CCF Transactions on Pervasive Computing and Interaction (2024) 6:281–295
https://doi.org/10.1007/s42486-024-00161-0

REGULAR PAPER

# Graph neural ordinary differential equations for epidemic forecasting

**Xiong Yanqin[1] · Wang Huandong[2] · Liu Guanghua[1] · Li Yong[2] · Jiang Tao[1]**

Received: 21 January 2024 / Accepted: 28 May 2024 / Published online: 12 July 2024


**Abstract**
Developing a practical model for predicting the dynamics of epidemic spread is critical, the results of which can be used to evaluate the effectiveness of prevention and control measures as well as the allocation of medical and health resources. However, accurate prediction is challenging because epidemic spread is closely associated with population mobility, which is nonlinear and complex, making reliable prediction difficult. Furthermore, the epidemic observed data is sparse and irregularly sampled, rendering the traditional time series models ineffective. Under these circumstances, this paper designs a graph neural ordinary differential equations approach, which combines Ordinary Differential Equation Networks (ODENet) and Graph Neural Networks (GNNs). This approach adopts a new attention mechanism taking into account the interaction between regional epidemic information and interaction between regions, achieving the precise continuous-time epidemic prediction based on non-Euclidean data. In addition, we use Transformer to deduce the value of the initial hidden state via future observable data in an innovative way, reconstructing the hidden state successfully. We conduct a lot of experiments based on the contact matrix and simulated epidemic data in San Francisco from March 2020 to May 2020, results show that our method can not only forecast the dynamics of epidemic spread, but also mine hidden patterns in observable data and extract hidden states.

**Keywords** Spatial temporal forecasting · Graph neural network · Neural ODE · Epidemic

## 1 Introduction

In recent years, a number of large-scale respiratory diseases, such as COVID-19, H1N1, and SARS, have erupted on a global scale, exerting a significant influence on both people's lives and social stability. Taking the current COVID-19 outbreak as an example, the World Health Organization (WHO) received reports of over 772 million confirmed cases of COVID-19 as of 17 December 2023, including nearly seven million deaths [1]. This epidemic has put existing medical resources and systems to the test. Accurately predicting the dynamics of epidemic spreading can assist health departments and medical institutions in making effective responses, such as resource allocation, prevention and control measures, and medical preparations to protect people's lives and health.

However, developing a successful data-driven approach to model regional-level epidemiological dynamics is very challenging: on the one hand, the spread of large-scale human infectious diseases is closely linked to the mobility behavior of populations, which is highly non-linear and complex. In terms of epidemic transmission at the regional level, due to different occupations, ages, genders, living habits, and other factors, the travel destinations and specific travel routes of residents in different regions will be quite different(Kleczkowski and Grenfell 1999). On the other hand, epidemic data is typically irregularly sampled,

✉ Wang Huandong
wanghuandong@tsinghua.edu.cn

Xiong Yanqin
xiongyanqin@hust.edu.cn

Liu Guanghua
guanghualiu@hust.edu.cn

Li Yong
liyong07@tsinghua.edu.cn

Jiang Tao
taojiang@hust.edu.cn

[1] Research Center of 6G Mobile Communications, School of Cyber Science and Engineering, and Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology, Wuhan 430074, Hubei, China

[2] Beijing National Research Center for Information Science and Technology (BNRist), Department of Electronic Engineering, Tsinghua University, Beijing 100084, China

the time period for nucleic acid sampling usually fluctuates flexibly with the intensity of the epidemic. As a result, the sampling time of epidemic data may be one day, one week, or one month. However, the time series prediction models that are now in use, such as the Long Short Term Memory (LSTM), Gated Recurrent Unit (GRU), and Temporal Convolutional Network (TCN) (Graves 2012; Cho et al. 2014; Bai et al. 2018), are based on regularly sampled data.

The capacity of graph neural networks (GNNs) to process non-Euclidean data well has made them a popular choice for tasks involving epidemic prediction in recent years, there has been a lot of interest in using GNN to simulate the geographic characteristics of epidemic transmission networks, representative works include causal-GNN (Wang et al. 2022), cola-GNN (Deng et al. 2020) and STAN (Gao et al. 2021), etc. Other studies aim to jointly capture the temporal and spatial properties of the epidemic transmission process by combining GNN with conventional time series processing models (such as Recurrent Neural Network (RNN)) (Sesti et al. 2021). Nevertheless, there are some issues that have consistently gone unattended. On the one hand, for the GNN model, when the number of layers stacked is too much, it will suffer from over smoothing, which will lead to poor performance of the model (Li et al. 2018; Zhou et al. 2020). On the other hand, most of the existing time series processing tools can only achieve the prediction task under the condition of equal interval sampling data and discrete time. The existing research shows that the adoption of Neural Ordinary Differential Equation Networks (ODENet) in GNN models can avoid or solve the above problems (Poli et al. 2019; Huang et al. 2021). Last but not least, the interaction between regional epidemic information and interregional population mobility is an important aspect that has been largely overlooked in current research.

In this paper, we propose a novel graph neural ordinary differential equations approach which combines GNN and ODENet to address the aforementioned issues. This method tries to integrate the epidemic information of the region itself and the interaction information between regions through a well-designed fusion matrix, and model differential equation systems employing GNNs on the graph to accurately represent the instantaneous rate of change of nodes' states. Transformer (Vaswani et al. 2017) is used to realize the reconstruction of the hidden state during the evolution of the epidemic. Our contributions can be summarized as follows:

- We propose to combine GNN and ODENet to make fine-grained epidemic forecasting. Thanks to the characteristics of ordinary differential equations (ODEs), we can predict the dynamics of epidemic spread on continuous time domain on the graph structure.
- We investigate a fusion attention matrix that takes into account the correlation between epidemic across regions and the influence of population movement on the spread of epidemic. By incorporating these factors, our approach achieves a more comprehensive and accurate analysis and prediction of epidemic.
- We incorporate the Transformer model in a novel way into our approach, taking advantage of its ability to globally model the input sequence via the self-attention mechanism. As a result, the Transformer model captures the sequence's underlying patterns and regularities (Voita et al. 2019), allowing us to infer the initial hidden state from the observed time series data, realising the reconstruction of the hidden state throughout the entire observation period.
- Extensive comparison experiments are performed on two COVID-19 datasets with different modeling settings. The results demonstrate that our proposed model[1] outperforms the baseline models and achieves up to 7.06% performance improvement in long-term prediction tasks. To confirm the efficacy of each component of our model, we also created a set of ablation studies. The verification experiment confirmed that our model can predict the number of people in the latent period. The experiments based on the multi-scale sampling training strategy also demonstrated the capability of our model to accurately predict epidemics using irregularly sampled data.

## 2 Related work

### 2.1 Spatial-temporal prediction

Spatial-temporal prediction refers to the prediction of the unknown system state in time and space. In the past few decades, spatial-temporal prediction has been widely used in traffic forecasting (Yu et al. 2017; Fang et al, 2021; Yang et al. 2021), weather forecasting (Shi et al. 2015), earthquake forecasting (Bhatia et al. 2018), etc. Liu et al. (2017) propose to use the Conv-LSTM and Bi-LSTM modules to concentrate the spatial-temporal feature and period feature to predict the traffic flow. Li et al. (2017) introduce a diffusion convolutional recurrent neural network (DCRNN), which models spatial dynamics and temporal dynamics using two-way random walks and recurrent neural networks, respectively.

With the rise of graph neural networks, many researchers have attempted to developed spatio-temporal graph neural networks (STGNNs) to solve spatiotemporal prediction problems. Spatio-Temporal Graph Convolutional Networks(STGCN) (Yu et al. 2017) uses a spatio-temporal

[1] A repository with our source code can be found at: https://github.com/xiongzhangxyq/GNNODENet.

blocks to integrate graph convolution and gated temporal convolution for traffic prediction. Spatial-Temporal Graph Ordinary Differential Equation Networks(STGODE) (Fang et al, 2021) is a new continuous representation of Graph Convolutional Network (GCN) to increase the depth of GCN, expand the spatial receptive field, and capture deeper spatial-temporal dependence. STAN (Gao et al. 2021) captures geographic and temporal trends using attention-based graph convolutional networks, and designs a loss term based on physical laws, improving the long-term prediction performance. In addition, some studies have focused on complex local and global spatiotemporal dependencies (Zhang et al. 2021; Liu et al. 2023), recognizing the value of multivariate time series (Jin et al. 2022) and proposing more sophisticated predictive models.

In view of the strong correlation between the epidemic's spread and the contact of the population in the city, epicemic prediction can be regarded as a spatial-temporal prediction problem here. However, different from traffic flow forecast and weather forecast, whose datas can be collected regularly, epidemic data collection usually shows more diversified granularity (e.g., days, weeks) with a delay and even missing, and epidemic outbreaks also exhibit abruptness. The introduction of an efficient epidemic prediction model for long-term predictions of epidemics on the continuous time domain based on limited data is extremely important. Here we adopt ODENet to solve the question, and achieve epidemic prediction on continuous time domain.

## 2.2 Epidemic modeling

According to the characteristics of different infectious diseases, mathematicians and epidemiologists have developed different models to predict epidemic. Some studies model epidemics based on differential equations, the two classical models of this category are the susceptible-infected-susceptible (SIS) model and the susceptible-infected-recovered (SIR) model. Some follow-up studies have expanded the SIS model and SIR model to further consider complex evolution process (Maier and Brockmann 2020; Zhang et al. 2020). In addition to establishing traditional dynamic equation models, many researchers have developed probability-based models (Balcan et al. 2009; Chang et al. 2021), due to their great abilities for modeling nonlinear properties. These approaches, while their excellent effectiveness, have some drawbacks, including numerous idealistic assumptions that cannot be guaranteed to be accurate.

Given the severity of the pandemic and the need for accurate forecasting of the epidemic's spread, machine learning approaches have recently started to emerge as a promising methodology to combat epidemic (Xie et al. 2022; Wang et al. 2020; Zhu et al. 2019). Wu et al. (2018) separately employ convolutional neural networks (CNNs) for spatial correlations and Recurrent Neural Networks (RNNs) for temporal correlations to make epidemic predictions.

However, it is obvious that non-Euclidean network structure data does not adapt itself to traditional convolution for regular grids. GNNs come into being, and have been widely used in epidemic forecasting. Some studies combine GNNs and time series forecasting models to capture spatial correlation and temporal correlation (Panagopoulos et al. 2021; Sesti et al. 2021; Kargas et al. 2021). Wang et al. (2022) proposed Causal-based Graph Neural Network (CausalGNN) which is a GNN-based model taking causal computations into account when making spatio-temporal epidemic forecasting. Deng et al. (2020) proposed a brand-new graphic messaging framework based on the idea of fusing the graph structure and time series features in the dynamic propagation process. Even though most of the above models can achieve better prediction performance, they are helpless in the face of non-equidistant sampling data, such as nucleic acid sampling data. In our work, we use GNN to aggregate the epidemic information of the region itself and the interaction information between regions, to simulate the spread of the epidemic in the network, and use the characteristics of ordinary differential equations to predict the epidemic development situation at any node at any time to adapt to non-equidistant sampling sequentially.

## 2.3 Continuous time series forecasting

Most existing time series methods (e.g., RNN) assume that the interval between these observations and latent variables is fixed, which often causes the estimated distribution to deviate from the population distribution. These issues are easily avoided by ODENet, which models the time series as a continually evolving trajectory, this allows it to make better use of the data's timestamp information and to generate predictions on the continuous time domain.

In recent years, some studies have combined ordinary differential equation (ODE) with deep learning techniques, and proposed a knowledge-data joint-driven ODENet method. This method was first proposed in the literature (Chen et al. 2018). Its core idea is to use deep learning neural network to fit the derivative in the differential equation, and at the same time combine the classical solution method of ODE to further complete the spatiotemporal situation modeling and forecasting of the epidemic spread on the irregular time interval sequence. At the same time, Xhonneux et al. (2020) combined ODENet with GCN to successfully model the temporal dynamic changes in graph structure, providing an important solution for modeling complex non-Euclidean spatial associations between different regions of the city. In our work, we use ODENet to realize the continuous GNN layers in mathematical sense, rather than using discrete superimposed GNN layers, which enables us to carry out

continuous time prediction tasks and avoid the over smoothing problem in the GNN model when there are too many layers.

## 3 Proposed method

### 3.1 Problem formulation

**Definition 1** (the epidemic transmission network $\mathcal{G}$) In our study, we assume that there are $N$ regions and each region can be regarded as a node, thus we can adopt a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ to represent the epidemic spread network which consists of $N$ nodes.

**Definition 2** (the state $\boldsymbol{X}$ on the graph) In our study, we express the observation of node $i$ at time $t$ as $x(t)^i \in \mathbb{R}^{F_d}$, where $F_d$ is the length of an observation vector. $X(t_i) = (x(t_i)^1, x(t_i)^2, ..., x(t_i)^N) \in \mathbb{R}^{N \times F_d}$ denotes the observations of all nodes at time $t_i$. $\boldsymbol{X} = (X(t_0), X(t_1), \ldots, X(t_T)) \in \mathbb{R}^{(T+1) \times N \times F_d}$ denotes the state of all nodes within the time range of length $T + 1$, and the unit of $T$ is unlimited.

**Definition 3** (the adjacency matrix $A_g$) Assuming there are $N$ nodes, which can be divided into two types of nodes: type $C$ nodes, which refer to census block groups (CBGs) such as residential districts, and type $P$ nodes, which refer to points of interest (POIs) such as shops, bars, gas stations, hospitals, and a series of places that provide people with daily services. It's worth noting that the $'P'$ and $'C'$ labels simply provide a practical distinction between the two types of nodes, and that the nodes within each type are independent of each other. This is not a heterogeneous graph problem.

Assuming that the visit matrix representing the size of population flow between type $C$ nodes and type $P$ nodes is $U$ at time $t$, it can be represented as follows structure:

$$\begin{array}{c|cccccc} & P_1 & P_2 & \cdots & P_j & \cdots & P_n \\ C_1 & u_{11} & u_{12} & \cdots & u_{1j} & \cdots & u_{1n} \\ C_2 & u_{21} & u_{22} & \cdots & u_{2j} & \cdots & u_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ C_i & u_{i1} & u_{i2} & \cdots & u_{ij} & \cdots & u_{in} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ C_m & u_{m1} & u_{m2} & \cdots & u_{mj} & \cdots & u_{mn} \end{array}$$

where $u_{ij}$ represents the number of population flow from node $i$ to node $j$, $m$ and $n$ are the number of type $C$ nodes and type $P$ nodes respectively, and $N = n + m$, the corresponding adjacency matrix $A_g$ representing the interaction between all nodes is as follows:

**Table 1** A list of commonly used notations

| Notation | Description |
|---|---|
| $\mathcal{G}$ | The graph |
| $\mathcal{T}$ | The set of time slots |
| $\mathcal{V}$ | The set of all nodes whose number is $N$ |
| $\mathcal{E}$ | The set of all edges |
| $\mathcal{U}$ | The population flow between regions |
| $\mathcal{I}$ | Identity matrix whose dimension is determined by context |
| $A_g \in R^{N \times N}$ | The adjacency matrix |
| $\hat{A} \in R^{N \times N}$ | The fusion matrix |
| $F_d$ | The number of dynamic features |
| $F_e$ | The features dimensions of input encoded vector |
| $C(t) \in R^{N \times F_e}$ | The hidden state |
| $Z_d(t) \in R^{N \times 2F_e}$ | The dynamic features at time $t$ |
| $Z_s \in R^{N \times F_s}$ | The static features |
| $F_s$ | The number of dynamic features |
| $\mathcal{L}(\Theta)$ | Loss function of the model |

$$\begin{bmatrix} 0 & U \\ U^T & 0 \end{bmatrix} \tag{1}$$

the adjacency matrix $A_g$ will be used to aggregate information in the follow-up work, and its normalized form is $\tilde{A_g}$.

*Epidemic Prediction on Population Flow Graphs* The basic goal of epidemic prediction is to forecast the most likely epidemic measurements in the next $L$ time steps based on the previous $H$ epidemic observations. At each time step $t_i$, the graph $\mathcal{G}$ is associated with a feature matrix $X(t_i) \in \mathbb{R}^{N \times F_d}$, the graph nodes are connected via an adjacency matrix $A_{gt_i} \in \mathbb{R}^{N \times N}$. The important notations are described in Table 1.

### 3.2 Model

The proposed framework depicted in the Fig. 1 consists of five major parts: (1) a *feature encoder block* which maps the original signal to the hidden space, so that further increases the express ability of model; (2) a *Transformer block*, this block's primary job is to infer the value of the initial hidden state based on the observable data for a future time period; (3) an *aggregate block* which mainly completes the fusion of static features and dynamic features, learnable parameter splicing, and high-dimensional mapping; (4) an *ODENet block* to integrate the output of the previous layer as an integrator; (5) a *feature decoder block* to recovery signal. In order to optimize the model, we may add a dropout layer after the decoding layer, and the final output of the network which is the predicted value will be used to calculate the error with the real value, the gradient descent method is

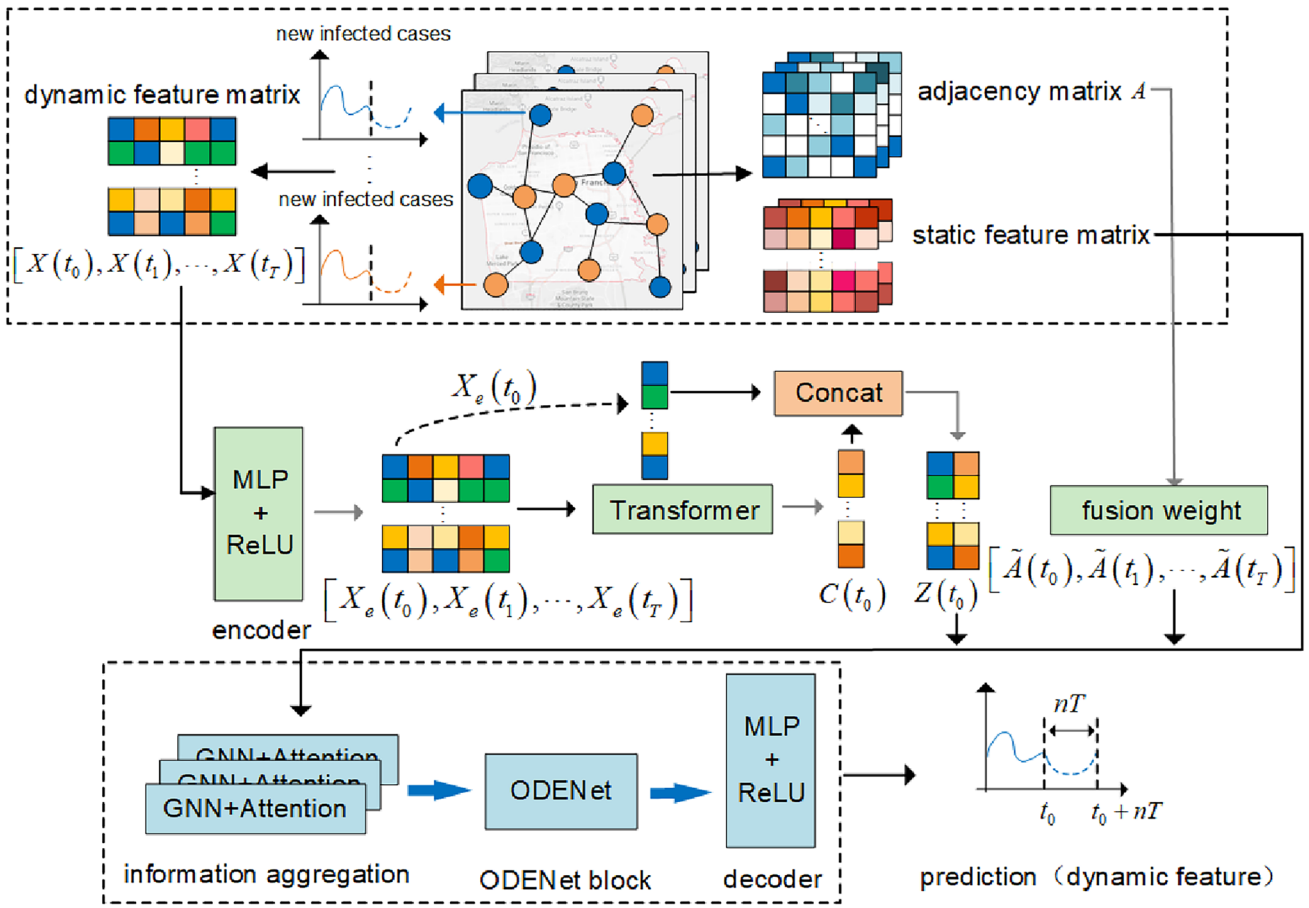



**Fig. 1** The overview of the problem and our proposed method

used to continuously optimize the model. We will elaborate details of each step tin the following sections.

*Feature Encoding* Taking into account the evolution of the epidemic, it is difficult to accurately obtain hidden state (features) including potential infected cases and so on, however, these data are crucial to the epidemic's spread, so we map the true input of network $X(t)$ into a high-dimensional space $X_e(t)$ to learn more abundant hidden state in the evolution of the epidemic. By using a feature encoding function $f_e$, let $X_e(t) \in \mathbb{R}^{N \times F_e}$ represent the matrix of hidden states of node features $X(t)$) for $N$ nodes.

$$f_e : X_e(t) = \sigma(X(t)W_e + b_e) \in \mathbb{R}^{N \times F_e}, \tag{2}$$

where $W_e \in \mathbb{R}^{N \times F_e}$, $b_e \in \mathbb{R}^{F_e}$ are model parameters and $\sigma$ is activation function.

*Transformer Block* During the development of the epidemic, individuals experience multiple hidden states, including the latent state. In a region, there may be many individuals who are still in the incubation period, and the data of these individuals cannot be observed for the statistical indicators of the entire region (e.g., the cumulative number of infections). However, from the perspective of time and epidemic dynamics, there is a temporal correlation that allows us to infer the number of people in the incubation period in the past period from the number of new infections in the future. This inference can be achieved using temporal information and past-to-future correlations. To capture temporal dependencies and extract these hidden states, we use a Transformer model to estimate the hidden state value at initial time. These estimates help us to extrapolate backwards from initial time to complete the reconstruction of the hidden state along the way.

Given its observable state $X_e(t)$ for a period of time in the future, we employ the Transformer as a powerful translator to infer the hidden state $C(t_0)$ at the start:

$$C(t_0) = Transformer(X_e(t_0), X_e(t_1), ..., X_e(t_{M-1})) \in \mathbb{R}^{N \times F_e}, \tag{3}$$

where $t_0$ are initial time, $M$ denotes that we utilize a time series of observable states of length $M$ to calculate the hidden state at time $t_0$. Among the many existing works, the state of the entire network during the training process is uniquely determined by $X(t)$, thus the status of the network cannot be memorized. Inspired by the LSTM (Graves 2012)

model, we try to set two state variables including $X(t)$ and $C(t)$, where $C(t)$ is hidden state mentioned above. At this time, the network state is not only determined by $X(t)$, but also determined by $C(t)$, which means that when the same $X(t)$ is input, if the corresponding $C(t)$ is different, the state of the network is also different, $X(t)$ can not completely determine the network state. Specifically, we concatenate the encoded $X(t)$, which is $X_e(t)$, and $C(t)$ into a new vector $Z(t)$ as follows:

$$Z(t) = [X_e(t), C(t)] \in \mathbb{R}^{N \times 2F_e}, \tag{4}$$

where [*,*] means concatenate two vectors, and the $Z(t)$ will be the true input of our model.

*Fusion Matrix* Given adjacency matrix $A_g \in \mathbb{R}^{N \times N}$, we generally normalize it to $\tilde{A_g} = I + D^{-\frac{1}{2}} A_g D^{-\frac{1}{2}}$, where $D$ is the degree matrix of $A_g$. Motivated by the Graph Neural Diffusion (GRAND) (Chamberlain et al. 2021), the information aggregation between core node and its neiborghoods is modeled with an attention function $a(Z_i, Z_j)$, here we use the scaled dot product attention $a(Z_i, Z_j)$ (Vaswani et al. 2017) which corresponding attention matrix is $A$, and its form is:

$$a(Z_i, Z_j) = softmax(\frac{(W_K Z_i)^T (W_Q Z_j)}{d_k}). \tag{5}$$

Considering that the information aggregation of the core node is affected by both the edge features and the neighbor node features, we use a novel attention function $a(Z_i, Z_j, \tilde{A}_{gij})$ to combine edge features with node features, thus Equation (5) updates to:

$$a(Z_i, Z_j, \tilde{A}_{gij}) = softmax(\frac{(W_K Z_i)^T (W_Q Z_j)}{d_k}) + b\tilde{A}_{gij} + c, \tag{6}$$

where $W_K$ and $W_Q$ are learnable matrices, $b$ and $c$ are learnable parameters, $d_k$ is a hyperparameter determining the dimension of $W_K$ and $W_Q$, $\tilde{A}_{gij}$ denotes the normalized number of visitors from region $i$ to region $j$. We use the multi-head attention which is useful to stabilise the learning by taking the expectation, $\hat{A}(Z) = \frac{1}{h} \sum_h \hat{A}^h(Z)$, where the fusion attention weight matrix $\hat{A}$ with $\hat{A}_{ij} = a(Z_i, Z_j, \tilde{A}_{gij})$, a more general form is

$$\hat{A} = A + b\tilde{A_g} + c, \tag{7}$$

the fusion matrix $\hat{A}$ will participate in the relevant operation of GNN model as the final adjacency matrix,

*ODENet* In our study, we adopt the ODENet to achieve the continuous time prediction. ODENet extends time $t$ from discrete to continuous, and assuming that its latent variable is $X(t)$, and the $X(t)$ to $t$ transfer function can be expressed as follows:

$$\frac{dX(t)}{dt} = f(X(t), \theta, t). \tag{8}$$

Through the classical ODE solving method, it is possible to model or predict the value of $X(t)$ and its corresponding observed variable based on an irregular time series $[t_0, t_1, ...t_T]$. The process can be expressed by the following continuous-time(depth) model:

$$X(t_T) = X(0) + \int_{t_0}^{t_T} \frac{dX(\tau)}{d\tau} d\tau = X(0) + \int_{t_0}^{t_T} f(X(\tau), \theta, \tau) d\tau, \tag{9}$$

where $f(X(\tau), \theta, \tau)$ will be parameterised by a neural network to model the hidden dynamic, and without requiring any internal operations, we may backpropagate the process through an ODE solver, enabling us to construct it solely as a building block for the entire neural network.

*Aggregation Information* The continuous-time dynamics on a graph can be represented by an ODENet (Zang and Wang 2019), and Equation (8) can be converted into:

$$\frac{dX(t)}{dt} = f(X(t), G, W(t), t), \tag{10}$$

where $G$ represents a graph composed of $N$ nodes, $X(t) \in \mathbb{R}^{N \times F}$ represents the state of nodes on the graph at $t \in [0, T]$, $W(t)$ controls the evolution direction of the entire system, the function $f : \mathbb{R}^{N \times F} \to \mathbb{R}^{N \times F}$ regulates the instantaneous rate of change of the node state evolution on the graph.

In our study, we transform the regional-level epidemic forecasting problem into a graph-based spatial-temporal situational forecasting model on the continuous time domain. Therefore, the differential equation on the epidemic spreading network we defined can be defined as:

$$\frac{dZ(t)}{dt} = f(Z(t), G, W(t), t), \tag{11}$$

where $Z(t) \in \mathbb{R}^{N \times 2F_e}$ represents the state of nodes on the graph at $t \in [0, T]$ (including observed state and hidden state). The rest of the parameters have the same meanings as in Equation (10).

In the real process of epidemic transmission, in addition to the dynamic features such as the cumulative number of infected cases and the infection rate that change over time, the static features such as the population base and the average dwell time will also have a great impact on the epidemic spread, that is the larger the population base, the higher the number of infected people may be, and the longer the average dwell time, the higher the probability of being infected in a certain place will be Chinazzi et al. (2020); Kucharski et al. (2020) [43]. Therefore we use the following equation to combine the static features and dynamic features of nodes:

$$Z(t) = \sigma(\hat{A}Z_d(t)W_1 + \hat{A}Z_sW_2 + b), \tag{12}$$

where $Z_d(t) \in \mathbb{R}^{N\times 2F_e}$ denotes the dynamic features, that is, $Z(t)$ in Equation , $Z_s \in \mathbb{R}^{N\times F_s}$ denotes the static features, $F_s$ denotes the dimension of the static features, $\sigma$ is the activation function and generally use the ReLU, $\hat{A} \in \mathbb{R}^{N\times N}$ represents the fusion matrix mentioned above, $W_1 \in \mathbb{R}^{2F_e\times y}$ and $W_2 \in \mathbb{R}^{F_s\times y}$ are the weights of the linear connection layer, $y$ is the dimension of final output.

*Feature Decoding* The model's output, that is, the predicted value of the dynamic features $X_e(t_T)$, will be extracted from the output $Z(t_T)$ of the previous layer, and the final predicted value $X(t_T)$ is obtained through the decoding function $f_d$:

$$f_d : X(t_T) = \sigma(X_e(t_T)W_d + b_d) \in \mathbb{R}^{N\times F}, \tag{13}$$

where $W_d \in \mathbb{R}^{y\times F}$, $F$ is the feature dimension of the desired output. In summary, our model is as follows:

$$\begin{cases} X_e(t) & = f_e(X(t)), \\ C(t_0) & = Transformer(X_e(t_0), X_e(t_2), ..., X_e(t_{M-1})), \\ Z(t) & = [X_e(t), C(t)], \\ \frac{dZ(t)}{dt} & = ReLU(\hat{A}(ReLU(\hat{A}Z_d(t)W_1 + \hat{A}Z_sW_2 + b))W_3 + b), t \in [t_0, t_T], \\ X(t_T) & = f_d(X_e(t_T)). \end{cases} \tag{14}$$

the first equation projects the original input $X(t)$ (dynamic features) , which is mentioned in Definition 2, into the hidden space and convert $X_e(t)$; the second equation, based on the Transformer, calculates the value of the hidden state $C(t_0)$ at the beginning based on the observable data $(X_e(t_0), X_e(t_1), ..., X_e(t_{M-1}))$ with a length of M in the future; the third equatio(t)n concatenates $X_e(t)$ and hidden state $C(t)$ as $Z(t)$, which represents the state of networks, and then a GCN layer will be used to combine the dynamic features $Z_d(t) \in \mathbb{R}^{N\times 2F_e}$ and static features $Z_s \in \mathbb{R}^{N\times F_s}$ mentioned above to form the ultimate input signal of GCN layer; the fourth equation will use $\frac{dZ(t)}{dt} = ReLU(\hat{A}(ReLU(\hat{A}Z_d(t)W_1 + \hat{A}X_sW_2 + b))W_3 + b) = f(Z(t), G, W(t), t)$ to control the hidden state in the high-dimensional space, and achieves the continuous-time dynamics prediction corresponding to any time by the method of numerical integration; the last equation converts the signal from the high-dimensional space back to the original low-dimensional space, and use a part of $Z(t_T)$ to compare with ground truth and simulate the prediction error, where $t_T$ represents the predicted endpoint timestamp.

### 3.3 Training

*Optimization* We compare each node's prediction value to the relevant ground truth and then using gradient descent to optimise a $L1$-norm loss:

$$\mathcal{L}(\Theta) = \int_0^T \left|X(t) - \hat{X(t)}\right| dt + \lambda\mathcal{R}(\Theta), \tag{15}$$

where $\hat{X(t)} \in \mathbb{R}^{N\times F}$ is the ture value available at time stamp $t \in [0, T]$, the $X(t)$ is the predicted value, and the $|.|$ denotes value difference between $X(t)$ and $\hat{X(t)}$ at time $t \in [0, T]$, $\Theta$ represents all of the training parameters, and $R(\Theta)$ is the regularization term (e.g. $L1$-norm).

**Algorithm 1** Graph Neural Ordinary Differential Equations

**Input:** Node dynamic feature sequences $X_d(t)$, Node static features sequences $X_s$, Normalized visit matrix sequences $\tilde{A}_g = (\tilde{A}_{g1}, \tilde{A}_{g2}, \ldots, \tilde{A}_{gT})$
**Output:** Optimum $\Theta$

```
for Each sampled mini batch do
    for i in 0,1,...,T do
        X_e(t_i) ← f_e(X(t_i))
        C(t_i) ← Transformer(X_e(t_i), X_e(t_{i+1}), ..., X_e(t_{i+M-1}))
        Z(t_i) ← [X_e(t_i), C(t_i)]
        f(θ) ← GCN(Z_d(t_i), Z_s, Ã_i)
        Z(t_{i+1}) ← ODESolve(f(θ), Z(t_i), (t_i, t_{i+1}))
        X(t_{i+1}) ← f_d(X_e(t_{i+1}));
    for each node pair (Z_i, Z_j) do
        â_ij ← FusionWeight(Z_i, Z_j, Ã_gij);
    Update the parameters Θ by optimizing L loss in equation (15);
```

## 4 Experiments

In this section, we will describe in detail the datasets, experimental settings, performance metrics, the baseline methods and their parametric settings, and the comparable results.

*Datasets* In this study, we focus on regional-level outbreak forecasting, including POIs and CBGs, and here we will introduce where the data used in the experiment was obtained from.

1. ***SafeGraph Mobility Data***: The datasets used in our experiment are all from SafeGraph,[2] including visit matrixs from origin CBGs to destination POIs, each CBG's population, each POI's average dwell time indicating indicates how long the individual stays in the region on average, estimated area, longitud and latitud, where each visit matrix is used as the metric of inter region mobility flow.[3] Here we mainly collect the relevant data of San Francisco for the experiments. It should be noted that due to there are 28713 POIs and 2943 CBGs, that is the each access matrix $U$ between them is not a square matrix, and considering that the matrix is too large computer computing power does not support it, so we have moderately aggregated POIs based on postal code.
2. ***COVID-19 data***: We use the concepts presented in the article [43] to develop a fine-grained epidemic simulator that simulates the state of epidemic spread from March 1, 2020 to May 2, 2020. The simulator superimposes an SEIR model on each CBG, taking into account information such as inter-regional visits, regional area, and regional population, and can fit the daily confirmed cases[4] in cities observed within the proposed period accurately. This also implies that we will be able to conduct reliable research using relevant epidemic data simulated by the simulator. We would like to emphasize that our study considers strain characteristics, specifically the length of the incubation period, during both the early and late stages of the epidemic (Alpha variant strain period and Omicron variant strain period). Based on this consideration, we make adjustments to the relevant parameters in our simulator to ensure the accuracy of our simulations. As a result, we obtain two distinct datasets, one for each variant strain.

*Performance Metrics* The *Mean Absolute Error (MAE)*, a measure of the absolute difference between two variables, the *Root Mean Square Error (RMSE)*, a measure of the divergence between the observed value and the real value, and *Mean Absolute Percentage Error (MAPE)* are the metrics employed in our experiment to assess the forecasting ability. The smaller the value of MAE, RMSE and MAPE, the better the model performance. The *Time Granularity* (abbreviated as *TG*) of time series data refers to the time interval or size of the time unit between each data point.

*Baselines* We use the following baseline methods for comparison, including the disease transmission model, the classic time series forecasting model, and its variants, as well as spatiotemporal prediction model: (1) infectious disease dynamics model: *SIR*, *SEIR* ; (2) recurrent neural network: *RNN* (Schuster and Paliwal 1997), *LSTM* (Graves 2012), *GRU* (Cho et al. 2014), *DCRNN* (Li et al. 2017; 3) graph neural network: *GCN* (Bruna et al. 2013), *GAT* (Veličković et al. 2017); spatiotemporal prediction model: *STGCN* (Yu et al. 2017), *STGODE* (Fang et al, 2021), *STAN* (Gao et al. 2021).

### 4.1 Learning regularly sampled data

*Settings and Implementation Details* We refer to existing research results and set the incubation period to 5 days (120 h) and 3.4 days (81.6 h) to simulate two data sets (Wu et al. 2022), which we call Raw Dataset and Omicron Dataset. We perform epidemic prediction experiments at various time granularities using statistical data from the total infected cases at 1512 h (24 h per day) from March 1, 2020 to May 5, 2020, including hourly level (TG = 1 h), daily level (TG = 24 h), bi-daily level (TG = 48 h), and weekly level (TG = 168 h). Since the access between CBGs and POIs is 168 h periodic, here we take 168 access matrices $U = (U_1, U_2, ..., U_T)$, and use them to construct visit matrices $A_g = (A_{g1}, A_{g2}, ..., A_{gT})$ mentioned above.

In our model, $F$ sets to 1, $F_s$ sets to 4, the hidden dimensions of encoder and decoder are set to 16, 16, and the length of observable data input to the Transformer module is set to 7. We split all datasets with a ratio 50:13 into training sets and test sets, batchsize is 16, random seed sets to 3407 (Picard 2021). During the training process, a learning rate of $5^{-3}$ is employed for the Raw dataset, while a learning rate of $10^{-4}$ is utilized for the Omicron dataset, the optimizer is Adam, all experiments are conducted using the PyTorch framework and trained on a NVIDIA A100 GPU with 80 GB memory. It should be noted that during the training process of the model, the data used to deduce the initial value of the hidden state through the Transformer block will not appear in the test set, so it can be guaranteed that there will be no future information leakage in our experiments.

*Forecasting Performance* We evaluated our approach in both short-term (TG = 1 h, 24 h) and long-term (TG = 48 h, 168 h) settings based on Raw Dataset and Omicon Dataset.

[2] https://www.safegraph.com.

[3] https://docs.safegraph.com/docs/places-schema.

[4] https://github.com/nytimes/covid-19-data.

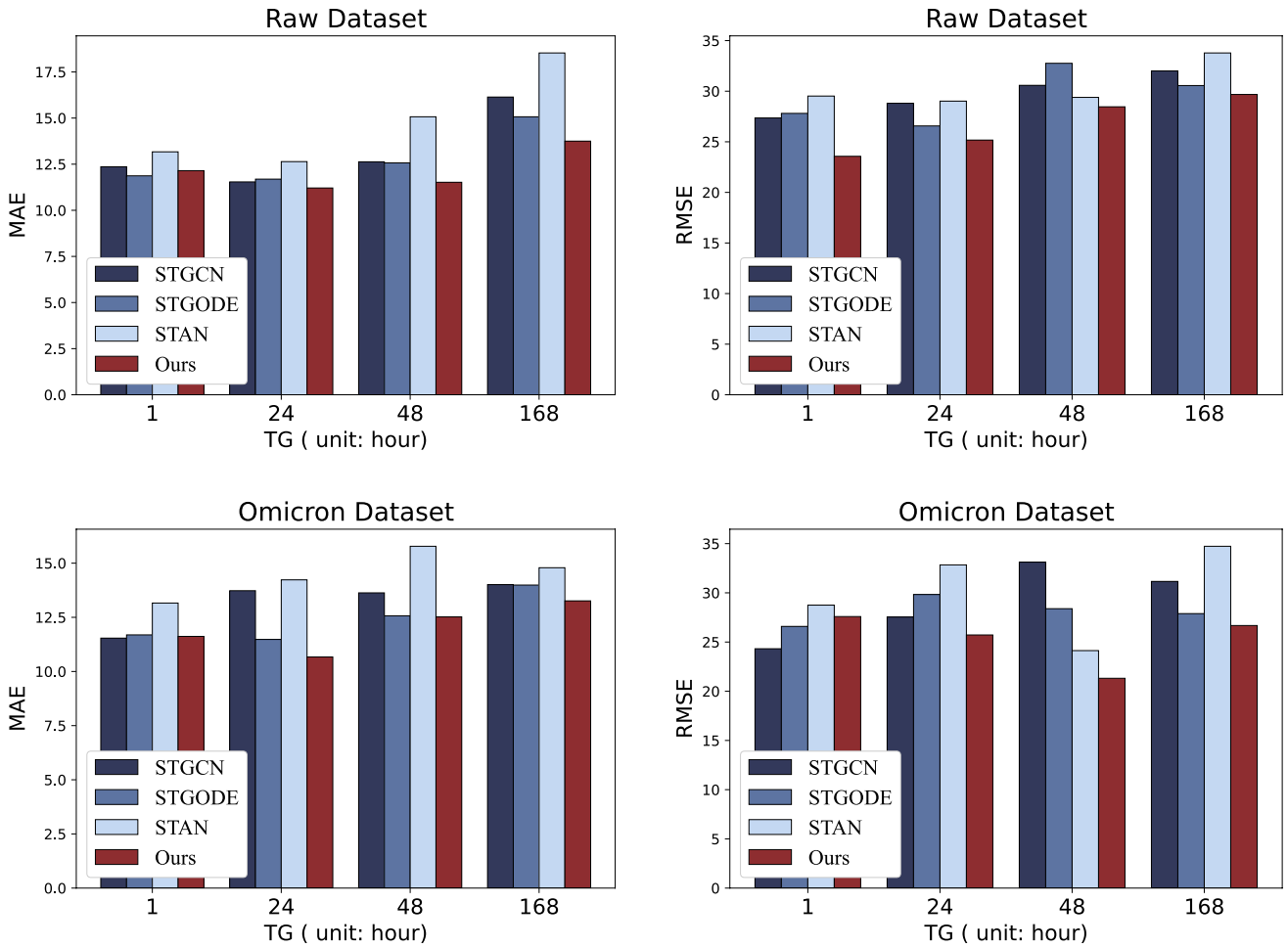



**Fig. 2** Performance of MAE and RMSE computed across two datasets across various TG settings

Table 2 displays our model's performance compared to a number of baselines.

We discover that our model has significant performance advantages at different time granularities, particularly in long-term prediction tasks. SIR and SEIR models have poor predictive performance compared to data-driven neural network models. Epidemic spread is typically characterized by complex nonlinear relationships, such as contact networks, behavioral changes, and environmental factors, among others, whereas SIR and SEIR models typically rely on a prior set parameters and are incapable of capturing these complex nonlinear relationships. Our model employs GNN to model the connection and interaction process between regions, and the results demonstrate the effectiveness of our design.

Based on the findings presented in rows 6–10 of Table 2, it is evident that conventional time series models exhibit suboptimal performance in either short-term forecasting (e.g., LSTM) or long-term forecasting (e.g., RNN, GRU), due to ignoring nonlinear relationships and complex interaction patterns in the data. Classical GNN-based models (e.g., GCN, GAT), primarily emphasize local node connections and information propagation from neighboring nodes in processing graph data, however, these models exhibit limited capability in capturing long-term dependencies, resulting in the absence of significant performance advantages. This observation emphasizes the significance of incorporating both temporal and spatial interdependencies when modeling epidemic propagation, as it plays a critical role in achieving superior predictive accuracy.

We focus on the performance differences between the three spatiotemporal convolution models, STAN, STGCN and STGODE, and our proposed model. Figure 3 compares the predicted values and true values of these four models on some nodes, Fig. 2 presents their prediction errors under different TG settings. The results show that our proposed model can learn potential patterns in the data more accurately. The traffic prediction models (e.g., STGCN, STGODE) focus primarily on the spatiotemporal dependence of local neighbourhoods. For longer-term spatiotemporal data, these model may be limited by the local spatiotemporal relationship in a

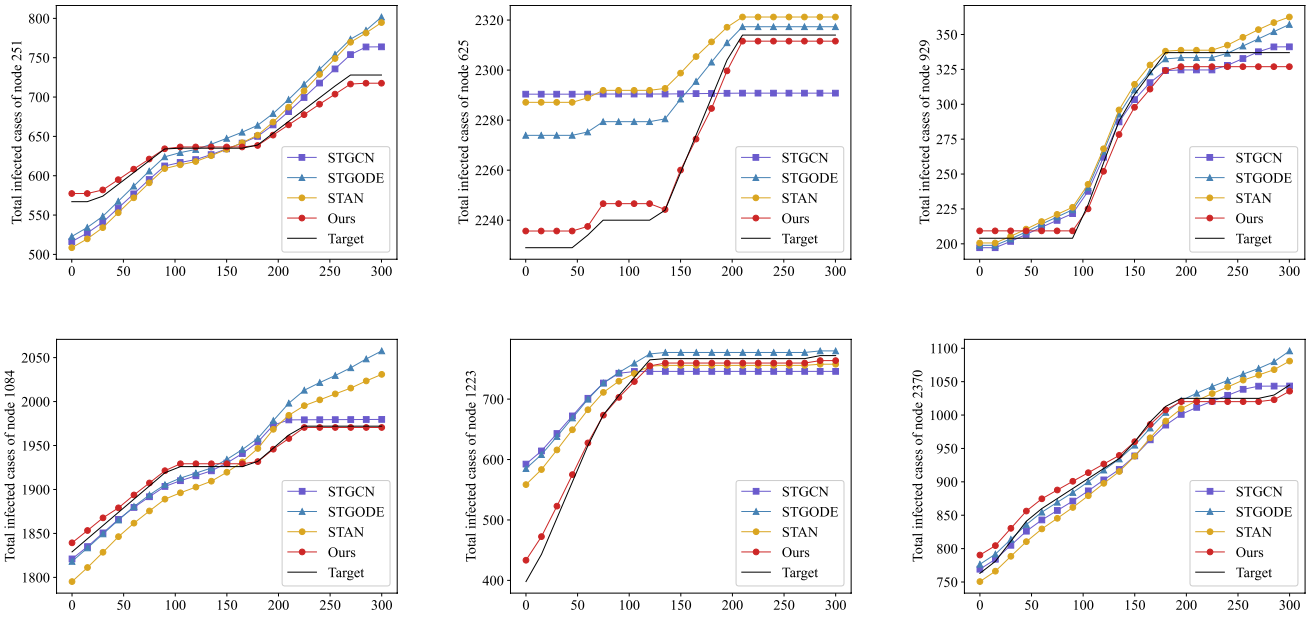



**Fig. 3** Prediction and Target for the Raw Dataset, consisting of total infected cases in 6 nodes base on different models f or the period from April 20, 2020 to May 2, 2020 (312 h)

**Table 2** Performance of different models in terms of key usability metrics on the two datasets for long-term prediction (TG = 168 h) and short-term prediction (TG = 24 h)

| | Raw dataset | | | | Omicron dataset | | | |
|---|---|---|---|---|---|---|---|---|
| | TG=24 h | | TG=168 h | | TG=24 h | | TG=168 h | |
| | MAE | RMSE | MAE | RMSE | MAE | RMSE | MAE | RMSE |
| SIR | 170.98 | 355.79 | 203.47 | 386.24 | 154.26 | 278.85 | 254.89 | 356.22 |
| SEIR | 120.31 | 290.67 | 239.13 | 417.54 | 139.76 | 211.53 | 217.36 | 424.59 |
| RNN | 81.31 | 143.39 | 66.12 | 190.76 | 70.03 | 198.64 | 67.32 | 152.75 |
| GRU | 42.13 | 84.63 | 46.27 | 125.79 | 60.52 | 131.29 | 47.49 | 151.26 |
| LSTM | 59.89 | 173.29 | 62.59 | 190.32 | 29.76 | 65.49 | 26.37 | 68.52 |
| GCN | 61.45 | 171.16 | 59.61 | 174.17 | 59.12 | 163.89 | 73.10 | 141.32 |
| GAT | 46.56 | 119.87 | 42.87 | 100.19 | 47.76 | 112.96 | 39.70 | 135.87 |
| STGCN | *11.54* | 28.81 | 16.14 | 32.01 | 13.73 | *27.54* | 14.01 | 31.16 |
| STGODE | 11.69 | *26.57* | *15.07* | *30.56* | *11.48* | 29.83 | *13.99* | *27.89* |
| STAN | 12.64 | 29.02 | 18.53 | 33.78 | 14.23 | 32.84 | 14.79 | 34.73 |
| Ours | **11.21** | **25.17** | **13.75** | **29.68** | **10.67** | **25.72** | **13.26** | **26.68** |
| Percentages | **2.86%** | **5.27%** | **2.12%** | **2.88%** | **7.06%** | **6.61%** | **5.22%** | **4.34%** |

All metrics are evaluated based on MAE and RMSE with lower values indicating better performance. In addition, the best result and the second-best result of each metric are marked with **bold** and *italic*, respectively

shorter time window and cannot fully consider the longer-term spatiotemporal dynamics.

The fact that our model performs well across all metrics for all datasets implies that our approach to modeling spatial-temporal dependency is successful.

*Ablation Study* We conduct the ablation studies based on the Raw Dataset and Omicron Dataset to assess the effects of each component in our framework. The results are presented in Table 3. By analyzing ablation models, we demonstrate that our model is the minimal model required for this task. We generate the following baselines while keeping the loss function constant.

**Table 3** Performance of ablation model in terms of key usability metrics on the Raw Dataset and Omicron Dataset for long-term prediction (TG = 168 h), the best result and the second-best result of each metric are marked with bold and italic, respectively

| | Raw dataset | | Omicron dataset | |
|---|---|---|---|---|
| | MAE | RMSE | MAE | RMSE |
| Ours | **14.75** | **33.68** | **13.76** | **24.68** |
| Ours w/o ED | 16.25 | 38.01 | 15.79 | 36.80 |
| Ours w/o FW | 25.14 | 66.01 | 24.34 | 59.32 |
| Ours w/o transformer | *14.79* | *34.93* | *13.81* | *25.49* |
| Percentages | **4.19%** | **3.58%** | **0.36%** | **3.18%** |

- Our model w/o ED: The model without encoding function $f_e$ and decoding function $f_d$, which removes the first line and fifth line in Equation (14), and here we call the removed components as ED.
- Our model w/o FW: The model uses a normalized adjacency matrix $\tilde{A_g}$ rather than fusion weight $\hat{A}$ (corresponding Equation (7)), which means we replace $\hat{A}$ with $\tilde{A_g}$ in Equation (14), and the removed components are referred to as FW.
- Our model w/o Transformer: The model uses a randomly initialized learnable hidden state $C(t)$ instead of inverting it with the aid of a Transformer block.

We summarize the observation as follows: three main components including encoding (decoding) function, fusion weight function, and Transformer block of our proposed model are to be effective. As shown in line 5, the adaption of the fusion weight function improves model performance on Raw Dataset and Omicron Dataset, it shows that the fusion weight constructed by us really well integrates the edge features and node features in the epidemic spread graph, and takes into account the interaction between edges and nodes, so as to improve the sensitivity and fitting ability of the model to epidemic data. By observing the line 6, we can find that Transformer block has a small impact on the performance of the model, and its role is mainly to deduce the value of unobservable data from the future observable data. It can also be seen from line 4 that encoding the input signal can improve the prediction accuracy of the model and improve the fitting ability of the model to a certain extent.

*Verification Experiment* We can simulate the number of people in the incubation period because our simulator is based on the SEIR model. Here we add this verification experiment to demonstrate that our model can appropriately predict the number of people throughout the latent phase. The following are the specific experimental steps: we build a simple model consisting of three layers of Multilayer Perceptron (MLP), and divide the previous stage experiment's training set into a new training set and a test set in a 35:15 ratio. At this point, $C(t)$ is determined using the ODENet parameters from the previous stage. We utilize $C(t)$ as the new experiment's input, and the output is the expected number of infected people during the incubation period. The difference from previous experiments is that, in this study, instead of using $C(t)$ as the target or ground truth value, the latent period population generated based on the simulator is used as the target value. Subsequently, the discrepancy between these target values and the output of the three-layer multilayer perceptron (MLP) is calculated to complete the backpropagation process. Table 4 shows the two data sets and experimental results for different TG, it can be concluded that our model can accurately predict the number of people in the latent period.

## 4.2 Learning irregularly sampled data

To validate that our model can achieve dynamic epidemic predictions based on irregularly sampled data, we enhanced the model training strategy building upon the experiments in Sect. 4.1. Subsequently, we conducted additional experiments on two datasets.

*Settings and Implementation Details* We employed a probability-based multi-scale training strategy in the following experiments, to validate the capability of our model in addressing the challenge of irregularly sampled data in epidemic prediction. In particular, we partitioned the data into training and testing sets in a 50:13 ratio. During the model training process, we assigned probabilities to sampling instances at intervals of $S_1 = 24$, $S_2 = 48$, and $S_3 = 168$, denoted as $P_1 = 1/6$, $P_2 = 1/2$, and $P_3 = 1/3$, respectively. These probabilities represent the likelihood of sampling epidemic data every $S_i$ hours in real life, for instance, sampling at $S_1$ implies predicting the cumulative number of infections 24 h later. In the model testing phase, we conducted experiments to predict data 24 h and 168 h ahead.

**Table 4** Performance of verification experiment on the Raw Dataset and Omicron Dataset for different TG settings

| | Raw dataset | | | Omicron dataset | | |
|---|---|---|---|---|---|---|
| Metrics | MAE | RMSE | MAPE | MAE | RMSE | MAPE |
| TG = 1 h | 378.42 | 721.98 | 0.27 | 296.93 | 566.71 | 0.28 |
| TG = 24 h | 162.76 | 238.34 | 0.18 | 157.98 | 217.56 | 0.17 |
| TG = 168 h | 73.56 | 153.29 | 0.11 | 68.57 | 117.59 | 0.09 |

**Table 5** Performance on irregularly sampled data, the best result and the second-best result of each metric are marked with bold and italic, respectively

| | Raw dataset | | | | Omicron dataset | | | |
|---|---|---|---|---|---|---|---|---|
| | TG=24 h | | TG=168 h | | TG=24 h | | TG=168 h | |
| | MAE | RMSE | MAE | RMSE | MAE | RMSE | MAE | RMSE |
| STGCN | 230.69 | 356.28 | *133.65* | 272.27 | 212.67 | 358.49 | 137.34 | 282.45 |
| STGODE | *136.31* | 267.37 | 138.21 | 275.32 | 144.25 | **256.64** | 136.23 | 273.55 |
| STAN | 238.62 | 470.45 | 146.67 | 264.61 | 235.56 | 367.38 | *130.67* | *266.52* |
| DCRNN | 144.56 | **195.43** | 152.25 | *207.64* | *139.86* | 376.33 | 150.99 | 312.67 |
| **Ours** | **127.90** | *254.24* | **130.66** | **190.32** | **135.54** | *268.43* | **127.17** | **255.37** |
| Percentages | **6.17%** | – | **2.24%** | **8.34%** | **3.09%** | – | **2.68%** | **4.18%** |

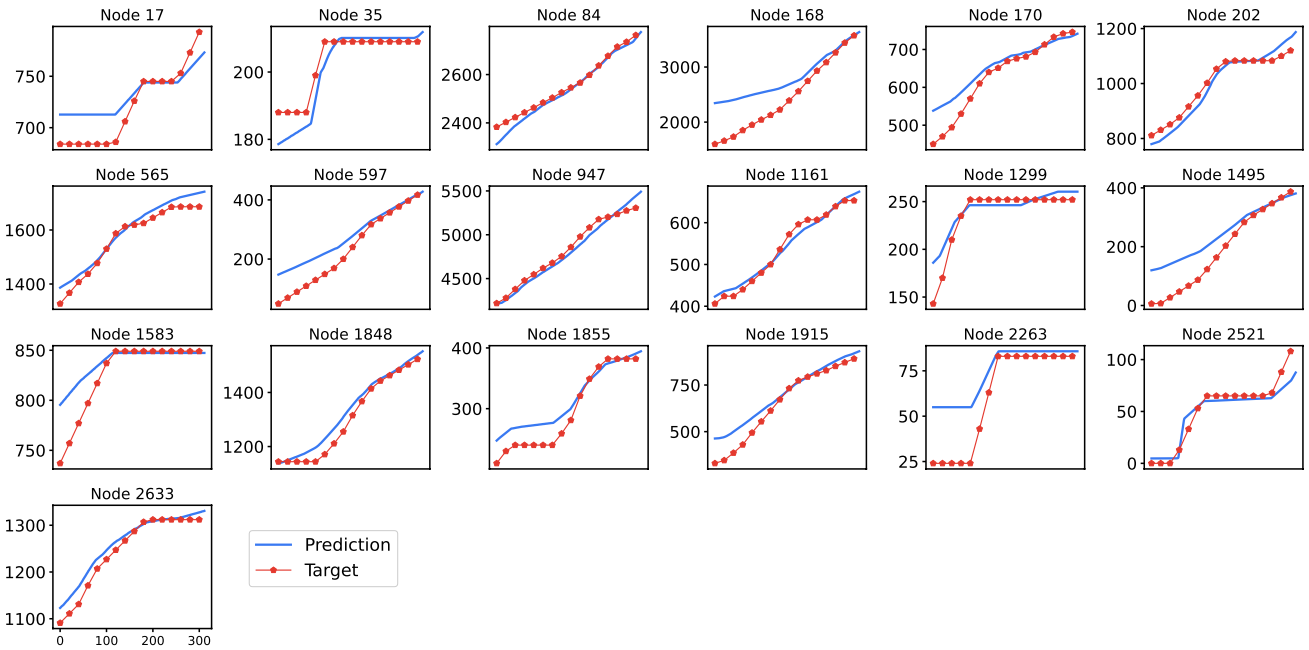



**Fig. 4** Prediction and Target for the Raw Dataset, consisting of total infected cases in the different nodes base on our model for the period from April 20, 2020 to May 2, 2020 (312 h)

*Forecasting Performance* Based on the experiments in the previous section, this section adds the DCRNN (Li et al. 2017) as a benchmark model, and we focuse on the difference in performance between the models proposed in this section and a series of spatio-temporal prediction benchmark models. Table 5 presents our experimental results. In two sets of datasets, our model demonstrates a significant advantage in long-term forecasting compared to classical spatiotemporal prediction models. Figure 4 illustrates the predicted cumulative infection numbers for numerous nodes based on our model. It can be observed that the epidemic development trends across nodes are highly diverse. Our model demonstrates the ability to accurately learn dynamic epidemic evolution patterns from non-uniformly sampled data, enabling precise predictions. During the experiment, we also noticed some unusual cases, such as the fact that the cumulative number of infected people in some nodes showed no discernible trend of change. We believe that this phenomenon is consistent with the fact that some areas of epidemic prevention and control work are performing admirably.

## 5 Conclusion

In this study, we propose a novel graph neural ordinary differential equations approach which combines GNN and ODENet to make fine-grained epidemic predictions. We have conducted a large number of experiments to prove the effectiveness of our proposed model, and its prediction

performance is better than the benchmark model. In our work, Multilayer Perceptron (MLP) encoders, decoders, and nonlinear high-dimensional maps are used to simulate the highly complex process of epidemic evolution and propagation. We creatively express the state evolution rate on the epidemic transmission network in the form of GNN and ODENet, and use the strong expression ability and interpretability of ODENet to make more flexible and accurate predictions. In addition, we design an fusion matrix constructed by attention matrix and geographic adjacency matrix to fully aggregate information from the dimensions of nodes and edges and improve the performance of the model, and we introduce a Transformer block to deduce the initial value of the hidden state based on the observed data in the future for a period of time, and use experiments to prove that our model can indeed obtain excellent performance under this task. We conducted experiments on two datasets, explored the short-term prediction performance and long-term prediction performance of the model by setting different time granularity, and verified the feasibility and efficiency of our proposed model. In future work, we can deeply explore the interpretability of the proposed model, try to enhance the flexibility and universality of the model, and apply the results to the formulation of epidemic prevention policies to provide references for policymakers with vivid visual effects.

**Acknowledgements** This work was supported in part by the National Natural Science Foundation of China under U21B2036, 62171260, and the Young Elite Scientists Sponsorship Program by CIC (Grant No. 2021QNRC001).

**Data availability** All the data used in this study can be obtained from https://github.com/xiongzhangxyq/GNNODENet.

## Declarations

**Conflict of interest** The author discloses no financial or non-financial interests that are directly or indirectly related to the work submitted for publication. In addition, there is no Conflict of interest with the Editor-in-Chief and Associate Editors.

## References

Bai, S., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. (2018) arXiv preprint arXiv:1803.01271

Balcan, D., Colizza, V., Gonçalves, B., Hu, H., Ramasco, J.J., Vespignani, A.: Multiscale mobility networks and the spatial spreading of infectious diseases. PNAS **106**(51), 21484–21489 (2009)

Bhatia, A., Pasari, S., Mehta, A.: Earthquake forecasting using artificial neural networks. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. **42**, 823–827 (2018)

Bruna, J., Zaremba, W., Szlam, A., LeCun, Y.: Spectral networks and locally connected networks on graphs (2013) arXiv preprint arXiv:1312.6203

Chamberlain, B., Rowbottom, J., Gorinova, M.I., Bronstein, M., Webb, S., Rossi, E.: Grand: Graph neural diffusion. In: International Conference on Machine Learning, pp. 1407–1418 (2021). PMLR

Chang, S., Pierson, E., Koh, P.W., Gerardin, J., Redbird, B., Grusky, D., Leskovec, J.: Mobility network models of covid-19 explain inequities and inform reopening. Nature **589**(7840), 82–87 (2021)

Chen, R.T., Rubanova, Y., Bettencourt, J., Duvenaud, D.K.: Neural ordinary differential equations. Adv. Neural Inf. Process. Syst. 31 (2018)

Chinazzi, M., Davis, J.T., Ajelli, M., Gioannini, C., Litvinova, M., Merler, S., Piontti, A.P., Mu, K., Rossi, L., Sun, K.: The effect of travel restrictions on the spread of the: novel coronavirus (covid-19) outbreak. Science **368**(6489), 395–40 (2019)

Chinazzi, M., Davis, J.T., Ajelli, M., Gioannini, C., Litvinova, M., Merler, S., Piontti, A., Mu, K., Rossi, L., Sun, K.: The effect of travel restrictions on the spread of the 2019 novel coronavirus (covid-19) outbreak. Science **368**(6489), 395–400 (2020)

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation (2014) arXiv preprint arXiv:1406.1078

Deng, S., Wang, S., Rangwala, H., Wang, L., Ning, Y.: Cola-gnn: cross-location attention based graph neural networks for long-term ili prediction. In: Proceedings of CIKM (2020)

Fang, Z., Long, Q., Song, G., Xie, K.: Spatial-temporal graph ode networks for traffic flow forecasting. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 364–373 (2021)

Gao, J., Sharma, R., Qian, C., Glass, L.M., Spaeder, J., Romberg, J., Sun, J., Xiao, C.: Stan: spatio-temporal attention network for pandemic prediction using real-world evidence. J. Am. Med. Inf. Assoc. **28**(4), 733–743 (2021)

Graves, A.: Long short-term memory. Supervised sequence labelling with recurrent neural networks, pp. 37–45 (2012)

Huang, Z., Sun, Y., Wang, W.: Coupled graph ode for learning interacting system dynamics. In: The 27th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD) (2021)

Jin, M., Zheng, Y., Li, Y.-F., Chen, S., Yang, B., Pan, S.: Multivariate time series forecasting with dynamic graph neural odes. IEEE Trans. Knowl. Data Eng. (2022)

Kargas, N., Qian, C., Sidiropoulos, N.D., Xiao, C., Glass, L.M., Sun, J.: Stelar: spatio-temporal tensor factorization with latent epidemiological regularization. In: Proceedings of AAAI (2021)

Kleczkowski, A., Grenfell, B.T.: Mean-field-type equations for spread of epidemics: the 'small world' model. Phys. A **274**(1–2), 355–360 (1999)

Kucharski, A.J., Russell, T.W., Diamond, C., Liu, Y., Edmunds, J., Funk, S., Eggo, R.M., Sun, F., Jit, M., Munday, J.D.: Early dynamics of transmission and control of covid-19: a mathematical modelling study. Lancet. Infect. Dis **20**(5), 553–558 (2020)

Li, Q., Han, Z., Wu, X.-M.: Deeper insights into graph convolutional networks for semi-supervised learning. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)

Li, Y., Yu, R., Shahabi, C., Liu, Y.: Diffusion convolutional recurrent neural network: data-driven traffic forecasting (2017) arXiv preprint arXiv:1707.01926

Liu, Z., Shojaee, P., Reddy, C.K.: Graph-based multi-ode neural networks for spatio-temporal traffic forecasting (2023) arXiv preprint arXiv:2305.18687

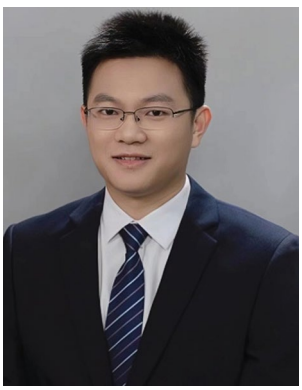
Liu, Y., Zheng, H., Feng, X., Chen, Z.: Short-term traffic flow prediction with conv-lstm. In: 2017 9th International Conference on Wireless Communications and Signal Processing (WCSP), pp. 1–6 (2017). IEEE

Maier, B.F., Brockmann, D.: Effective containment explains subexponential growth in recent confirmed covid-19 cases in china. Science **368**(6492), 742–746 (2020)
Panagopoulos, G., Nikolentzos, G., Vazirgiannis, M.: Transfer graph neural networks for pandemic forecasting. In: Proceedings of AAAI (2021)
Picard, D.: Torch. manual_seed (3407) is all you need: on the influence of random seeds in deep learning architectures for computer vision (2021) arXiv preprint arXiv:2109.08203
Poli, M., Massaroli, S., Park, J., Yamashita, A., Asama, H., Park, J.: Graph neural ordinary differential equations (2019) arXiv preprint arXiv:1911.07532
Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Trans. Signal Process. **45**(11), 2673–2681 (1997)
Sesti, N., Garau-Luis, J.J., Crawley, E., Cameron, B.: Integrating lstms and gnns for covid-19 forecasting (2021) arXiv preprint arXiv:2108.10052
Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., Woo, W.-C.: Convolutional lstm network: a machine learning approach for precipitation nowcasting. Adv. Neural Inf. Process. Syst. 28 (2015)
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Adv. Neural Inf. Process. Syst. 30 (2017)
Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks (2017) arXiv preprint arXiv:1710.10903
Voita, E., Talbot, D., Moiseev, F., Sennrich, R., Titov, I.: Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned (2019) arXiv preprint arXiv:1905.09418
Wang, P., Zheng, X., Li, J., Zhu, B.: Prediction of epidemic trends in covid-19 with logistic model and machine learning technics. Chaos Solitons Fract. **139**, 110058 (2020)
Wang, L., Adiga, A., Chen, J., Sadilek, A., Venkatramanan, S., Marathe, M.: Causalgnn: Causal-based graph neural networks for spatio-temporal epidemic forecasting. In: Proceedings of AAAI (2022)
WHO: COVID-19 epidemiological update–22 December 2023. https://www.who.int/publications/m/item/covid-19-epidemiological-update---22-december-2023. Accessed December 25, 2023
Wu, Y., Yang, Y., Nishiura, H., Saitoh, M.: Deep learning for epidemiological predictions. In: Proceedings of ACM SIGIR (2018)
Wu, Y., Kang, L., Guo, Z., Liu, J., Liu, M., Liang, W.: Incubation period of covid-19 caused by unique sars-cov-2 strains: a systematic review and meta-analysis. JAMA Netw Open **5**(8), 2228008–2228008 (2022)
Xhonneux, L.-P., Qu, M., Tang, J.: Continuous graph neural networks. In: International Conference on Machine Learning, pp. 10432–10441 (2020). PMLR
Xie, F., Zhang, Z., Li, L., Zhou, B., Tan, Y.: Epignn: Exploring spatial transmission with graph neural network for regional epidemic forecasting (2022) arXiv preprint arXiv:2208.11517
Yang, H., Li, X., Qiang, W., Zhao, Y., Zhang, W., Tang, C.: A network traffic forecasting method based on sa optimized arima-bp neural network. Comput. Netw. **193**, 108102 (2021)
Yu, B., Yin, H., Zhu, Z.: Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting (2017) arXiv preprint arXiv:1709.04875
Zang, C., Wang, F.: Neural dynamics on complex networks (2019)
Zhang, X., Huang, C., Xu, Y., Xia, L., Dai, P., Bo, L., Zhang, J., Zheng, Y.: Traffic flow forecasting with spatial-temporal graph diffusion network. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 15008–15015 (2021)
Zhang, J., Litvinova, M., Liang, Y., Wang, Y., Yu, H.: Changes in contact patterns shape the dynamics of the covid-19 outbreak in china. Science, 8001 (2020)
Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., Sun, M.: Graph neural networks: a review of methods and applications. AI Open **1**, 57–81 (2020)
Zhu, X., Fu, B., Yang, Y., Ma, Y., Hao, J., Chen, S., Liu, S., Li, T., Liu, S., Guo, W.: Attention-based recurrent neural network for influenza epidemic prediction. BMC Bioinf. **20**(18), 1–10 (2019)



**Xiong Yanqin** received the B.E. degree in communication engineering from Minzu University of China in 2021. She is currently pursuing a Master's degree with the School of Electronics Information and Communications Huazhong University of Science and Technology, Wuhan. Her main research interests include data mining and urban scientific computing.

**Wang Huandong** received his B.S. degrees in electronic engineering and his second B.S. degree in mathematical sciences from Tsinghua University, Beijing, China, in 2014 and 2015, respectively, and he received the Ph.D. degree in electronic engineering from Tsinghua University, Beijing, China, in 2019. He is currently a research associate with the Department of Electronic Engineering, Tsinghua University. His research interests include urban computing, urban emergency, mobile big data mining, and machine learning.



**Liu Guanghua** is currently an Associate Professor with the Research Center of 6G Mobile Communications, School of Cyber Science and Engineering, Huazhong University of Science and Technology, China. His research interests include wireless underground/underwater communications and wireless power transfer.

**Li Yong** (Senior Member, IEEE) received the BS degree in electronics and information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2007, and the PhD degree in electronic engineering from Tsinghua University, Beijing, China, in 2012. He is currently a faculty member of the Department of Electronic Engineering, Tsinghua University. He has served as general chair, TPC chair, SPC/TPC member for several international workshops and conferences, and he is on the editorial board of two IEEE journals. His papers have total citations more than 6900. Among them, ten are ESI Highly Cited Papers in Computer Science, and four receive conference best paper (run-up) awards. He received IEEE 2016 ComSoc Asia-Pacific Outstanding Young Researchers, Young Talent Program of China Association for Science and Technology, and the National Youth Talent Support Program.

**Jiang Tao** received the B.S. and M.S. degrees in applied geophysics from China University of Geosciences, Wuhan, P. R. China, in 1997 and 2000, respectively, and the Ph.D. degree in information and communication engineering from Huazhong University of Science and Technology, Wuhan, P. R. China, in April 2004. Then, he joined the Brunel University, UK. In Oct. 2006, he moved to the University of Michigan, USA. He is now with the Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan, 430074, P. R. China. He has authored or co-authored over 40 technical papers in major journals and conferences and five books/chapters in the areas of communications. His current research interests include the areas of wireless communications and corresponding signal processing, especially for OFDM, UWB and MIMO systems, cooperative networks, cognitive radio and wireless sensor networks. He serves on the editorial boards of some international journals, such as Wiley of Wireless Communications and Mobile Computing (WCMC), and as a Technical Program Committee Member for some major international conferences, including IEEE Globecom, ICC, VTC and Chinacom. Dr. Jiang is a Member of IEEE Communication Society and IEEE Broadcasting Society.