-Infected-Removed (SIR) and Susceptible-Exposed-Infected-Removed (SEIR) [3] models are widely used by epidemiologists due to easy modeling, which only use the infectious rate, exposed rate, recovered rate to describe the development trend of infectious diseases. However, because the infection situation in the real world is complicated, existing parameter estimation methods cannot achieve high fitting performance using SIR and SEIR [3]. Some studies based on SIR or SEIR models further propose modeling improvements to fit the different population trends [4–6]. These works often use a complex improved SEIR model with multiple parameters to analyze a specific area to achieve high-precision fitting performance. However, due to each region's particularities (different population conditions, social distance, isolation measures, etc.), it is difficult to directly migrate these models to other areas. Developing a different model for a specific region is very time-consuming, but precise modeling can help estimate the spread of COVID-19 diseases. Besides, because the infection situations between different areas are also very inconsistent (some countries have just begun to spread while the infection cases in others have disappeared), different areas may use different models to analyze. Therefore, simple models with high fitting accuracy that are more generally adapted to various countries or regions and more easily migrated to another area are of great significance in the modeling of COVID-19.

Machine learning has achieved remarkable results in solving many complex data-driven problems such as medical data prediction. Thus, it is also used to model the COVID-19 data [7–11], offering great data fitting capabilities in early data modeling. However, the forecasts under different development situations in the middle and late periods have not been explored. Furthermore, these machine learning models' black-box mechanism allows their users to obtain the predicted results directly. However, it is unclear what the reason is the prediction. Thus, these models have low explanability, and their prediction process cannot be analyzed and quantified.

Due to the limitations of existing approaches, we propose a deep-learning model called deep dynamic epidemiological (DDE), which combines neural ordinary differential equations (Neural ODE) [12] and epidemiological equations in COVID-19 data fitting. Because of different countries and regions' characteristics, we propose four variant models based on SIR and SEIR models, namely SIRD, SEIRD, SMCRD, and SEMCRD. These four models consider the population group division's situation and can deal with different data sources (for example, some countries' published information does not give a specific number of mild and severe cases). Secondly, we developed the DDE implemented on these four variants to solve the equations. In our DDE

model, we based on the Neural ODE, a novel algorithm, which can use numerical solutions of ordinary differential equations to build networks and complete data fitting and modeling. Also, we design an additional neural network under Neural ODE solving to fit the effect function to reflect better the impact of the diversity of regional and national intervention policies on the infection rate. Therefore, based on understanding the parameters, including infection rate, recovery rate, mortality rate, and so on, involved in the SEIR model, it can obtain outstanding solution accuracy. The DDE is used to solve the problem, which can better consider the impact of different isolation policies and realize the possibility of designing a universal model. Specifically, the contributions of this paper are summarised as follows.

1. We propose the DDE method that can easily integrate neural networks (NN) and Neural ODE to solve SEIR-like equations. It generates infection, mortality, and recovery rates through network training and can visualize the change in those rates. Moreover, DDE has achieved a Pearson correlation coefficient above 0.98 in all data.
2. We design four SIR/SEIR variants: SIRD, SEIRD, SMCRD, and SEMCRD, which can adapt to regions and countries' diversity. Further, the DDE method implemented on the four variants is more precise than traditional parameter estimation methods.
3. We compare the performance of the DDE method with other learning-based models. The DDE method gains average Pearson coefficients higher than 0.85. Further, we analyze the output trend of NN and show the real-world correlation of the trend.

2 Related Work

2.1 SIR/SEIR and their variants

SIR and SEIR models are commonly used in epidemiological analysis. In the SEIR model, the country's population is divided into four parts: susceptible people S, exposed people E, infected patients I, and recovered group R, and their relative growths are based on a set of coupled ordinary differential equations. The SIR model is not considered the exposed group E. In the SARS and MARS, which once caused the global epidemic, many existing studies are based on the SEIR model to model and analyze infectious diseases [3, 13, 14]. Moreover, in COVID-19 studies, many works focus on improved SEIR modeling based on regions or cities. L. Peng et al. [4] proposed a generalized SEIR model to analyze the COVID-19 epidemic in China. Additionally, [6] Choi's work introduced an improved SEIR model called the SEIAQIm model based on Korea's data. These improved SEIR models are aimed at a specific country or region and have quite complex parameter designs.

2.2 Statistical methods

Statistical models are widely used in specific aspects modeling of real-world data. For example, in Kraemer's work [15], the author used Generalized linear models (GLM) to analyze the effect of human mobility and control measures in early Wuhan, China. Furthermore, in [16], a global epidemic and mobility model (GLEAM) is proposed to analyze the effect of travel restrictions on the spread of COVID-19 in the world. This

work used experienced parameters obtained in SARS or MARS coronavirus epidemiology instead of independently from COVID-19 data. Moreover, work [17] focuses on the impact of cases exported from Wuhan on other regions using a stochastic transmission dynamic model. The widely used statistical models have brought valuable solutions to the analysis of some specific aspects, but the models are diverse, and the designs are complex.

2.3 Machine-learning methods

Some studies on the COVID-19 data trend prediction issue have also introduced machine learning attempts due to the SEIR models and statistical methods' deficiencies. Z. Yang's work [8] integrated the population migration data before and after January 23 and the latest COVID-19 epidemiological data into the SEIR model to derive the epidemic curve. Also, it used artificial intelligence (AI) methods trained on 2003 SARS data to predict the epidemic. Besides, F. Rustam et al.[7] proposed to use of some machine learning models including linear regression (LR), the least absolute shrinkage and selection operator (LASSO), support vector machine (SVM), and exponential smoothing (ES) to predict the population of infection, recovery, and death. These attempts either used previous infectious disease data for training, or the models used were insufficient in terms of interpretability. Moreover, in work [18], researchers tried to use the Neural ODE method to stimulate data change in the infection and recovery group. However, this method only explores the SIR model's improved performance and the fit of real data. This method has only been experimentally studied during the early development of COVID-19 (before April) and has not compared model performance with other methods. Furthermore, the way does not analyze the data fitting situation of different size regions (countries, regions, and cities). Therefore, these factors also hinder the application of the advanced Neural ODE method in practice.

3 Method

In this section, we propose a data-driven model for real-world infection data fitting. First, we introduce four variants based on SEIR and SIR models for better fitting of COVID-19 data. Next, the DDE method is illustrated for dynamic parameter estimation. In our DDE, multi-layer neural networks are designed to stimulate intervention influence function to assess the effect of quarantine policies in different countries. Furthermore, Fig 2 illustrates the computing process in our DDE method for dynamic parameter estimation.

The SIR model divides the population into three groups: the susceptible (S), the infectious (I), and the recovered (R). The S represents healthy people, and the I is those who have been infected. Also, the R is for those who have recovered from the infection. In the SEIR model modeling, the exposed group (E) who may be infected is further considered. The formula of SEIR is shown below:

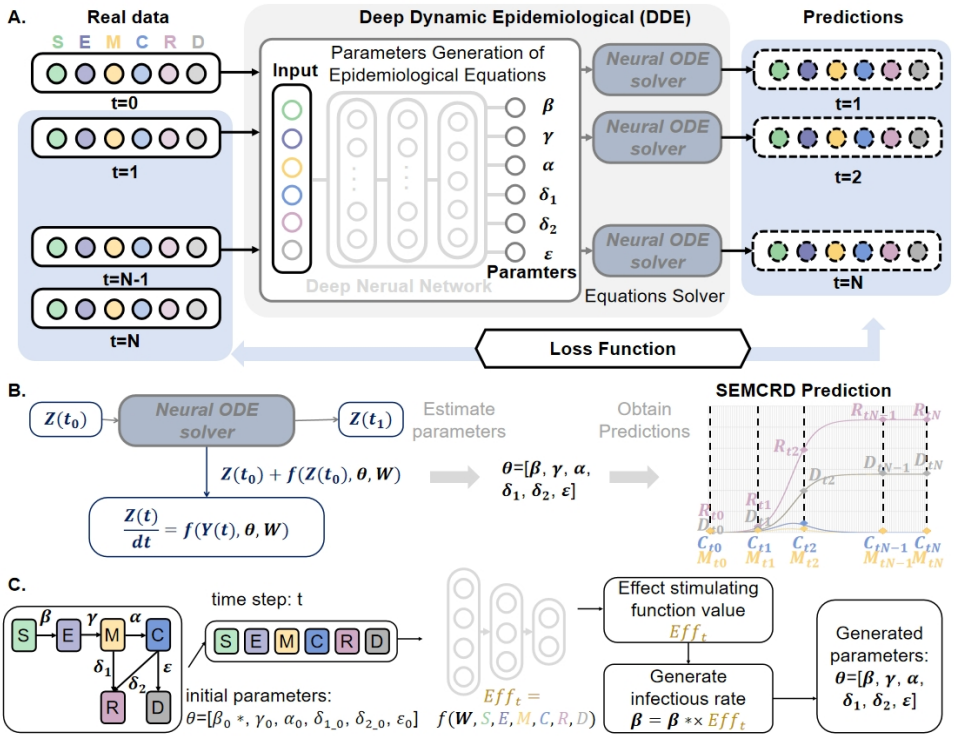

Fig. 2 The schematic diagram shows the system structure of the DDE model. **A.** The method architecture is shown in Part A. The real data are represented by solid frames, and the predictions are shown by dotted frames. The system consists of four parts: First, we input the value of the equations at the current moment into the parameter generation network to obtain the corresponding parameters. After that, the generated parameters and current values are used to obtain the function value at the next moment through the Neural ODE solver. Then, repeat the process several times until the solution obtains the predicted value. Finally, we calculate the loss function between the predicted value and the true value to obtain the best prediction through gradient descent. **B.**The schematic plot illustrates the process of solving the epidemiological equations. **C.** The process figure shows the details of generating parameters.

$$
\begin{aligned}
\frac{dS}{dt} &= -\beta \cdot \frac{S \cdot I}{N} \\
\frac{dE}{dt} &= \beta \cdot \frac{S \cdot I}{N} - \gamma \cdot E \\
\frac{dI}{dt} &= \gamma \cdot E - \delta \cdot I \\
\frac{dR}{dt} &= \delta \cdot I
\end{aligned}
\tag{1}
$$

Where β is the infectious rate, γ is the exposed rate, δ is the recovery rate, and $N = S + I + E + R$ is the number of the total population.

3.1 Four variants: SIRD, SEIRD, SMCRD, and SEMCRD

The classic SEIR model has been employed in countless prior studies [3, 13, 14]. In research about COVID-19, the SEIR model is also prevalent [19–21]. According to the characteristics of COVID-19, we consider the following aspects of improving the initial SEIR and SIR models:

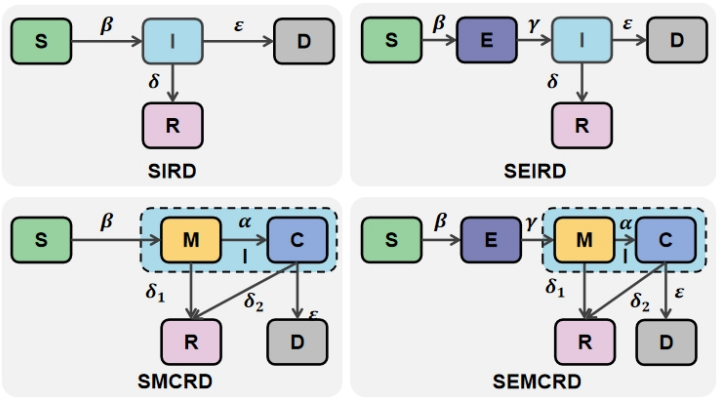

Fig. 3 The architectures display the four variants based on SEIR and SIR models: SIRD, SMCRD, SEIRD, SEMCRD. In the SIRD model, the infection rate β, recovery rate δ, and mortality rate ε are considered. Furthermore, the SEIRD model takes the expose group E into account and assumes the exposed rate as γ. The SMCRD and SEMCRD models divide the infection group into mild patient M and critical patient C. Thus, the probability of transferring mild patients to severe patients is α, the recovery rate of the mild patient is δ_1, and the recovery rate of the critical patient is δ_2.

1. We make a new way to classify the rehabilitation population (including death and disease rehabilitation). The original recovery group is divided into the death population and the recovery (disease rehabilitation) population. We propose two parameters (mortality and recovery rate) to better respond to different countries' death and recovery situations due to different medical conditions.
2. We accurately define the infected population as the mild population (including asymptomatic infected persons, self-recovering mild patients) and critical patients (including severe patients who need to be admitted to the hospital). Further, we consider three parameters: the mild infectious rate, the transition rate from mild to critical, the mild recovery rate, and the critical recovery rate, helping us better simulate real infections.

Therefore, we implement four variants: SIRD, SEIRD, SMCRD, and SEMCRD. Fig. 3 shows the model structures of the four models. As shown in Fig. 3, compared with the SIR and SEIR models, the SIRD and SEIRD models add the consideration of the death population, which can help understand the death growth caused by diseases. On the other hand, SMCRD and SECRD are based on the SIRD and SEIRD models and further divide the infection group I into mild and critical cases. The SMCRD and SEMCRD models are in line with the characteristics of the COVID-19 disease and can be better modeled in real-world data.

The equation of these models can be seen as follows:

$$\frac{d\mathbf{Z}(t)}{dt} = \mathscr{F}(\mathbf{Z}(t), t, \theta), \text{ with } \mathbf{Z}(t_0) = \mathbf{Z}_0 \tag{2}$$

where $t \in \{t_0, ..., t_i, ..., t_T\}$ (t_0 stands for the initial day, and t_i represents the i^{th} day from t_0), $\mathbf{Z}(t) \in \mathbb{R}^D$, $\mathbf{Z}_0 = [S_0, E_0, I_0, R_0, D_0]$ for SEIRD model, $\mathbf{Z}_0 = [S_0, I_0, R_0, D_0]$ for SIRD model, $\mathbf{Z}_0 = [S_0, E_0, M_0, C_0, R_0, D_0]$ for SEMCRD model, and $\mathbf{Z}_0 = [S_0, M_0, C_0, R_0, D_0]$ for SMCRD model, $\mathbf{Z}_0$ is the function value at time t_0. The $\mathscr{F}(\cdot)$ is a known and continuous function with parameter θ, and $\mathbf{Z}(t)$ is the unknown function that must be approximated. For the SIRD model, the parameter set θ includes $\beta, \delta, \varepsilon$. The SEIRD model's parameter set θ includes $\beta, \gamma, \delta, \varepsilon$. Further, for SMCRD model, θ includes $\beta, \delta_1, \delta_2, \alpha, \varepsilon$. In SEMCRD model, θ includes $\beta, \gamma, \delta_1, \delta_2, \alpha, \varepsilon$. As Fig. 3 shows, the epidemiological equations for the SEMCRD model can be seen in the bottom right part, and the equation set in the SMCRD model is in the bottom left part. The difference between these two models is that the SMCRD model does not consider the exposed group E.

From a computational point of view, knowing that $\mathbf{Z}(t_0) = \mathbf{Z}_0$, you can calculate the value of $\mathbf{Z}(t_i) = \mathbf{Z}_i$ in any step t_i by performing piecewise integration from previously known points:

$$\mathbf{Z}_i = \mathbf{Z}_{i-1} + \int_{t_{i-1}}^{t_i} \mathscr{F}(\mathbf{Z}_{i-1}, t_{i-1}, \theta) \text{ with } i \in \{1, ..., T\} \tag{3}$$

When $\Delta\tau = t_i - t_{i-1}$ is small enough, we can get the approximation result:

$$\mathbf{Z}_i = \mathbf{Z}_{i-1} + \Delta\tau \cdot \mathscr{F}(\mathbf{Z}_{i-1}, t_{i-1}, \theta) \tag{4}$$

Thus, in each time step t_i, the value of function $\mathbf{Z}_i$ can be obtained by deduction of the function value $\mathbf{Z}_{i-1}$ at the previous moment t_{i-1}.

3.2 DDE Method applying in SMCRD and SEMCRD models

In real-world data, the infection condition is always influenced by social distancing, quarantine measurement, and people's compliance. In recent SEIR model studies, the effect function based on the factors mentioned above is considered. In our DDE model, we also propose a way to stimulate the effect function: we design multi-layer neural networks to fit the effect function's value and use it as a dynamic parameter involved in the solution process of Neural ODE. Fig 4 illustrates how to use multi-layer neural networks in parameter designing. The estimation of the designed parameters is based on the Neural ODE method.

The neural ODE [12] method is a novel technique that uses reverse-mode differentiation (also known as backpropagation) to solve the ODE function. In our work, the ODE equation sets in SEMCRD and SMCRD models are shown above. Using the Neural ODE method in our task can make full use of neural networks' fitting

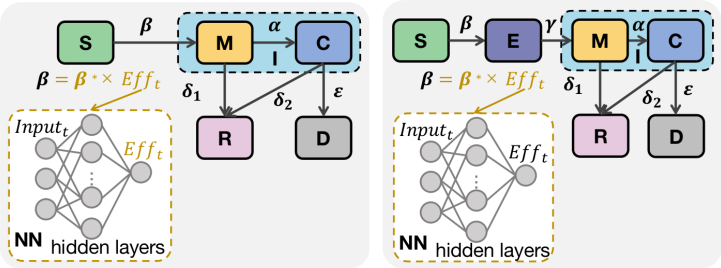

Fig. 4 The DDE method is implemented in SMCRD and SEMCRD models. In the DDE-SMCRD and DDE-SEMRD models, neural network (NN) models are used to calculate dynamic β (initial $\beta \times EEf_t$) based on the neural ODE method.

performance within the framework of epidemiological equations. Our model uses infection rate, recovery rate, mortality rate, and so on, which have real-world meanings as parameters. Also, we use neural networks to perform fitting to achieve high precision and high interpretability.

Neural networks are known for their strong fitting ability. Also, because of Neural ODE's characteristics, we could develop multi-layer neural networks as an influence function value stimulator that reflects the influence on infection rate. In the previous part, we propose four variants based on the SEIR or SIR model and give the design of parameters involved in SIRD, SEIRD, SMCRD, and SEMCRD. Further, in real data fitting, different countries and regions often adopt different intervention isolation measures and policies, and people respond to policies differently. The impact of these problems on the infection rate is also different. For example, in some countries, very strict movement bans have been implemented, which can highly impact the infection situation. The sharp drop in the number of contacts will also slow down the increase in the number of people infected with COVID-19. In countries that have not adopted control measures, the infection situation's impact may be small, and the number of infections will continue to increase. Thus, we develop an effect stimulation function to model the various changes in infection situations. The parameter set θ is changed:

$$\begin{aligned} \theta &= [\beta*, W, \delta, \varepsilon] \text{ , for SIRD} \\ \theta &= [\beta*, W, \gamma, \delta, \varepsilon] \text{ , for SEIRD} \\ \theta &= [\beta*, W, \delta_1, \delta_2, \alpha, \varepsilon] \text{ , for SMCRD} \\ \theta &= [\beta*, W, \gamma, \delta_1, \delta_2, \alpha, \varepsilon] \text{ , for SEMCRD} \end{aligned} \tag{5}$$

where the $\beta*$ is the infection rate which could be influenced by effect function Eff_t. The effect function Eff_t uses the multi-layer neural networks to fit. The W is the weight in the multi-layer neural networks. The structure of the neural networks is shown in Fig. 5.

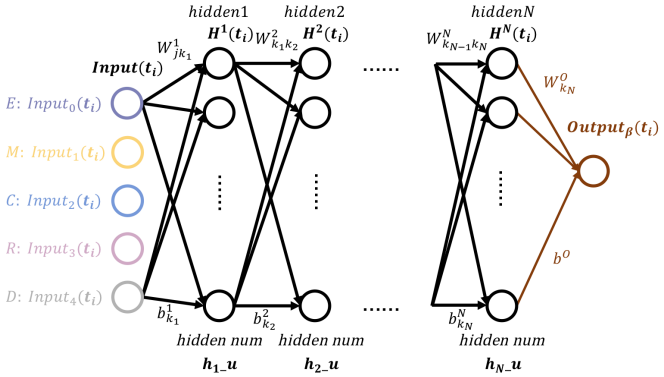

Fig. 5 The structure figure shows the multi-layer neural networks that are used to fit the effect function Eff_t on the dynamic infection rate β.

According to Fig. 5, we can obtain the output of the first hidden layer of the neural network as follows:

$$H^1_{k_1}(t_i) = a(\sum_{j=1}^{input_u} W^1_{j \cdot k_1} \cdot Input_j(t_i) + b^1_{k_1}) \tag{6}$$

where $H^1_{k_1}(t_i)$ is the ${k_1}^{th}$ of unit of the first hidden layer output in time step t_i, the input tensor $Input(t_i)$ =$[Input_0(t_i), Input_1(t_i), Input_2(t_i), Input_3(t_i), Input_4(t_i)]$ for SEMCRD ($Input_0$ is E_{t_i}, $Input_1(t_i)$ is M_{t_i}, $Input_2(t_i)$ is C_{t_i}, $Input_3(t_i)$ is R_{t_i}, and $Input_4(t_i)$ is D_{t_i}) and $Input(t)$ =$[Input_0(t_i), Input_1(t_i), Input_2(t_i), Input_3(t_i)]$ for SMCRD ($Input_0(t_i)$ is M_{t_i}, $Input_1(t_i)$ is C_{t_i}, $Input_2(t_i)$ is R_{t_i}, and $Input_3(t_i)$ is D_{t_i}), the $input_u$ is the number of $Input(t_i)$, it is 5 for SEMCRD model or is 4 for SMCRD model. The $W^1_{j \cdot k_1}$ represent the weight of the input layer and the first hidden layer in the connection weight of the ${j}^{th}$ input unit and the ${k_1}^{th}$ of the first hidden unit. The $b^1_{k_1}$ is the bias of the ${k_1}^{th}$ unit of the first hidden layer, and the $a(\cdot) = ELU(\cdot)$ is the activation function of the hidden layer.

The output of the 2^{rd} hidden layer to the N^{th} hidden layer also can be deduced in the following:

$$\begin{aligned} H^2_{k_2}(t_i) &= a(\sum_{k_1=1}^{h_1\_u} W^2_{k_1 \cdot k_2} \cdot H^1_{k_1}(t_i) + b^2_{k_2}) \\ &\cdots\cdots \\ H^N_{k_N}(t_i) &= a(\sum_{k_{N-1}=1}^{h_{N-1}\_u} W^N_{k_{N-1} \cdot k_N} \cdot H^{N-1}_{k_{N-1}}(t_i) + b^N_{k_N}) \end{aligned} \tag{7}$$

where $a(\cdot)$ is the activation function, and all hidden layers use the same activation function, h_{N-1}_u is the number of hidden units in the $N-1^{th}$ hidden layer, $b^N_{k_N}$ is

the bias of the ${k_N}^{th}$ unit of the N^{th} hidden layer, $W^N_{k_{N-1}\cdot k_N}$ represents the parameters between the $N-1^{th}$ hidden layer and N^{th} hidden layer. And, the $H^{N-1}_{k_{N-1}}(t_i)$ is the ${k_{N-1}}^{th}$ units of the $N-1^{th}$ hidden layer in time step t_i, and $H^N_{k_N}(t_i)$ is the ${k_N}^{th}$ units of the N^{th} hidden layer in time step t_i. Similarly, replace N with 2 to get the mean of $H^2_{k_2}(t_i)$, h_1_u, $W^2_{k_1\cdot k_2}$, b^2, and $H^1_{k_1}(t_i)$.

Thus, we can calculate the output of this network:

$$Output_\beta(t_i) = (1 + e^{-\sum_{k_N=1}^{h_N\_u} W^O_{k_N}\cdot H^N_{k_N}(t_i)+b^O})^{-1} \tag{8}$$

where $Output_\beta(t_i)$ in the i^{th} time step t_i as the output of N-layer NN, h_N_u is the number of hidden units in the N^{th} hidden layer, W^O represents the parameters between the N^{th} hidden layer and the output layer and the $W^O_{k_N}$ stands for the ${k_N}^{th}$ of the weight matrix W^O, $H^N(t_i)$ is the output of the N^{th} hidden layer in time step t_i and the $H^N_{k_N}(t_i)$ is the output of the ${k_N}^{th}$ of the hidden layer output, and the b^O stands for the bias of the output layer. Moreover, we can present a simper form of $Output_\beta(t_i)$ with weight of N hidden layer is $W = \{W^1, W^2, ..., W^N\}$, and W^O is the weight of output layer as:

$$\begin{aligned} Output_\beta(t_i) = a_O(W^O(a(W^N... \\ a(W^1 \cdot Input(t_i) + b^1)...) + b^N) + b^O) \end{aligned} \tag{9}$$

where the activation function of output layer is $a_O(\cdot)$, and the activation function of hidden layers is $a(\cdot)$. The bias $\{b^1, ..., b^N\}$ is of the 1^{th} to N^{th} of hidden layers.

The output of the NN represents the effect function Eff_t. To solve the ODE equations $\mathbf{Z}(t)$, we need to use an ODE solver. Efficient and accurate ODE solvers have been used for 120 years and can guarantee the growth of approximation error, monitor the level of error, and adapt their evaluation strategy on the fly to achieve the requested level of accuracy [12]. In specific time t_i, the function $\mathbf{Z}_i$ can be obtained into the form:

$$\begin{aligned} \mathbf{Z}_i = ODESlover(\mathbf{Z}_{i-1}, \mathscr{F}(\mathbf{Z}_{i-1}, t_{i-1}, \theta), t_{i-1}, \theta), \\ \text{with } \mathbf{Z}(t_0) = \mathbf{Z}_0 \end{aligned} \tag{10}$$

According to the equation (10), each $\mathbf{Z}_i$ can be calculated. Different models have various parameter sets θ. The specific content of θ has been given in Equ. 5.

Further, we design the loss function. The loss function L with multi-layer neural networks attempt to minimize the value of loss function by adjust parameters including $\beta*, W, \gamma, \delta_1, \delta_2, \alpha, \varepsilon$, and weight of the multi-layer NN for SEMCRD model, and $\beta*, W, \delta_1, \delta_2, \alpha, \varepsilon$, and weight of the multi-layer NN for SMCRD model. The simper

loss function is:

$$\begin{aligned} L(X,\hat{X}) &= \frac{1}{3}\sum_{t=0}^{T}\min_{\theta}\{(log(X_t)-log(\hat{X}_t))^2\} \\ &= \frac{1}{3}\sum_{t=0}^{T}\min_{\theta,W}\{||log(I_t)-log(\hat{M}_t+\hat{C}_t)||^2 \\ &+||log(R_t)-log(\hat{R}_t))||^2+||log(D_t)-log(\hat{D}_t)||^2\} \end{aligned} \tag{11}$$

Where $X=\{X_0,X_1,...,X_T\}$ represent real data considered in cost function in each time step and $\hat{X}=\{\hat{X}_0,\hat{X}_1,...,\hat{X}_T\}$ stands for predicted data per day. In the i^{th} of time t_i, the real data used in calculate loss is $X_i=[I_i,R_i,D_i]$, and the predicted data is $\hat{X}_i=[\hat{M}_i+\hat{C}_i,\hat{R}_i,\hat{D}_i]$ (I_i, R_i, and D_i represent the number of real infection, recovery, death group, relatively, and $\hat{M}_i$, $\hat{C}_i$, $\hat{R}_i$, and $\hat{D}_i$ stand for the number of predicted mild patients, critical patients, recovery, death group.).

To minimize L, understanding how the gradient of the loss depends on equation set $\mathbf{Z}(t)$. According to the chain rules, we can know that:

$$\begin{aligned} \frac{d\mathbf{a}(t)}{dt} &= -\mathbf{a}(t)^T\frac{\partial f(\mathbf{Z}(t),t,\theta)}{\partial \mathbf{Z}} \\ \text{where,}\mathbf{a}(t) &= \frac{\partial L}{\partial \mathbf{Z}(t)} \end{aligned} \tag{12}$$

The $\frac{\partial L}{\partial \mathbf{Z}(t_0)}$ can be computed by the *ODESolver* mention above. The *ODESolver* must start from the initial value of $\frac{\partial L}{\partial \mathbf{Z}(t_T)}$ and run backward. A complicated situation is that solving this ODE requires knowing the value of $\mathbf{Z}(t)$ along its entire trajectory. However, we can simply recalculate $\mathbf{Z}(t)$ from its final value $\mathbf{Z}(t_T)=\mathbf{Z}_T$ and work backward with time.

The gradients of loss function L depends on both $\mathbf{Z}(t)$ and $\mathbf{a}(t)$:

$$\frac{dL}{d\theta} = -\int_{t_T}^{t_0}\mathbf{a}(t)^T\frac{\partial f(\mathbf{Z}(t),t,\theta)}{\partial\theta}dt \tag{13}$$

The $\mathbf{a}(t)^T\frac{\partial \mathscr{F}}{\partial \mathbf{Z}}$, $\mathbf{a}(t)^T\frac{\partial \mathscr{F}}{\partial \theta}$ and $\mathscr{F}$ can be obtained through automatic differentiation of computing vector-Jacobian products. Finally, all gradients including $\frac{\partial L}{\partial \mathbf{Z}(t_0)}$ and $\frac{\partial L}{\partial \theta}$ can be calculated at once by calling the ODE solver. From the steps, we first define the initial state, and then design the dynamics system. After that, we compute vector-Jacobian products to obtain $\mathbf{a}(t)^T\frac{\partial \mathscr{F}}{\partial \mathbf{Z}}$, $\mathbf{a}(t)^T\frac{\partial \mathscr{F}}{\partial \theta}$ and $\mathscr{F}$. In the end, the *ODESolver* is used to solve the reverse-time SMCRD and SEMCRD equations and gain the gradients.

4 Experiments and Discussions

4.1 Implemental Details

In the experiment, we collected two kinds of data: countries and regions. In countries, three are used in model training and testing: the United States of America (USA), Columbia (CO), and South Africa (ZA). Moreover, two regions are considered in our model implementation: Wuhan City in China (WH) and Piedmont in Italy (PD). For data collection, we use data including the number of total accumulated infection cases, the number of disease recoveries, and the death number in [22]. The data include January 24^{th} to April 15^{th} for CN-WH and February 24^{th} to June 8^{th} for PD. In countries data collection, we have January 23^{rd} to August 12^{th} data for the USA, March 6^{th} to August 11^{th} for CO, and March 7^{th} to August 12^{th} for ZA. All statistical data are collected from official notifications of various countries, WHO, National Health Commission, etc. For model testing, we used the last 20 days' data to calculate the model performance, and the last data were used in model training.

We used the experienced initial value of parameters based on [23] because both traditional parameter estimation methods and our DDE algorithm need initialization. According to the experiment result, the median time from onset to clinical recovery for mild cases is approximately two weeks, so we chose 0.07 as parameter δ_1. Due to 3-6 weeks of recovery time for patients with severe or critical disease, the initial δ_2 is 0.03. The period from onset to the development of the severe diseases is one week according to the previous study [23]. 0.15 is chosen for the initial α. The time from symptom onset to outcome ranges from 2-8 weeks among patients who have died. Thus, we consider the middle time of death time like six weeks and the initial ε set to 0.03. For the model with NN, we choose 0.15 of initial γ, 0.15 of initial α, 0.07 of initial δ_1, 0.03 of initial δ_2 and 0.03 of ε. For using NN to fit β, 0.5 is used as the initial bias of NN, 0.0 is the mean value of the initial weight of NN, and 0.01 is the standard deviation (std) of the initial weight of NN. All the experiments have been implemented on Intel XeonE5-2630 v4 @ 2.20GHzz CPU and NVIDIA RTX 2080Ti GPU on ArchLinux. We implement all models in Pytorch. The DDE algorithms implemented in all epidemiological equations are trained for 5000 iterations. The learning rate (LR) is selected as 1e-3, and it will decay to 0.95×lr after 400 iterations.

To compare the fitting effects of different models and prove our DDE algorithm's advantages, we conducted several experiments. First, we implement the DDE algorithm based on two classical epidemiological equations (SIR and SEIR) and four variants designed by ourselves (SIRD, SEIRD, SMCRD, and SEMCRD). The four models of DDE algorithms are called DDE-SIRD, DDE-SEIRD, DDE-SMCRD, and DDE-SEMCRD. We study the difference between our DDE algorithm and traditional parameter estimation methods (Part A). In part B of the experiment, we introduced the comparison results of parameter estimation (minimizing algorithms: Nelder-Mead, Powell, Truncated Newton Conjugate-Gradient (TNC)), neural ODE, and our DDE. Furthermore, We compared the four models with state-of-the-art machine learning (Decision Tree, Extremely randomized trees, Random Forest) and deep learning methods (RNN, LSTM, GRU), and comparison results are shown in part C.

4.2 Part A: Comparisons of traditional parameter estimation methods and DDE

Table 1 Mean square error (MSE) of SIR, SEIR, SIRD, SEIRD, SMCRD, SEMCRD models in different optimization methods, including Neural ODE(Adam) [12], minimizing algorithms(Nelder-Mead [24], Powell [25], BFGS [26], Truncated Newton Conjugate-Gradient (TNC) [27]). The unit of MSE is ten thousand people.

Models	Parameter Estimation Method	Courtries			Regions	
		CO	USA	ZA	CN-WH	Italy-PD
SIR [13]	Neural ODE	19.317	16550.121	183.378	2.407	4.697
	Nelder-Mead	40.253	2979.545	166.426	5.875	11.609
	Powell	30.002	2093.261	821.494	0.851	3.714
	BFGS	28.419	2094.101	196.961	1.125	8.182
	TNC	27.632	2703.972	54.257	1.099	5.102
	DDE	**0.210**	**52.799**	**1.143**	**0.140**	**0.068**
SEIR [3]	Neural ODE	23.231	10523.219	38.832	0.967	6.812
	Nelder-Mead	22.111	2172.293	195.785	5.677	8.903
	Powell	51.602	2686.344	190.284	1.950	4.922
	BFGS	33.297	2767.012	78.357	8.627	24.941
	TNC	30.671	2882.301	36.361	6.777	3.573
	DDE	**0.192**	**56.901**	**0.967**	**0.250**	**0.034**
SIRD	Neural ODE	29.148	10190.655	137.386	2.910	4.063
	Nelder-Mead	59.437	2722.980	177.995	5.303	8.003
	Powell	23.496	2105.093	84.720	0.975	3.714
	BFGS	57.518	2693.094	176.266	1.067	8.582
	TNC	29.590	2218.704	59.395	1.128	3.179
	DDE	**0.173**	**53.628**	**1.047**	**0.096**	**0.034**
SEIRD	Neural ODE	9.664	10858.052	39.060	0.366	1.271
	Nelder-Mead	57.171	2727.172	184.118	5.298	8.083
	Powell	32.180	2738.686	112.195	1.546	4.001
	BFGS	63.887	2657.202	70.900	8.750	54.941
	TNC	29.307	2611.273	32.233	6.500	3.598
	DDE	**0.182**	**52.377**	**0.370**	**0.138**	**0.028**
SMCRD	Neural ODE	30.634	13553.997	137.763	2.899	4.043
	Nelder-Mead	28.687	7476.677	85.259	2.884	0.705
	Powell	14.957	4737.634	25.595	1.532	2.936
	BFGS	28.687	7476.649	85.260	2.786	7.909
	TNC	24.860	7901.017	84.018	3.678	0.722
	DDE	**0.526**	**44.209**	**2.074**	**0.027**	**0.028**
SEMCRD	Neural ODE	5.475	5521.649	69.415	0.727	0.720
	Nelder-Mead	16.453	5899.235	85.259	2.464	0.405
	Powell	16.181	4182.335	25.209	3.436	2.989
	BFGS	30.886	7476.546	99.747	17.681	1.311
	TNC	27.640	2094.233	115.132	3.072	3.403
	DDE	**0.459**	**41.598**	**0.386**	**0.033**	**0.014**

Using parameter estimation methods to assess essential parameters, including infection, mortality, and the recovery rate is crucial for SEIR and the improved SEIR model to determine these parameters. Therefore, we compared the effects of using traditional parameter estimation methods and using the DDE method we proposed. As shown in Table 1, we obtained the comparison performance of different models in the

national data as follows. In the USA, the MSE obtained by the best method among the traditional parameter estimation methods is 2093.261, and the best MSE of the DDE method is 41.598. The MSE obtained by the best method among the parameter estimation methods is 5.475, and the best MSE obtained by the DDE model is 0.173. On ZA, the MSE obtained by the best method among the parameter estimation methods is 25.209, and the DDE method's MSE is 0.370.

Table 2 Pearson coefficients (Pearson) of SIR, SEIR, SIRD, SEIRD, SMCRD, SEMCRD models in different optimization methods, including Neural ODE(Adam) [12], minimizing algorithms(Nelder-Mead [24], Powell [25], BFGS [26], Truncated Newton Conjugate-Gradient (TNC) [27]). N represents the prediction with no correlation to real data.

Models	Parameter Estimation Method	Courtries			Regions	
		CO	USA	ZA	CN-WH	Italy-PD
SIR [13]	Neural ODE	91.95%	72.04%	82.92%	34.71%	49.77%
	Nelder-Mead	84.53%	86.98%	86.98%	29.60%	46.82%
	Powell	91.28%	88.19%	89.45%	48.41%	55.91%
	BFGS	88.23%	80.29%	85.82%	56.80%	47.27%
	TNC	80.80%	86.92%	91.69%	47.28%	39.98%
	DDE	**99.17%**	**97.14%**	**97.93%**	**98.33%**	**95.18%**
SEIR [3]	Neural ODE	90.64%	78.31%	90.64%	90.77%	59.33%
	Nelder-Mead	90.18%	82.98%	88.69%	22.02%	32.67%
	Powell	84.88%	82.63%	87.94%	45.99%	65.87%
	BFGS	63.17%	84.23%	26.39%	55.88%	47.55%
	TNC	57.44%	80.28%	91.25%	69.52%	35.62%
	DDE	**99.77%**	**98.75%**	**99.01%**	**98.03%**	**94.47%**
SIRD	Neural ODE	90.82%	70.90%	85.09%	31.79%	42.98%
	Nelder-Mead	89.95%	82.56%	86.01%	29.72%	40.54%
	Powell	93.31%	84.57%	89.22%	51.43%	45.74%
	BFGS	90.02%	82.00%	85.89%	23.86%	40.10%
	TNC	92.58%	85.08%	90.61%	44.17%	46.93%
	DDE	**99.47%**	**97.23%**	**98.88%**	**97.33%**	**94.22%**
SEIRD	Neural ODE	95.97%	76.20%	91.66%	93.27%	54.99%
	Nelder-Mead	90.14%	82.56%	85.80%	29.72%	40.45%
	Powell	92.33%	82.70%	87.98%	40.01%	45.21%
	BFGS	53.76%	82.03%	29.45%	59.88%	47.91%
	TNC [27]	92.57%	82.26%	94.00%	68.70%	45.98%
	DDE	**99.47%**	**97.35%**	**99.61%**	**97.32%**	**95.47%**
SMCRD	Neural ODE	90.77%	70.89%	85.06%	31.81%	42.94%
	Nelder-Mead	1.73%	1.86%	1.49%	8.90%	2.34%
	Powell	96.78%	76.35%	93.98%	39.98%	47.46%
	BFGS	-2.78%	6.16%	-5.87%	-6.09%	40.57%
	TNC	-1.62%	17.34%	14.24%	8.36%	5.98%
	DDE	**98.95%**	**97.15%**	**98.71%**	**97.42%**	**95.38%**
SEMCRD	Neural ODE	92.13%	N	86.98%	58.02%	52.35%
	Nelder-Mead	26.34%	N	-2.32%	11.07%	61.82%
	Powell	93.47%	-79.99%	93.50%	32.17%	46.21%
	BFGS	92.03%	N	88.81%	23.07%	48.45%
	TNC	92.80%	N	87.80%	33.07%	46.15%
	DDE	**99.17%**	**87.17%**	**99.48%**	**96.94%**	**97.59%**

Furthermore, on the regional data, the best method among the parameter estimation methods obtains an MSE of 0.405 on PD, and the best MSE of DDE models is 0.014. Also, the best method among the parameter estimation methods gets an MSE of 0.366 on CN-WH, and the best MSE of the DDE method is 0.027 on CN-WH. From this, we can find that the best result of the parameter estimation models in the USA is about 50 times that of the DDE model and the best parameter estimation method in the CO is about 32 times that of the DDE method, about 68 times that of DDE on ZA, about 29 times than DDE on PD, and about 14 times than DDE on CH-WH. Thus, our DDE method has outstanding fitting performance compared to traditional parameter estimation methods.

Also, we used the Pearson coefficient to evaluate the performance of all methods (Table 2). The performance of the DDE method in the Pearson coefficient is also far better than all traditional parameter estimation methods. The best correlation in the USA's DDE method is 98.75%, and that of the best method of traditional parameter estimation methods is 88.19%. In CO, the best DDE method obtains 99.77% in Pearson coefficient, and in the best parameter estimation method, the Pearson coefficient is 95.97%. The Pearson coefficient obtained in ZA is 99.61% of the best DDE model, and in the traditional parameter estimation method, the Pearson value is 93.98%. Moreover, in cities, the best Pearson coefficient of the best DDE in CH-WH is 98.33%, and the Pearson value of the best parameter estimation method is 93.27%. In Italy-PD, the best DDE gains 97.59% in Pearson value, and the best traditional parameter estimation method obtains 61.82%. Therefore, we can conclude that the DDE model outperforms all traditional parameter estimation methods in correlation and precision.

4.3 Comparisons of DDE with Learning-based Methods

To further assess our DDE method's fitting ability, we also compare the learning-based models with four of our DDE models (DDE-SIRD, DDE-SEIRD, DDE-SMCRD, DDE-SEMCRD). Fig. 6 shows the overall forecast results on national data (CO, ZA, and the United States). From the overall trend, as a representative of the deep learning method, LSTM can often achieve better results when the 20-day test data changes slowly, but the stability is not good. The DDE models can better capture the changing laws of the data, and some of the more complex changes are closer to the development of real data: on the infection curve in South Africa. Further, in Fig 7, the regional data (Wuhan, China, Piedmont, Italy) results of the 20-day test data show that our DDE models obtain the best prediction performance in most situations. Deep learning methods, such as GRU or LSTM obtained close-fitting performance compared with DDE. However, sometimes it even got the opposite trend of the real-world data, as on Piedmont's death curve.

Moreover, the 20-day test results on national data (CO, ZA, and the USA) are shown in Fig. 6. The results of USA data show that our DDE-SEIR models can better fit the data trends of I and R, while on D, the growth of D will be slightly overestimated. The method corresponding to deep-learning has basically no increase in I, R, and D, and the effect of estimating the trend is not obvious. Furthermore, on CO, the two models of DDE-SIRD and DDE-SEIRD in our model can better fit the basic trend of I, but some slight drops cannot be predicted. The fitting of R is more

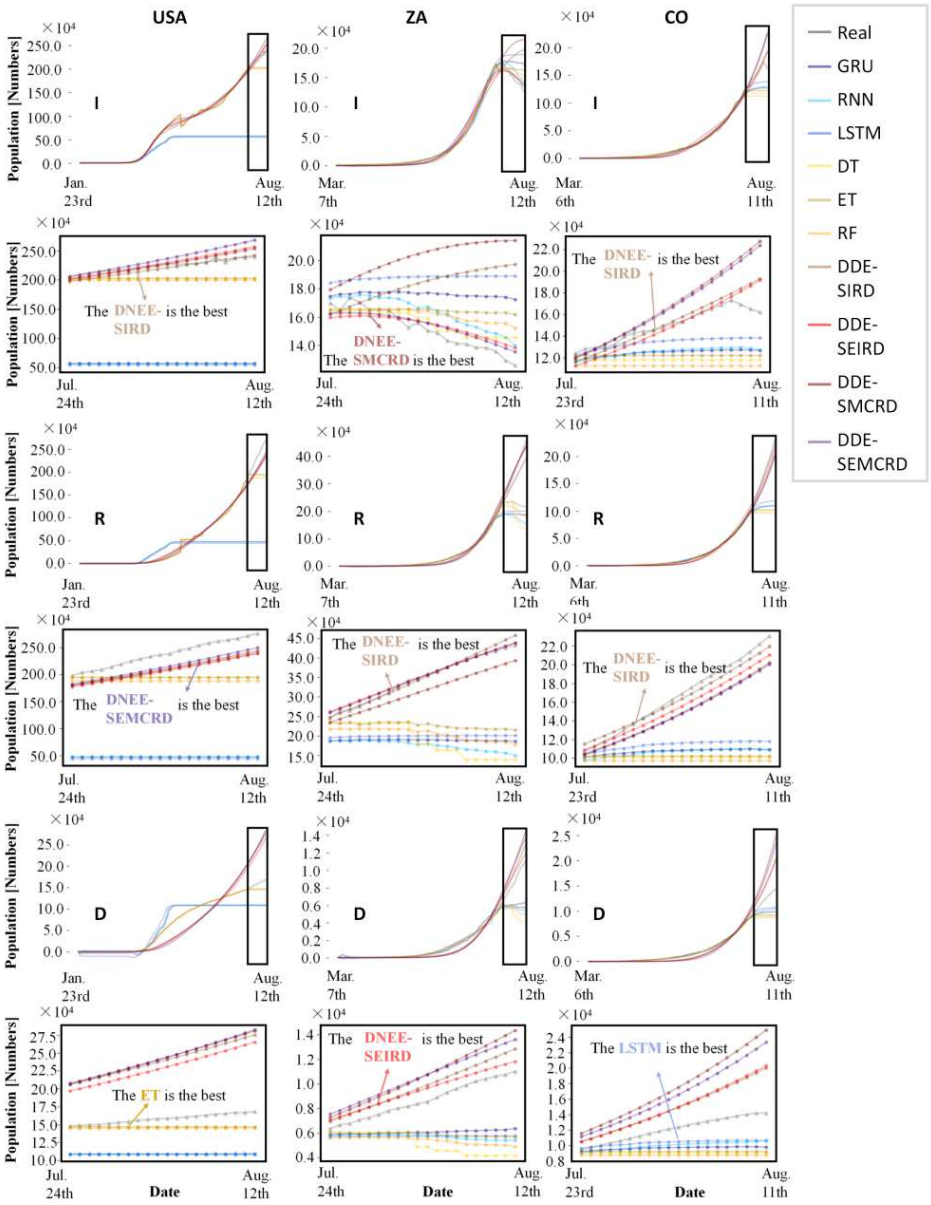

Fig. 6 The figure shows all models' performance (machine-learning: RF, ET, DT, deep-learning: LSTM, RNN, GRU, our models: DDE-SIRD, DDE-SEIRD, DDE-SMCRD, DDE-SEMCRD) in predicting the progression of infectious diseases at the country level. There are six pictures in each column in the figure, and every two pictures are a group. In a group of images, the above image represents overall trend fitting, and the bottom illustrates the prediction of the 20-day testing data. The group of pictures on the top represents the prediction of infection I. The group of pictures in the center represents the prediction of recovery R. The group of two pictures on the bottom represents death D. Furthermore, in the picture shown in the 20-day test situation, we have given the best model's name corresponding to the predicted performance. From the overall predictions, our DDE model has better prediction performance in the prediction of three countries.

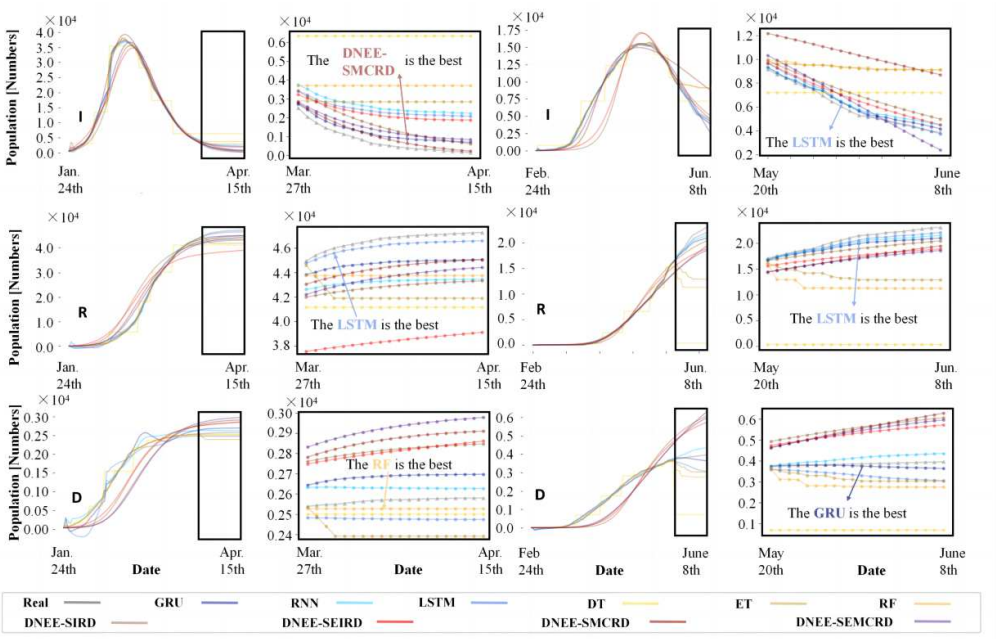

Fig. 7 The figure shows all models' performance (machine-learning: RF, ET, DT, deep-learning: LSTM, RNN, GRU, our models: DDE-SIRD, DDE-SEIRD, DDE-SMCRD, DDE-SEMCRD) in predicting the progression of infectious diseases at the region-level. There are six pictures in each column in the figure, and every two pictures are a group. In a group of images, the above image represents overall trend fitting, and the bottom illustrates the prediction of the 20-day testing data. The group of pictures on the top represents the prediction of infection I. The group of pictures in the center represents the prediction of recovery R. The group of two pictures on the bottom represents death D. Furthermore, in the picture shown in the 20-day test situation, we have given the best model's name corresponding to the predicted performance. From the overall predictions, our DDE model has better prediction performance in the prediction of two regions.

accurate, but there is a certain overestimation of D. However, in CO, the deep-learning method still has the problem that the trend is not obvious. In the fitting of ZA, since the trend of I has decreased, the difficulty of correct estimation has increased. The DDE-SIRD model and the DDE-SMCRD model have incorrect estimates of trends. The DDE-SEIRD and SECRD with NN models can accurately estimate the downward trend of I. Furthermore, on R and D, all of our four models are good estimates of the trend. The deep-learning method can also better predict the downward trend on I, and the RNN estimate is the closest to the true value. However, in the fitting of R and D, the deep-learning method not only fails to predict the direction of the data but obtains the opposite result from the true value.

Furthermore, Fig. 7 also shows the 20-day test results on the regional data (PD and CN-WH). No matter whether in PD or CN-WH, all methods can basically estimate the trend of predicted data. Furthermore, some deep-learning methods have achieved better results in CN-WH and PD. No matter whether in PD or CN-WH, all methods can basically estimate the trend of predicted data. Furthermore, some deep-learning

Table 3 Model Correlation Coefficient (Pearson) of RNN, LSTM, GRU, Decision Tree (DT), Extremely randomized trees (ET), Random Forest(RF), and DDE models based on SIRD, SEIRD, SMCRD, and SEMCRD)). N represents the prediction results and the real data are without correlation.

Areas	Data	Models									
		GRU	LSTM	RNN	DT	ET	RF	DDE-SIRD	DDE-SEIRD	DDE-SMCRD	DDE-SEMCRD
CO	I	92.68%	92.96%	94.52%	N	N	N	**95.77%**	95.13%	94.97%	94.98%
	R	88.72%	90.97%	90.86%	N	N	N	**99.86%**	99.85%	99.78%	99.80%
	D	93.85%	95.28%	98.51%	N	N	N	98.83%	98.77%	98.52%	**98.53%**
USA	I	N	-64.96%	N	N	N	N	**97.81%**	97.55%	97.54%	97.49%
	R	N	62.31%	N	N	-87.91%	N	**99.76%**	99.73%	99.73%	99.71%
	D	N	64.81%	N	N	N	N	**99.71%**	99.69%	99.69%	99.67%
ZA	I	72.39%	-77.85%	95.69%	92.24%	79.95%	93.10%	-95.06%	94.83%	-91.85%	**96.39%**
	R	-43.67%	88.58%	-92.86	-92.80%	-82.24%	-90.96%	99.35%	**99.70%**	99.61%	99.69%
	D	98.63%	82.90%	-92.61%	-92.92%	-90.31%	-95.18%	**99.80%**	99.66%	99.79%	99.73%
CN-WH	I	**0.9985**	99.79%	99.84%	N	85.52%	N	98.23%	99.78%	98.87%	99.46%
	R	99.78%	**99.84%**	99.80%	N	-94.92%	N	98.23%	94.67%	98.73%	98.08%
	D	97.35%	-98.93%	-93.32%	N	-73.68%	N	**98.89%**	96.17%	99.17%	98.75%
Italy-PD	I	99.82%	99.76%	**99.82%**	N	98.45%	96.66%	99.23%	99.34%	98.15%	98.15%
	R	99.73%	**99.84%**	99.78%	N	-94.92%	-70.83%	99.33%	99.41%	98.74%	99.44%
	D	-85.10%	-98.71%	99.20%	N	-92.77%	-75.72%	**99.83%**	99.81 %	99.78%	99.77%

methods have achieved better results in CN-WH and PD. It can be found that on regional data, all models have a better fitting effect; on national data, due to the complexity, the deep-learning method is not as effective as our method.

The correlation metrics including Pearson correlation coefficient (Pearson) for our DDE-SIRD, DDE-SEIRD, DDE-SMCRD, DDE-SEMCRD, and other learning-based models (deep-learning method: RNN, LSTM, GRU, and traditional machine learning method: Random Forest, Extra Tree, Decision Tree) of 20-day test data are shown in Table 3. According to the Pearson metric for real-world data in Table 3, deep learning methods such as LSTM and RNN models have achieved good results in regional data fitting. In the data fitting of CN-WH, the best Pearson on I and R are obtained by GRU and LSTM respectively. The situation of PD data is similar. The best Pearson for I and R are obtained on RNN and LSTM respectively. However, there are still unstable problems in the fitting of these deep learning algorithms, and the negative correlation between the predicted data and the real data often occurs.

In addition, machine learning methods including random forest, extreme tree, and decision tree are not very effective in fitting regional data. There is a situation where the prediction has no correlation or the correlation is negative with real data. However, our DDE models (DDE-SIRD, DDE-SEIRD, DDE-SMCRD, and DDE-SEMCRD) have achieved relatively stable and excellent performance in the overall prediction of regional I, R, and D data. Based on the Wuhan data, the four deformed DDE-SEIR models have achieved I, R, and D average correlation coefficients of 98.45%, 97.87%, 98.92%, and 98.76% respectively. Furthermore, on the PD data, they also obtained the average Pearson correlation coefficients of 99.46%, 99.52%, 98.56%, and 99.12% respectively.

4.4 The estimated parameters obtained by DDE method

In the USA, the average initial infection rate $\beta*$ is 0.469, and we obtain the average initial infection rate $\beta*$ of 0.222 in the CO. The average initial infection rate $\beta*$ is

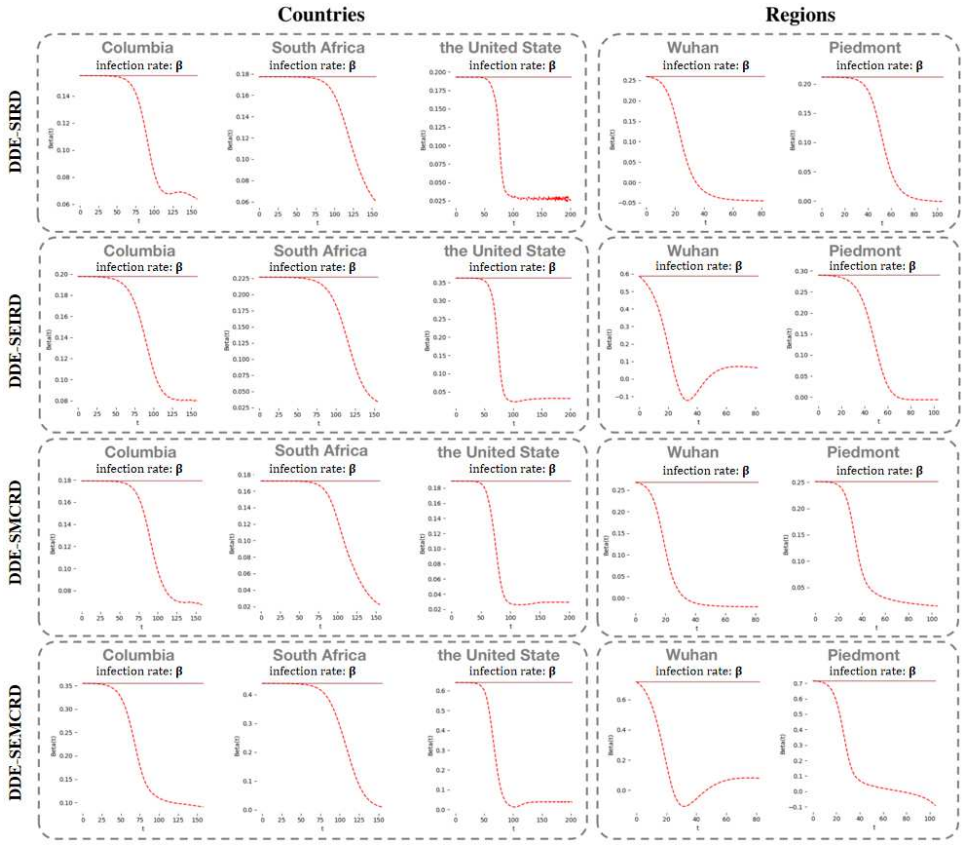

Fig. 8 Schematic diagram of the infection rate beta generated by each model and different countries and regions. Models include DDE-SIRD, DDE-SEIRD, DDE-SMCRD and DDE-SEMCRD. The dark red line in the figure represents the initial infection rate, and the red dotted curve line represents the change in the infection rate under the influence of the diversity of each country. The curves are simulated by a neural network.

0.254 in the ZA. In regions, the average $\beta*$ in Wuhan is 0.4578, and Piedmont gains the average initial infection rate of 0.367. The infection rates our DDE method obtained are suitable for real-world situations. In Fig. 8, the DDE also shows that it can simulate various infection rate changes using neural networks to fit the effect function. The DDE models reflect the basic trend of the gradual decline in the infection rate and more accurately estimate the initial infection rate.

5 Conclusion

An essential tool for modeling and estimating epidemiological equations (EE) is needed for fitting and analyzing epidemic disease data. We propose a DDE method that incorporates NN into EE modeling, achieving high performance in multi-level geographic regions. Experimental results show that our DDE model can obtain high and stable ability in different regions, surpassing traditional parameter estimation methods.

While learning-based models like LSTM and random forest have comparable data fitting performance, they lack interpretability for epidemiologists. To address this, our DDE method combines EE with neural networks and utilizes the Neural ODE method to improve data fitting. This way, the output of the NN carries a specific meaning. For instance, in the SEMCRD model using DDE, the network's output represents the effective function of the infection rate. Our model has demonstrated better accuracy and correlation with real-world data. Additionally, our method offers novel ideas and tools for future analysis of infectious diseases.

Author Contribution

Bin Sheng, Huating Li, Ping Zhang, and David Dagan Feng conceived and supervised the project. Ruhan Liu and Jiajia Li designed the deep-learning algorithm and the computational framework. Ruhan Liu, Jiajia Li, and Yang Wen designed the study and contributed to the initial drafting of the manuscript. Ruhan Liu collected and organized data and performed statistical analysis. Ping Zhang, Huating Li, and Bin Sheng provided critical comments and reviewed the manuscript. All authors discussed the results and approved the final version before submission.

Funding

We would like to thank all the investigators and study subjects for this study. This work is supported by the Project of Intelligent Management Software for Multimodal Medical Big Data for New Generation Information Technology, Ministry of Industry and Information Technology of People?s Republic of China (TC210804V).

Data Availability

The data being used in the current study is provided at https://github.com/Liuruhan/DENN.

Data Availability

The code being used in the current study for developing the algorithm is provided at https://github.com/Liuruhan/DENN.

References

[1] Cyranoski, D.: What China's coronavirus response can teach the rest of the world. Nature **579**(7800), 479–480 (2020)

[2] Kwok, K.O., Chan, H.H.H., Huang, Y., Hui, D.S.C., Tambyah, P.A., Wei, W.I., Chau, P.Y.K., Wong, S.Y.S., Tang, J.W.T.: Inferring super-spreading from transmission clusters of COVID-19 in Hong Kong, Japan and Singapore. Journal of Hospital Infection **105**(4), 682–685 (2020)

[3] Saito, M.M., Imoto, S., Yamaguchi, R., Sato, H., Nakada, H., Kami, M., Miyano, S., Higuchi, T.: Extension and verification of the SEIR model on the

2009 influenza a (H1N1) pandemic in Japan. Bellman Prize in Mathematical Biosciences **246**(1), 47–54 (2013)

[4] Peng, L., Yang, W., Zhang, D., Zhuge, C., Hong, L.: Epidemic analysis of COVID-19 in China by dynamical modeling. ArXiv preprint **arXiv:2002.06563** (2020)

[5] Wei, Y., Lu, Z., Du, Z., Zhang, Z., Zhao, Y., Shen, S., Wang, B., Hao, Y., Chen, F.: Fitting and forecasting the trend of COVID-19 by SEIR + CAQ dynamic model. Chinese Journal of Epidemiology **41**(4), 470–475 (2020)

[6] Choi, S.C., Ki, M.: Estimating the reproductive number and the outbreak size of COVID-19 in Korea. Epidemiology and Health **42**, 1–10 (2020)

[7] Rustam, F., Reshi, A.A., Mehmood, A., Ullah, S., On, B.-W., Aslam, W., Choi, G.S.: Covid-19 future forecasting using supervised machine learning models. IEEE Access **8**(8), 101489–101499 (2020)

[8] Yang, Z., Zeng, Z., Wang, K., Wong, S.S., Liang, W., Zanin, M., Liu, P., Cao, X., Gao, Z., Mai, Z., Liang, J., Liu, X., Li, S., Li, Y., Ye, F., Guan, W., Yang, Y., Li, F., Luo, S., Xie, Y., Liu, B., Wang, Z., Zhang, S., Wang, Y., Zhong, N., He, J.: Modified SEIR and AI prediction of the epidemics trend of COVID-19 in China under public health interventions. Journal of Thoracic Disease **12**(3), 165–174 (2020)

[9] Zhu, Y., Wang, S., Wang, S., Wu, Q., Wang, L., Li, H., Wang, M., Niu, M., Zha, Y., Tian, J.: Mix contrast for COVID-19 mild-to-critical prediction. IEEE Trans. Biomed. Eng. **68**(12), 3725–3736 (2021)

[10] Wojtusiak, J., Bagchi, P., Durbha, S.S.K.R.T.N., Mobahi, H., Nia, R.M., Roess, A.: COVID-19 symptom monitoring and social distancing in a university population. J. Heal. Informatics Res. **5**(1), 114–131 (2021) https://doi.org/10.1007/s41666-020-00089-x

[11] Wesner, J.S., Peursem, D.V., Flores, J.D., Lio, Y., Wesner, C.A.: Forecasting hospitalizations due to COVID-19 in south dakota, USA. J. Heal. Informatics Res. **5**(2), 218–229 (2021) https://doi.org/10.1007/s41666-021-00094-8

[12] Chen, T.Q., Rubanova, Y., Bettencourt, J., Duvenaud, D.: Neural ordinary differential equations. In: Neural Information Processing Systems, pp. 6572–6583 (2018)

[13] Fang, H., Chen, J., Hu, J.: Modelling the SARS epidemic by a lattice-based Monte-Carlo simulation. In: IEEE Engineering in Medicine and Biology 27th Annual Conference, vol. 7, pp. 7470–7473 (2005)

[14] Smirnova, A., deCamp, L., Chowell, G.: Forecasting epidemics through nonparametric estimation of time-dependent transmission rates using the SEIR model.

Bulletin of Mathematical Biology **81**(11), 4343–4365 (2019)

[15] Kraemer, M.U.G., Yang, C.H., Gutierrez, B., Wu, C.H., Klein, B., Pigott, D.M., Plessis, Faria, N.R., Li, R., Hanage, W.P., Brownstein, J.S., Layan, M., Vespignani, A., Tian, H., Dye, C., Pybus, O.G., Scarpino, S.V.: The effect of human mobility and control measures on the COVID-19 epidemic in China. Science **368**(6490), 493–497 (2020)

[16] Chinazzi, M., Davis, J.T., Ajelli, M., Gioannini, C., Litvinova, M., Merler, S., Piontti, A.P., Mu, K., Rossi, L., Sun, K., Viboud, C., Xiong, X., Yu, H., Halloran, M.E., Longini, I.M., Vespignani, A.: The effect of travel restrictions on the spread of the 2019 novel coronavirus (COVID-19) outbreak. Science **368**(6489), 395–400 (2020)

[17] Kucharski, A.J., Russell, T.W., Diamond, C., Liu, Y., Edmunds, J., Funk, S., Eggo, R.M., Sun, F., Jit, M., Munday, J.D., Davies, N., Gimma, A., Zandvoort, K., Gibbs, H., Hellewell, J., Jarvis, C.I., Clifford, S., Quilty, B.J., Bosse, N.I., Abbott, S., Klepac, P., Flasche, S.: Early dynamics of transmission and control of COVID-19: A mathematical modelling study. Lancet Infectious Diseases **20**(5), 553–558 (2020)

[18] Raj, D., George, B.: Quantifying the effect of quarantine control in COVID-19 infectious spread using machine learning. MedRxiv preprint **medRxiv:2020.04.03.20052084** (2020)

[19] Read, J.M., Bridgen, J.R., Cummings, D.A., Ho, A., Jewell, C.P.: Novel coronavirus 2019-nCoV: early estimation of epidemiological parameters and epidemic predictions. MedRxiv preprint **medRxiv:2020.01.23.20018549** (2020)

[20] Tang, B., Wang, X., Li, Q., Bragazzi, N.L., Tang, S., Xiao, Y., Wu, J.: Estimation of the transmission risk of the 2019-nCoV and its implication for public health interventions. Journal of Clinical Medicine **9**(462), 1–13 (2020)

[21] Wu, J.T., Leung, K., Leung, G.M.: Nowcasting and forecasting the potential domestic and international spread of the 2019-nCoV outbreak originating in Wuhan, China: A modelling study. The Lancet **395**(10225), 689–697 (2020)

[22] Xu, B., Kraemer, M.U.G., Gutierrez, B., Mekaru, S., Sewalk, K., Loskill, A., Wang, L., Cohn, E., Hill, S., Zarebski, A., Li, S., Wu, C.H., Hulland, E., Morgan, J., Scarpino, S., Brownstein, J., Pybus, O., Pigott, D., Kraemer, M.: Open access epidemiological data from the COVID-19 outbreak. Lancet Infectious Diseases **20**(5), 534–534 (2020)

[23] Backer, J.A., Klinkenberg, D., Wallinga, J.: Incubation period of 2019 novel coronavirus (2019-nCoV) infections among travellers from Wuhan, China, 20-28 January 2020. Eurosurveillance **25**(5, pii. 2000062), 1–6 (2020)

[24] Nelder, J.A., Mead, R.: A simplex method for function minimization. Comput. J. **7**(4), 308–313 (1965)

[25] Powell, M.J.D.: An iterative method for finding stationary values of a function of several variables. Comput. J. **5**(2), 147–151 (1962)

[26] Moghrabi, I.A.R.: Extra multistep BFGS updates in quasi-newton methods. Int. J. Math. Math. Sci. **2006**, 12583–1125838 (2006)

[27] Hsia, C., Chiang, W., Lin, C.: Preconditioned conjugate gradient methods in truncated newton frameworks for large-scale linear classification. In: Proc. ACML, vol. 95, pp. 312–326 (2018)