Journal of the American Medical Informatics Association, 28(4), 2021, 733–743
doi: 10.1093/jamia/ocaa322
Advance Access Publication Date: 22 January 2021
Research and Applications

Research and Applications

STAN: spatio-temporal attention network for pandemic prediction using real-world evidence

Junyi Gao,[1,4] Rakshith Sharma,[2] Cheng Qian,[1] Lucas M. Glass,[1,3] Jeffrey Spaeder,[1] Justin Romberg,[2] Jimeng Sun,[4] and Cao Xiao[1]

[1]IQVIA, Cambridge, Massachusetts, USA, [2]Georgia Institute of Technology, Atlanta, Georgia, USA, [3]Temple University, Philadelphia, Pennsylvania, USA and [4]University of Illinois at Urbana-Champaign, Champaign, Illinois, USA

Corresponding Author: Cao Xiao, PhD, IQVIA, 201 Broadway Floor 5, Cambridge, MA 02139, USA (cao.xiao@iqvia.com)

Received 22 July 2020; Editorial Decision 25 November 2020; Accepted 2 December 2020

ABSTRACT

Objective: We aim to develop a hybrid model for earlier and more accurate predictions for the number of infected cases in pandemics by (1) using patients' claims data from different counties and states that capture local disease status and medical resource utilization; (2) utilizing demographic similarity and geographical proximity between locations; and (3) integrating pandemic transmission dynamics into a deep learning model.
Materials and Methods: We proposed a spatio-temporal attention network (STAN) for pandemic prediction. It uses a graph attention network to capture spatio-temporal trends of disease dynamics and to predict the number of cases for a fixed number of days into the future. We also designed a dynamics-based loss term for enhancing long-term predictions. STAN was tested using both real-world patient claims data and COVID-19 statistics over time across US counties.
Results: STAN outperforms traditional epidemiological models such as susceptible-infectious-recovered (SIR), susceptible-exposed-infectious-recovered (SEIR), and deep learning models on both long-term and short-term predictions, achieving up to 87% reduction in mean squared error compared to the best baseline prediction model.
Conclusions: By combining information from real-world claims data and disease case counts data, STAN can better predict disease status and medical resource utilization.

Key words: pandemic prediction, deep learning, graph attention network, real world evidence

INTRODUCTION

Epidemic/pandemic prediction models

Traditional epidemic prediction models use compartmental based models that estimate disease transmission dynamics at the population level, such as SIR and SEIR models and their variants.[2,3] Some works also utilize time series learning approaches for pandemic prediction, for example, applying curve-fitting[3] or autoregression.[5] Besides these traditional statistical models, deep learning models were developed to cast epidemic or pandemic modeling as time series prediction problems. Many works[6–8] combine deep neural networks (DNN) with causal models for influenza-like illness incidence forecasting. Deng *et al*[9] proposed a graph message passing framework to combine learned feature embeddings and an attention matrix to model disease propagation over time. Yang *et al*[3] used previous pandemic data to pretrain the LSTM and then apply it to predict COVID-19 progression in China. Kapoor *et al*[10] utilized a simple graph neural network (GNN) for COVID-19 prediction. However, these models only predict the next day instead of long-term progression. It is still challenging to make deep learning-based models achieve good long-term prediction performance. Moreover, DNN-based methods have a significant issue: they can only predict known trends from the input data without understanding the long-term progression trend. For example, at the early stage of the pandemic if all case counts are increasing, it is unlikely for these models to predict a declining trend in the future. Hence the long-term prediction is often difficult for DNN models.

Incorporate disease transmission dynamics in graph neural network

Recently, several studies have attempted to incorporate knowledge about physical systems into deep learning. For example, Wu *et al* and Beucler *et al*[11,12] introduced statistical and physical constraints in the loss function to regularize the model's predictions. However, their studies only focused on spatial modeling without temporal dynamics. Seo *et al*[13,14] integrated physical laws into GNNs. However, they focused on using physical laws to optimize node–edge transitions instead of concentrating on prediction results. In particular, those models only predict graph signals for the next time point instead of long-term outcomes. In our work, we also incorporate physics laws, (ie, disease transmission dynamics) to regularize model predictions to overcome the limitations of the prior models. These regularizations will be applied over a time range to ensure we can predict long-term pandemic progression. Since our proposed method is applied to extracted temporal and spatial embeddings of locations as an extra loss term, it does not introduce extra hyperparameters; hence, it is easier to train.

OBJECTIVE

Pandemic diseases, such as the novel coronavirus disease (COVID-19), have been spreading rapidly across the world and pose a severe threat to global public health. Up to July 2020, COVID-19 has affected 14.1 million people, caused more than 597K deaths worldwide,[1] and caused significant disruption to people's daily lives as well as substantial economic losses. Therefore, it is critical to be able to predict pandemic outbreaks early and accurately to help design appropriate policies and reduce losses.

Many epidemiological models (eg, susceptible-infected-removed [SIR], susceptible-exposed-infected-removed [SEIR]), and deep learning models (eg, long short-term memory [LSTM] networks) have been applied to predict the COVID-19 pandemic.[1–4] However, they face 3 major challenges: (1) they usually build a separate model for each location (eg, 1 model per county) without incorporating geographic proximity and interactions with nearby regions. Or the forecasts only depend on some observed patterns from other locations,[2,3] while interregional interactions are not directly modeled. In fact, a location often shows similar disease patterns with its nearby locations or demographically similar locations due to population movements or demographic similarity.[5] (2) Existing models are mainly built on COVID-19 case report data. These data are known to have severe underreporting or other data quality issues. (3) Epidemiological models such as SIR and SEIR are deterministic models. They use a set of differential equations to fit the entire curve of disease counts. These models are determined by only a few parameters, making them unable to capture complex short-term patterns, such as superinfection or time-varying infectivity.[4] Conversely, deep learning-based models can only predict known data patterns and lead to accurate predictions only within a short time period. Therefore, while there are techniques that allow for either short-term or long-term predictive models of disease outbreaks, existing models do not provide accurate models over both time horizons.

In this work, we propose a Spatio-Temporal Attention Network (STAN) for pandemic prediction using real-world evidence, such as claims data and COVID-19 case surveillance data. We map locations (eg, a county or a state) to nodes on a graph and construct the edges based on geographical proximity and demographic similarity between locations. Each node is associated with a set of static and dynamic features extracted from multiple real-world evidence in medical claims data that capture disease prevalence at different locations and medical resource utilization conditions. We utilize a graph attention network (GAT) to incorporate interactions of similar locations. Then we predict the number of infected patients for a fixed period into the future while concurrently imposing physical constraints on predictions according to transmission dynamics of epidemiological models. We apply STAN to predict both state-level and county-level future number of infected cases, achieving up to 87% reduction in mean squared error compared to the best baseline model. This study has been determined by the UIUC IRB as nonhuman subject research.

MATERIALS AND METHODS

Problem formulation

In this article, we develop the STAN model to predict the number of COVID-19 cases, for a fixed number of days into the future, at the county or state-level across the US, using the following input data: county-level historical daily numbers of positive cases, county-level population-related statistics, and the frequencies of relevant medical codes extracted from medical claims data. Our goal is to better predict the number of cases by utilizing the rich information captured by these different data sources.

Throughout the article, we use N to denote the number of spatial locations (counties), X to represent the feature matrix of size $N \times (F_S + T \times F_D)$, where F_S is the number of *static* features per county (or state), and F_D is the number of *dynamic* features for each county (or state). T denotes the total number of time steps (ie, days) for each location. Finally, we are interested in predicting $I(t)$, the number of infected patients at the t^{th} time step for all the locations.

As depicted in Figure 1, STAN is enabled by the following components: 1) a GNN that captures the geographic trends in disease transmission; 2) an RNN that captures the temporal disease patterns at each location; 3) Both short-term prediction loss and long-term transmission dynamics constraint loss to regularize learned hidden representations of node embeddings. We describe each of these aspects below.

Graph construction

The input data include dynamic data and static data. Dynamic data is a 3D tensor that includes location (eg, states, counties, etc), timestamp (eg, days/weeks), and the dynamic features at each location (eg, the number of active COVID-19 cases and the numbers of other related ICD codes). Static data is a 2D matrix that includes location and the static features for each location. We also form an *attributed graph* to capture the spatio-temporal epidemic/pandemic dynamics. In particular, we model the geographic proximity and demographic similarity between the different locations as edges in a location graph.

Graph nodes: We construct an attributed graph $G(\mathcal{V}, \mathcal{E})$ to represent the input data. A location is modeled as a graph node and is associated with a feature matrix that contains both static and dynamic features across all the timestamps for that location. In total, we have 193 nodes for the county-level graph (1 for each county with more

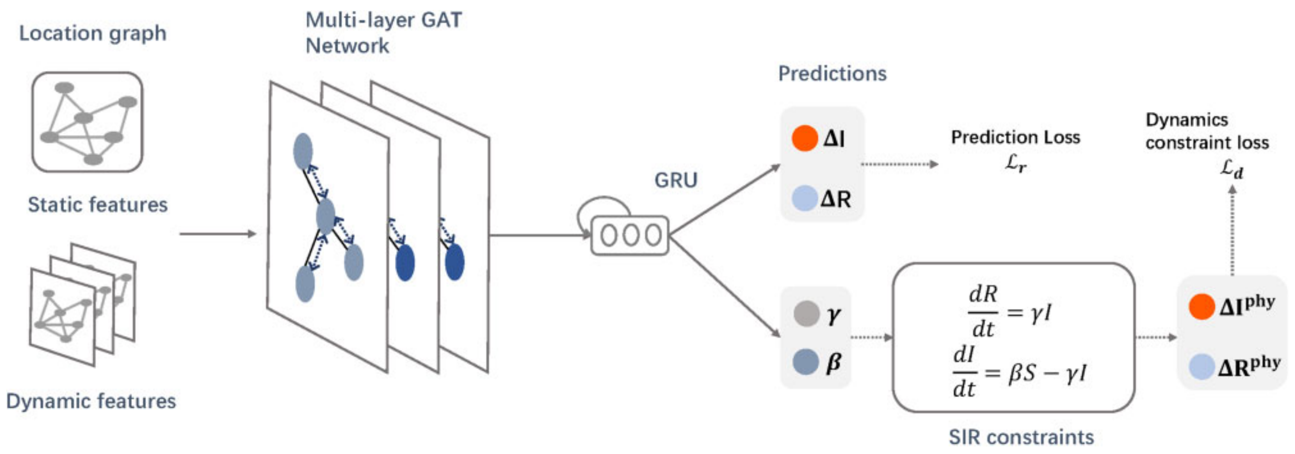

Figure 1. The STAN model: We construct the location graph using location-wise dynamic and static features as nodes and geographic proximity as edges. The graph is fed into graph attention network to extract spatio-temporal features and learn the graph embedding for the target location. Then the graph embedding is fed into the GRU to extract temporal relationships. The hidden states of GRU will be used to predict future number of infected and recovered cases. We use an additional transmission dynamics loss based on pandemic transmission dynamics to optimize the model.

than 1000 infected cases) or 45 nodes for the state-level graph (1 for each state).

Graph edges: The edges are constructed based on geographical proximity and the population sizes of the nodes (ie, locations). In particular, we designate the weight of an edge between nodes i and j as $w_{ij} \propto p_i^{\alpha} p_j^{\beta} \exp\left(-\frac{d_{ij}}{r}\right)$, where p_i and p_j are the population size of the nodes, d_{ij} is the geographical distance between them, α, β and r are hyperparameters. The above model is based on the idea that disease transmission patterns highly depend on crowd mobility. If there is a high mobility rate between a pair of nodes, we expect that the nodes have the similar disease spread parameters. Hence, our process includes an edge with a large weight between such a pair of nodes. Note that the distance parameter d_{ij} can incorporate any general notion of distance, including inverse of the volume of air and car travels.

Node features (static): Each node n_i has an associated static feature vector of size 4, consisting of the static features including latitude, longitude, population size, and population density.

Node features (dynamic): Each node n_i also has a set of dynamic features in the form of a matrix. The dynamic features include the number of active cases, total cases, the current number of hospitalizations, and ICU stays due to COVID-19, which are calculated by aggregating the related procedure codes. We also include the number of each of the 48 COVID-19 related diagnosis codes extracted from claims data according to the Centers for Disease Control and Prevention guideline (https://www.cdc.gov/nchs/data/icd/COVID-19-guidelines-final.pdf). We outline the specific diagnosis codes used in the description of the dataset.

Modeling spatio-temporal patterns using graph attention networks

Obtaining the complex spatial dependencies is a crucial problem to pandemic prediction. By utilizing spatial similarity, our model can make more accurate predictions of a location by considering similar locations' disease transmission status. Here we employ the GAT model[15] to extract spatio-temporal similarity features. The basic idea of GAT is updating the embedding of each node by aggregating its neighboring nodes. In our setting, each location will receive information from its adjacent locations based on mobility to model spatio-temporal disease transmission patterns. This consideration is based on the real-world scenario that adjacent locations may have different impacts on the infectious status of the focused location. For example, if 1 city has a large population size and increasing infected cases, this city may have a considerable impact on its adjacent counties.

We use a 2-layer GAT to extract spatio-temporal features from the attributed graph. We use the latest values from historical data within a sliding window to construct the graph. Mathematically, at time step t, the input features to node i are $\boldsymbol{X}_t^i$, where $\boldsymbol{X}_t^i \in \mathbb{R}^{L_I(F_D+F_S)}$, and L_I denotes the length of the input window. Intuitively at the t-th timestep, we concatenate historical features (ie, L_I days of features) as input. The longer the L_I is, the more historical information and patterns the model will use. Then we apply the graph attention mechanism and calculate the node representation $z_t^i \in \mathbb{R}^{F_z}$ for each node, where F_z denotes the output dimension of the GAT layer.

Concretely, to calculate the z_t^i, we use the multihead mechanism to calculate K independent attention scores following the self-attention strategy.[16] A multihead attention mechanism can help the model get more accurate predictions by generating different attention weights. Intuitively different attention heads may focus on different features in the graph to more comprehensively model the locations. The attention weight of the k-th head between 2 nodes i and j as:

$$e_{ij}^k = \sigma(\mathbf{W}_a^k(\mathbf{W}_z^k\boldsymbol{X}_t^i|\mathbf{W}_z^k\boldsymbol{X}_t^j))$$

where $\mathbf{W}_z^k \in \mathbb{R}^{F_z \times |\boldsymbol{X}_t^i|}$ denotes the linear transformation weight matrix for the k-th head, which will transform the input to the output dimension. $\mathbf{W}_a^k \in \mathbb{R}^{1 \times 2F_z}$ represents the attention computation matrix for the k-th head, and $(\cdot|\cdot)$ denotes the vector concatenation. σ is the nonlinear activation function, and here we use the leaky rectified linear function (LeakyReLU):

$$\sigma(x) = \begin{cases} x, & if\ x \geq 0 \\ 0.01x, & otherwise \end{cases}$$

which is the same as the original GAT model.[15]

Next, we use the softmax function to calculate the attention score:

$$a_{ij}^k = \mathrm{softmax}\left(e_{ij}^k\right) = \frac{\exp\left(e_{ij}^k\right)}{\sum_{n=1}^N \exp\left(e_{in}^k\right)}$$

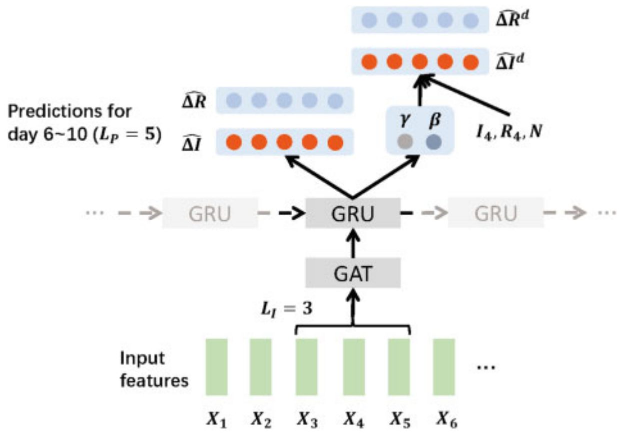

Figure 2. STAN prediction process for a single location at day 5 with $L_I = 3$ and $L_P = 5$.

Each edge of node i will receive an attention score, which assesses how much information should be aggregated from neighboring node j. Finally, we sum up all embedding vectors from multiple heads to obtain the final representation z_t^i for node i as:

$$z_t^i = \frac{1}{K}\sum_{k=1}^{K}\sum_{j=1}^{N} a_{ij}^k \mathbf{W}_z^k \mathbf{X}_t^i$$

where N denotes the number of locations.

Modeling temporal features using recurrent neural networks

Pandemic prediction is not only spatial-related, but also a temporal-related task. The graph information will change along with time. We want to model the spatio-temporal patterns to better predict future trends. We first use the MaxPooling operator, which is to select the maximum value in each column of a matrix to reduce the dimension, to generate embedding for the entire graph as:

$$\tilde{z}_t = \text{MaxPool}([z_t^0, z_t^1, \ldots, z_t^N])$$

where $[z_t^0, z_t^1, \ldots, z_t^N]$ is a matrix and the i-th column is z_t^i, thus $\tilde{z}_t$ incorporates the most important features of all nodes extracted from the graph.

On the same day, the pandemic may have just emerged for some locations, but for other locations, the pandemic may have reached the peak. So, we cannot model temporal patterns for all locations simultaneously using the same model parameters. We build a different model for each location. These locations share the same model structure and graph structure but have different model parameters. The STAN model is an end-to-end model, which means each location's model will adaptively extract most related spatio-temporal patterns from the attributed graph. All following equations are for 1 specific location, and we omit location index i to reduce clutter.

We input the graph embedding to gated recurrent unit (GRU)[17] network to learn temporal features. GRU is a type of recurrent neural network that can effectively model temporal sequences and is also widely used in many sequence analysis tasks.[18,19] The GRU's hidden state is calculated as:

$$\boldsymbol{h}_t = GRU(\tilde{z}_1, \tilde{z}_2, \ldots, \tilde{z}_t)$$

The obtained hidden state of the GRU $\boldsymbol{h}_t$ at the t-th timestep for a specific location contains both spatial and temporal patterns learned from real-world data.

Multitask prediction and transmission dynamics inspired loss function

Our objective is to predict the number of infected cases for both long-term and short-term. In our method, we tackle this issue by using a multitask learning framework to consider short-term and long-term prediction performance jointly.

The idea is to use short-term prediction loss and long-term transmission dynamics constraint loss to regularize learned hidden representations of node embeddings (ie, hidden state of the GRU) $\boldsymbol{h}_t$. In Figure 2, we provide the model prediction process for a single location on day 5 (the input window $L_I = 3$). Since the prediction process is the same for each timestep, we will omit the time index t for simplicity. Concretely, the model output consists of 2 parts:

1. **Transmission/Recovery rate**: The traditional SIR-based model assumes that the disease transmission/recovery rate β and γ remain constant time. But in practice, those rates may easily change over time due to policies or disease evolution reasons. To solve this issue, we define a prediction window L_P that the β and γ will be specific for this window. In Figure 2, $L_P = 5$, and the prediction window starts on Day 6 and ends on Day 10. The model will predict β and γ for the prediction window as:

$$\beta, \gamma = \text{sigmoid}(\text{MLP}(\boldsymbol{h}))$$

where MLP$(\cdot)$ denotes the multi-layer perceptron, and we use sigmoid activation since both β and γ are between 0 and 1.

2. **Daily increased number of infected/recovered cases**: For the i-th prediction window, the model will predict the increment of the number of infected and recovered cases $\widehat{\Delta \boldsymbol{I}}$ and $\widehat{\Delta \boldsymbol{R}}$ as:

$$\widehat{\Delta \boldsymbol{I}}, \widehat{\Delta \boldsymbol{R}} = \text{MLP}(\boldsymbol{h})$$

Note that $\widehat{\Delta \boldsymbol{I}}$ and $\widehat{\Delta \boldsymbol{R}}$ are vectors, since we are predicting for L_P days.

To optimize the model parameters, we design 2 loss terms to make the model predict better for both short-term and long-term progression. Concretely, the loss function also consists of 2 parts:

1. **Transmission dynamics constraint loss.** The first loss term is a transmission dynamics constraint loss to regularize long-term prediction trends. In this term, we do not directly optimize the predictions using the ground truth numbers. Instead, we hope the model can use pandemic dynamics to regularize longer progressions. In particular, we use the predicted transmission/recovery rate β and γ to compute dynamic-based predictions $\widehat{\Delta \boldsymbol{I}}^d$ and $\widehat{\Delta \boldsymbol{R}}^d$.

Then we can optimize the learned β and γ by optimizing these dynamic-based predictions. Concretely, based on the SIR differential equations and given the prediction window in Figure 2, we can calculate the transmission dynamics-based increment number of infected cases and recovered cases iteratively as:

$$\begin{aligned}\widehat{\Delta \boldsymbol{I}}^{d} &= \left[\widehat{\Delta I}^{d}_{t+1},\ \widehat{\Delta I}^{d}_{t+2},\ldots,\widehat{\Delta I}^{d}_{t+L_p}\right],\ \textit{each}\ \widehat{\Delta I}^{d}_{i} = \beta S_{i-1} - \gamma I_{i-1} \\ &= \beta\left(N_P - \widehat{I}^{d}_{i-1} - \widehat{R}^{d}_{i-1}\right) - \gamma \widehat{I}^{d}_{i-1}\end{aligned}$$

$$\widehat{\Delta \boldsymbol{R}}^{d} = \left[\widehat{\Delta R}^{d}_{t+1},\ \widehat{\Delta R}^{d}_{t+2},\ldots,\widehat{\Delta R}^{d}_{t+L_p}\right],\ \textit{each}\ \widehat{\Delta R}^{d}_{i} = \gamma \widehat{I}^{d}_{i-1}$$

where $\widehat{I}^{d}_{i-1}, \widehat{R}^{d}_{i-1}$ can be calculated iteratively using the ground truth value of the infected and recovered cases at the day before the current prediction window (ie, I_4 and R_4 in our example), N_P denotes the population size of the current location, and in Figure 2, $t = 5$ since we are making predictions at the 5th time step. Finally, the transmission dynamics constraint loss is calculated as:

$$\ell_d = \left(\widehat{\Delta \boldsymbol{I}}^{d} - \Delta \boldsymbol{I}\right)^2 + \left(\widehat{\Delta \boldsymbol{R}}^{d} - \Delta \boldsymbol{R}\right)^2$$

where $\boldsymbol{I}_t$ and $\boldsymbol{R}_t$ denote the ground truth number of infected and recovered cases. This loss term calculates the mean squared error of transmission dynamics-based predictions in order to make the prediction results in line with the long-term trend of pandemics.

2. **Prediction loss**. The second loss term is a regular mean squared error loss for the second task:

$$\ell_r = \left(\widehat{\Delta \boldsymbol{I}} - \Delta \boldsymbol{I}\right)^2 + \left(\widehat{\Delta \boldsymbol{R}} - \Delta \boldsymbol{R}\right)^2$$

this loss term is to make the prediction results as close as possible to the short-term variation.

By combining 2 loss terms, the final loss function can be calculated by summing up all locations and timesteps as:

$$\ell = \sum_{i}^{T}\sum_{j}^{N}\left(\ell_r + \ell_d\right)$$

Prediction with STAN

Once the model is trained, we can use the trained model to predict COVID cases at all locations given the length of input window L_I, the length of prediction window L_p, and graph information G as input. Assuming our data are collected between t_1 and t_2: first, the graph information between t_1 and t_2 is fed into GAT to generate the graph embedding; then the graph embeddings are fed into GRU. At the last timestep of the GRU, the model will output predicted daily increased number of infected and recovered cases (ie, $\widehat{\Delta \boldsymbol{I}}, \widehat{\Delta \boldsymbol{R}}$) for future L_p days after t_2; finally, we can calculate the total infected cases number by summing up the increased number. Note that we do not use the transmission dynamics constraints in the prediction time because this module is only used for optimizing the model in training time. We can obtain longer predictions by using a larger prediction window L_P or adding the latest graph data incrementally.

EXPERIMENTS

Dataset description

In this article, we used a US county-level dataset that consists of COVID-19 related data from 2 resources: Johns Hopkins University (JHU) Coronavirus Resource Center[20] and IQVIA's claims data.[21] The data from JHU Coronavirus Resource Center were collected from March 22, 2020 to June 10, 2020. It has the number of active cases, confirmed cases, and deaths related to COVID-19 for different US locations. We select states with more than 1000 confirmed cases by May 17 to ensure the data source accuracy, and finally we have 45 states and 193 counties in the dataset. For those counties, we set the number of cases before their respective first record dates as zero. The IQVIA's claims data are from the IQVIA US9 Database. We export patient claims data and prescription data from March 22, 2020 to June 10, 2020, from which we obtain the number of hospital and ICU visits and the term-frequency of each medical code per county per day. Detailed dataset descriptions are shown in Supplementary Material. The dataset has records for a total of 453 089 patients across the entire time span of the JHU dataset. The 48 unique ICD-10 codes related to COVID-19 are listed in the Supplementary Material.

Baseline models

We compare STAN with the following baselines.

1. **SIR:** the susceptible-infected-removed (SIR), a basic disease transmission model that uses differential equations to simulate an epidemic. S, I, and R represent the number of susceptible, infected, and recovered individuals.
2. **SEIR:** the susceptible-exposed-infected-removed (SEIR) epidemiological model as another transmission dynamics constraint-based baseline. Compared to the SIR model, SEIR adds exposed population size to the equation.
3. **GRU:**[17] We input the latest number of infected cases into a naïve GRU and predict future numbers.
4. **ColaGNN:**[9] ColaGNN uses a location graph to extract spatial relationships for predicting pandemics. Different from STAN, graph nodes in ColaGNN only consist of time series of numbers of infected cases.
5. **CovidGNN:**[10] CovidGNN uses a GNN with skip connections to predict pandemics. They use the graph embedding to predict the future number of cases without using RNN to extract temporal relationships.

To explore the performance enhancement by transmission dynamics constraints and graph structures, we also compare STAN with the following reduced models.

1. **STAN-PC** removes transmission dynamics constraints from STAN.
2. **STAN-Graph** removes the GNN layers and graph data from STAN.

The implementation details of all models are shown in Supplementary Material. We have made our codes available on a public repository (https://github.com/v1xerunt/STAN).

Tasks and evaluation strategy

We predict the future number of active cases on both county level and state level. To evaluate the ability of STAN for both long-term predictions and short-term predictions, we set the prediction window L_P to 5, 15, and 20 (ie, predict for future 5, 15, and 20 days). All training sets start from March 22, and all test sets start from May 17. We also split L_P days from the training sets as evaluation sets to determine model hyperparameters. We set L_I to 5. All locations are used in training and testing set by splitting along the time dimension, where early time windows are used for training the model, and later time windows are used for testing the model. All the models use the same training data and are also evaluated and tested using the same prediction time window.

We use the mean square error (MSE), mean absolute error (MAE) to evaluate our model. We also use the average concordance correlation coefficient (CCC) to evaluate the results. The CCC measures the agreement between 2 variables, and it is computed as:

$$CCC = \frac{2\rho\sigma_x\sigma_y}{\sigma_x^2 + \sigma_y^2 + \left(\mu_x - \mu_y\right)^2}$$

where μ_x and μ_y are the means for the 2 variables, and σ_x^2 and σ_y^2 are the corresponding variances. ρ is the correlation coefficient between the 2 variables. Note that we chose not to use the coefficient of determination (R^2) because the range of R^2 is $(-\infty, 1)$, so some extreme value may significantly affect the average value. But the range of CCC is between -1 and 1, so we can evaluate model results more reliably. To estimate a 95% confidence interval, according to previous pandemic research,[22] we resample the locations 1000 times, calculate the score on the resampled sets, and then use 2.5 and 97.5 percentiles of these scores as our confidence interval estimate.

RESULTS

Table 1 shows the average performance and a 95% confidence interval for state-level predictions of our model and all baseline models. STAN achieves the best performance under different lengths of the prediction window. When the length of the prediction window $L_P = 5$, STAN achieves 59% lower MSE, 33% lower MAE, and 23% higher CCC than the best baseline ColaGNN. When the length of the prediction window $L_P = 15$, STAN achieves 87% lower MSE, 56% lower MAE, and 47% higher CCC than ColaGNN. When the length of the prediction window $L_P = 20$, STAN achieves 48% lower MSE, 37% lower MAE, and 32% higher CCC than ColaGNN.

Table 2 shows the performance for county-level prediction results. STAN also achieves the best performance under different lengths of the prediction window. When the length of the prediction window $L_P = 5$, STAN acquires 26% lower MSE, 29% lower MAE, and 25% higher CCC than ColaGNN. When the length of the prediction window $L_P = 15$, STAN achieves 55% lower MSE, 34% lower MAE, and 30% higher CCC than ColaGNN. When the length of the prediction window $L_P = 20$, STAN achieves 55% lower MSE, 37% lower MAE, and 29% higher CCC than ColaGNN.

The results show STAN can conduct more accurate long-term and short-term prediction than SIR and SEIR models on both state and county levels. Since county-level graph data are more granular, STAN can benefit more by utilizing such data than the traditional dynamics-based model. It is also worth noting that both reduced model STAN-PC and STAN-Graph also outperform other baselines. This indicates that both transmission dynamics constraints and real-world evidence provide valuable information for pandemic progression prediction. We reported the detailed performance of each location in the Supplementary Material. We conducted a T-test between STAN and each baseline model to check the performance difference statistically. The P value is shown in Table 3. The results show that for each baseline model, STAN can significantly outperform statistically (P value $< .001$). The detailed t-test results are shown in the Supplementary Material.

DISCUSSION AND LIMITATIONS

In this section, we will discuss the advantages and also the limitations of our model. We draw the predicted curve of 20 days from May 16 to Jun 5 for 2 counties, El Paso, TX and Lake, IN, and 2 states, CA and MA. As shown in Table 4, for the 2 counties, STAN shows up to 99% relatively lower MSE compared to the SEIR and SIR model. For the 2 states, the performance improvement is much greater, STAN can achieve at most 95% lower MSE compared to the best SEIR models. And as shown in Figures 3 and, 4 the curve also fits the actual trend better for both counties and states. One obvious drawback of SIR and SEIR models is the overfitting issue. The SIR and SEIR models tend to predict the peak will come right after current data, which is especially apparent in the prediction curve of Lake, IN, and MA. This is because these traditional models do not incorporate the influence and interdependency of transmission between geographic regions. The characteristics of transmission of infectious diseases in 1 area are unlikely to be decoupled from those of nearby areas unless there are barriers to interaction between the regions such as topography (rivers with limited bridges or mountain ranges with limited road connections) or controlled borders. Such decoupling is infrequently present between counties in the US. The inability to account for this geographic interdependency removes an important variable in the SIR and SEIR models and impedes their ability to predict the future progression using limited data at the early pandemic stage.

Though deep learning-based methods can achieve better performance than traditional statistical methods in various time series analysis and prediction tasks, there are still 2 major limitations in our work. The first limitation is the prediction window setting in our method. The traditional SIR and SEIR models use all historical training data to fit the model and generate the entire curve to be less affected by the fluctuations in the data. However, our model divides historical data into prediction windows in training time. This setting allows STAN to dynamically model the pandemic progressions. Therefore, it can better simulate situations, such as changes in reporting policy or dynamical changes in transmissibility or contagion. However, if the number of cases fluctuates drastically due to inaccuracy in the data collection process, it is difficult for the STAN model to learn valid and stable transmission and recovery rates. This issue can be further solved by applying dynamic data smoothing to smooth such abnormal data points.

The second limitation is that the transmission dynamics constraints may be too simple to reflect real-world situations, such as home isolation and pandemic control policies. A lot of research focuses on improving the traditional SIR model by adding more population groups and transmission equations.[3,23] These variants can be quickly adopted in our model by modifying the transmission dynamic loss.

The third limitation is about data quality for constructing the attribute graph. For node features, indeed, sometimes ICD codes may not reflect real pandemic status due to report delay or other reasons. And the edge mobility calculation can also be improved by incorporating more data sources such as traffic info or mobile geolocation tracking. Besides, the dynamic features and static features are processed together without considering their different characteristics. Future work can incorporate more data sources and extend the transmission dynamics constraints into STAN to further enhance the prediction performance with richer data and process different data types more reasonably. Our model can also be easily adopted for mortality or hospitalization prediction tasks by adding related statistics such as COVID-19 related treatment codes and ICU codes to the data.

Table 1. Performance comparison for state-level predictions

Prediction window $L_P = 5$			
Model	**MSE**	**MAE**	**CCC**
SIR	2 968 711 (814 014–4 152 617)	921.06 (776.93–1209.22)	0.41 (0.37–0.45)
SEIR	1 890 708 (612 049–3 562 890)	679.64 (681.57–1197.38)	0.49 (0.44–0.54)
GRU	925 701 (501 309–1 792 855)	582.43 (479.50–842.38)	0.55 (0.50–0.60)
ColaGNN	601 840 (381 907–982 354)	440.26 (323.57–568.44)	0.66 (0.59–0.72)
CovidGNN	830 517 (430 127–1 109 311)	500.11 (367.55–645.72)	0.58 (0.53–0.64)
STAN-PC	323 325 (213 702–450 314)	313.72 (280.39–392.01)	0.75 (0.70–0.79)
STAN-Graph	472 245 (276 391–612 099)	362.04 (310.08–452.39)	0.67 (0.63–0.71)
STAN	**237 412** **(159 995–290 801)**	**220.50** **(172.71–272.03)**	**0.84** **(0.81–0.87)**
Prediction window $L_P = 15$			
Model	**MSE**	**MAE**	**CCC**
SIR	22 939 910 (11 682 896–35 393 542)	2438.95 (1807.71–3119.30)	0.32 (0.25–0.40)
SEIR	12 993 900 (5 234 542–21 928 077)	1781.66 (1305.68–2295.44)	0.49 (0.44–0.55)
GRU	9 205 382 (3 708 352–15 534 698)	1710.09 (1253.23–2203.22)	0.38 (0.34–0.43)
ColaGNN	7 192 031 (2 897 281–12 137 035)	1290.41 (945.67–1662.52)	0.57 (0.51–0.63)
CovidGNN	9 609 283 (3 871 062–16 216 309)	1611.19 (1180.75–1075.80)	0.45 (0.40–0.50)
STAN-PC	1 785 304 (1 032 754–2 895 702)	774.22 (650.39–904.74)	0.72 (0.68–0.76)
STAN-Graph	2 897 053 (1 352 076–4 309 806)	964.09 (784.01–1011.36)	0.66 (0.60–0.71)
STAN	**972 192** **(622 425–1 404 284)**	**586.56** **(484.11–690.61)**	**0.84** **(0.80–0.87)**
Prediction window $L_P = 20$			
Model	**MSE**	**MAE**	**CCC**
SIR	46 732 397 (23 701 239–71 720 863)	3439.25 (2538.28–4423.32)	0.25 (0.18–0.33)
SEIR	25 296 100 (9 038 485–44 609 676)	2451.60 (1849.00–3120.60)	0.43 (0.37–0.50)
GRU	15 901 430 (5 681 699–28 042 173)	2046.32 (1543.34–2604.72)	0.52 (0.44–0.60)
ColaGNN	9 317 132 (3 971 773–19 667 264)	1645.42 (1240.98–2094.72)	0.63 (0.54–0.72)
CovidGNN	16 739 642 (6 623 891–27 756 861)	2 081.25 (1569.69–2649.19)	0.54 (0.46–0.62)
STAN-PC	5 929 321 (3 082 515–10 309 710)	1209.41 (1032.75–1564.71)	0.75 (0.72–0.78)
STAN-Graph	9 509 671 (5 909 301–15 408 623)	1689.90 (1342.09–2031.74)	0.68 (0.64–0.72)
STAN	**4 909 604** **(1 999 607–8 811 535)**	**1088.48** **(820.78–1366.01)**	**0.82** **(0.79–0.86)**

Abbreviations: CCC, concordance correlation coefficient; GNN, graph neural network, GRU, gated recurrent unit; MAE, mean absolute error; MSE, mean square error; SEIR, susceptible-exposed-infectious-recovered SIR, susceptible-infectious-recovered; STAN, spatio-temporal attention network.

CONCLUSION

In this work, we propose a spatio-temporal attention network model (STAN) for the COVID-19 pandemic prediction. We map locations (eg, a county or a state) to nodes on a graph. We use a set of static and dynamic features extracted from multiple real-world evidence, including real-world medical claims data, to construct nodes and use geographical proximity and demographic similarity between locations to construct edges. We use a GAT to incorporate the variant influence of the different neighboring locations of a node and

Table 2. Performance comparison for county-level predictions

Prediction window $L_P = 5$			
Model	**MSE**	**MAE**	**CCC**
SIR	93 512	151.33	0.40
	(44 864–159 117)	(125.49–177.86)	(0.38–0.44)
SEIR	134 494	165.14	0.35
	(50 223–251 893)	(136.94–194.09)	(0.32–0.38)
GRU	79 982	121.76	0.47
	(39 820–136 096)	(100.96–143.10)	(0.43–0.51)
ColaGNN	61 627	110.91	0.53
	(36 176–104 864)	(91.97–130.36)	(0.49–0.58)
CovidGNN	71 664	120.01	0.47
	(37 718–121 941)	(99.16–140.55)	(0.43–0.51)
STAN–PC	53 194	107.69	0.58
	(32 961–103 211)	(87.63–123.12)	(0.53–0.63)
STAN-Graph	50 331	104.99	0.57
	(29 023–97 304)	(85.09–117.33)	(0.53–0.61)
STAN	**44 177**	**79.80**	**0.66**
	(13 028–79 916)	**(66.17–93.79)**	**(0.60-0.71)**
Prediction window $L_P = 15$			
Model	**MSE**	**MAE**	**CCC**
SIR	884 249	415.79	0.29
	(438 613–1 353 544)	(353.08–480.88)	(0.26–0.32)
SEIR	1 102 601	391.06	0.33
	(495 229–2 014 880)	(318.76–469.78)	(0.30–0.36)
GRU	810 362	322.90	0.48
	(382 092–1 008 423)	(291.54–397.06)	(0.45–0.51)
ColaGNN	465 104	286.55	0.56
	(290 623–878 780)	(258.72–352.37)	(0.54–0.59)
CovidGNN	635 401	310.84	0.50
	(342 312–935 801)	(280.64–382.23)	(0.47–0.53)
STAN–PC	393 790	246.01	0.65
	(251 323–675 304)	(215.42–305.33)	(0.63–0.68)
STAN-Graph	339 082	242.74	0.66
	(210 403–612 392)	(219.02–298.37)	(0.63–0.69)
STAN	**157 243**	**193.85**	**0.72**
	(100 712–218 922)	**(170.52–220.06)**	**(0.69–0.74)**
Prediction window $L_P = 20$			
Model	**MSE**	**MAE**	**CCC**
SIR	1 881 144	585.14	0.23
	(511 054–2 953 878)	(497.63–682.66)	(0.20–0.26)
SEIR	2 238 468	538.57	0.29
	(647 839–3 998 162)	(437.36–644.32)	(0.26–0.32)
GRU	981 064	461.86	0.46
	(419 891–1 294 611)	(426.74–585.42)	(0.43–0.49)
ColaGNN	703 377	410.13	0.55
	(301 042–928 175)	(378.95–519.86)	(0.52–0.58)
CovidGNN	1 043 261	468.44	0.43
	(446 511–1 376 685)	(432.83–593.77)	(0.40–0.47)
STAN-PC	492 374	271.63	0.67
	(287 672–853 034)	(233.17–325.09)	(0.64–0.70)
STAN-Graph	555 681	281.62	0.68
	(300 626–879 030)	(239.01–356.71)	(0.64–0.71)
STAN	**326 258**	**253.86**	**0.71**
	(187 532–505 796)	**(219.53–291.86)**	**(0.69–0.74)**

Abbreviations: CCC, concordance correlation coefficient; GNN, graph neural network, GRU, gated recurrent unit; MAE, mean absolute error; MSE, mean square error; SEIR, susceptible-exposed-infectious-recovered SIR, susceptible-infectious-recovered; STAN, spatio-temporal attention network.

predict the number of infected patients for a fixed period into the future. We also impose transmission dynamics constraints on predictions according to transmission dynamics. STAN achieves better prediction performance than either of the traditional SIR and SEIR models and other deep learning methods and shows less overfitting issues at the early stage of the pandemic. We hope our model can help governments and researchers better allocate medical resources and make policies to control the pandemic earlier. Our model can also be easily extended to predict hospitalization of COVID-19 as future work.

Table 3. T-Test for STAN and other baseline models (prediction window in 5–20 days)

Model	State-5	State-15	State-20	County-5	County-15	County-20
SIR	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
SEIR	0.00E+00	0.00E+00	4.04E-263	1.46E-203	6.21E-253	9.74E-257
GRU	5.27E-16	3.46E-12	2.70E-08	8.46E-18	3.57E-84	5.28E-56
ColaGNN	9.54E-09	4.08E-20	1.03E-04	2.83E-07	5.23E-60	9.85E-57
CovidGNN	2.02E-12	4.38E-24	5.27E-08	3.07E-20	4.32E-69	2.57E-61
STAN-PC	3.84E-04	6.23E-06	3.95E-03	1.98E-06	3.99E-32	7.21E-21
STAN-Graph	5.34E-07	6.62E-09	7.16E-05	4.76E-04	6.31E-16	4.55E-12

Abbreviations: GNN, graph neural network, GRU, gated recurrent unit; SEIR, susceptible-exposed-infectious-recovered; SIR, susceptible-infectious-recovered; STAN, spatio-temporal attention network.

Table 4. Prediction performance for 2 counties (El Paso, TX and Lake, IN) and 2 states (CA and MA)

Model	MSE	MAE	CCC
El Paso, TX			
SIR	867 272	922.38	0.27
SEIR	196 954	403.36	0.47
STAN	3889	47.14	0.99
Lake, IN			
SIR	448 839	619.16	0.06
SEIR	182 627	361.95	0.19
STAN	842	24.67	0.99
CA			
SIR	122 495 498	8498.43	0.45
SEIR	58 379 735	5848.03	0.79
STAN	2 782 908	1306.15	0.99
MA			
SIR	592 535 941	21 652.34	−0.11
SEIR	28 495 989	4458.38	0.37
STAN	2 771 603	1461.30	0.93

Abbreviations: CCC, concordance correlation coefficient; MAE, mean absolute error; MSE, mean square error; SEIR, susceptible-exposed-infectious-recovered SIR, susceptible-infectious-recovered; STAN, spatio-temporal attention network.

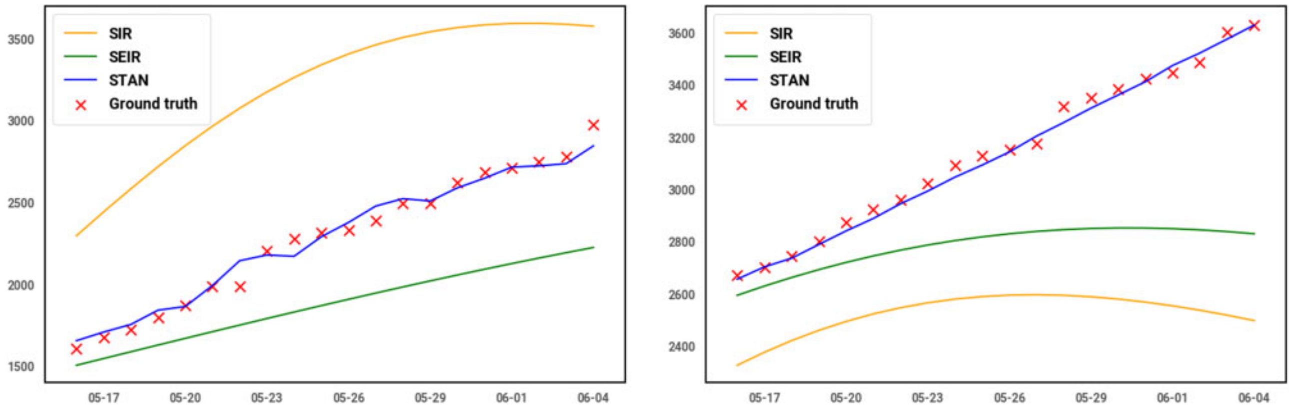

Figure 3. Predicted curve for 2 counties: El Paso, TX (left) and Lake, IN (right).

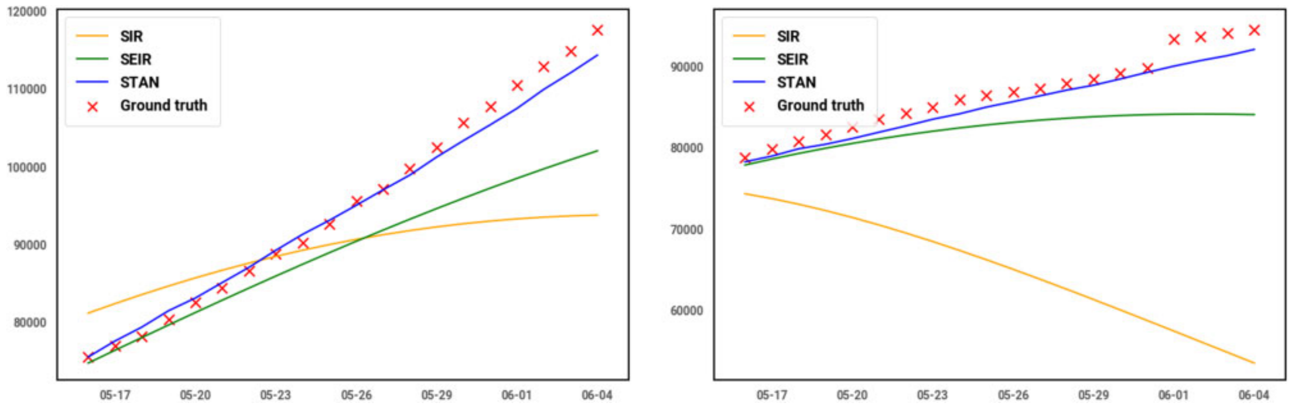

Figure 4. Predicted curve for 2 states: CA (left) and MA (right).

FUNDING

This work was in part supported by the National Science Foundation award SCH-2014438, IIS-1418511, CCF-1533768, IIS-1838042 and the National Institute of Health award NIH R01 1R01NS107291-01 and R56HL138415.

AUTHOR CONTRIBUTIONS

JG and RS implemented the method and conducted the experiments. All authors were involved in developing the ideas and writing the paper.

DATA AVAILABILITY

The COVID-19 statistics data used in this article are provided by Johns Hopkins University (JHU) Coronavirus Resource Center,[20] and they are publicly available at https://github.com/CSSEGISandData/COVID-19. The claims data used in this article are provided by IQVIA,[21] and they will be shared on request to https://www.iqvia.com/solutions/real-world-evidence/real-world-data-and-insights.

CONFLICT OF INTEREST STATEMENT

None declared.

SUPPLEMENTARY MATERIAL

Supplementary material is available at *Journal of the American Medical Informatics Association* online.

REFERENCES

1. Wikipedia. COVID-19 pandemic data. Secondary COVID-19 pandemic data 2020. https://en.wikipedia.org/wiki/Template: COVID-19_pandemic_data Accessed July 2, 2020.
2. Pei S, Shaman J. Initial Simulation of SARS-CoV2 Spread and Intervention Effects in the Continental US. *medRxiv* 2020. 03.21.20040303; doi: 10.1101/2020.03.21.20040303
3. Yang Z, Zeng Z, Wang K, *et al.* Modified SEIR and AI prediction of the epidemics trend of COVID-19 in China under public health interventions. *J Thorac Dis* 2020; 12 (3): 165–74.
4. Roberts M, Andreasen V, Lloyd A, Pellis L. Nine challenges for deterministic epidemic models. *Epidemics* 2015; 10: 49–53.
5. Durbin J, Koopman SJ. *Time Series Analysis by State Space Methods*. Oxford, UK: Oxford University Press; 2012.
6. Wang L, Chen J, Marathe M. DEFSI: deep learning based epidemic forecasting with synthetic information. In: *Proceedings of the AAAI Conference on Artificial Intelligence*; February 1, 2019; Honolulu, Hawaii.
7. Qian-Li M, Qi-Lun Z, Hong P, Tan-Wei Z, Jiang-Wei Q. Multi-step-prediction of chaotic time series based on co-evolutionary recurrent neural network. *Chinese Phys B* 2008; 17 (2): 536–42.
8. Du B, Xu W, Song B, Ding Q, Chu S-C. Prediction of chaotic time series of rbf neural network based on particle swarm optimization. In: *Intelligent Data Analysis and Its Applications, Volume II*. Switzerland: Springer; 2014: 489–97.
9. Deng S, Wang S, Rangwala H, Wang L, Ning Y. Graph message passing with cross-location attentions for long-term ILI prediction. *arXiv preprint arXiv: 1912.10202*; 2019.
10. Kapoor A, Ben X, Liu L, *et al.* Examining COVID-19 Forecasting using Spatio-Temporal Graph Neural Networks. *arXiv preprint arXiv: 2007.03113*; 2020.
11. Wu J-L, Kashinath K, Albert A, Chirila D, Xiao H. Enforcing statistical constraints in generative adversarial networks for modeling chaotic dynamical systems. *J Comput Phys* 2020; 406: 109209.
12. Beucler T, Pritchard M, Rasp S, Gentine P, Ott J, Baldi P. Enforcing analytic constraints in neural-networks emulating physical systems. *arXiv preprint arXiv: 1909.00912*; 2019.
13. Seo S, Liu Y. Differentiable physics-informed graph networks. *arXiv preprint arXiv: 1902.02950* 2019.
14. Seo S, Meng C, Liu Y. Physics-aware difference graph networks for sparsely-observed dynamics. In: proceedings of the *International Conference on Learning Representations*; April 26, 2020; virtue conference.
15. Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y. Graph Attention Networks. *International Conference on Learning Representations*; April 30, 2018; Vancouver, Canada.
16. Vaswani A, Shazeer N, Parmar N, *et al.* Attention is all you need. In: *Advances in Neural Information Processing Systems*; December 4, 2017; Long Beach, CA.
17. Chung J, Gulcehre C, Cho K, Bengio Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. In: *NIPS 2014 Workshop on Deep Learning;* December 2014; Montreal, Quebec, Canada.
18. Tokgöz A, Ünal GA. RNN based time series approach for forecasting turkish electricity load. In: *2018 26th Signal Processing and Communications Applications Conference (SIU)*. Izmir, Turkey: IEEE.
19. Dai G, Ma C, Xu X. Short-term traffic flow prediction method for urban road sections based on space–time analysis and GRU. *IEEE Access* 2019; 7: 143025–35.

20. Dong E, Du H, Gardner L. An interactive web-based dashboard to track COVID-19 in real time. *Lancet Infect Dis* 2020; 20 (5): 533–4.
21. IQVIA. Real World Data and Insights. Secondary Real World Data and Insights 2020. https://www.iqvia.com/solutions/real-world-evidence/real-world-data-and-insights Accessed July 2, 2020.
22. Tessmer HL, Ito K, Omori R. Can machines learn respiratory virus epidemiology?: A comparative study of likelihood-free methods for the estimation of epidemiological dynamics. *Front Microbiol* 2018; 9: 343.
23. Li MY, Smith HL, Wang L. Global dynamics of an SEIR epidemic model with vertical transmission. *SIAM J Appl Math* 2001; 62 (1): 58–69.