Into the Unobservables: A Multi-range Encoder-decoder Framework for COVID-19 Prediction

Yue Cui, Chen Zhu, Guanyu Ye, Ziwei Wang, and Kai Zheng*
University of Electronic Science and Technology of China, China
{cuiyue,zhengkai}@uestc.edu.cn,{chenzhu,ygy,ziwei}@std.uestc.edu.cn

ABSTRACT

The ongoing COVID-19 pandemic has dramatically changed people's daily lives. A robust forecasting model for COVID-19 infections is essential for governments and institutions to plan timely and perform accurate interventions. Mainstream solutions for COVID-19 prediction fit reported data only by considering observed cases. However, the neglected facts that positive samples are incomplete and many facts of the novel disease are unknown may be prone to cause severe error accumulation, especially in long-term predictions. To fully understand the spreading patterns of the virus, we propose an encoder-decoder framework: (i) in the encoder we embed historical case data into multiple expose-infection ranges and learn message passing between time slices and across ranges with coarse-grained human mobility data incorporated; (ii) in the decoder, we decode the embedded features based on reported cases as well as deaths to jointly consider the effect of both observed and hidden data. We model the spreading of disease in over 60 counties of California and New York, which are two of the most metropolitan areas in the US. The proposed framework significantly outperforms state-of-the-art baselines on JHU COVID-19 dataset on both weekly prediction and daily prediction tasks. We design detailed ablation studies to verify the effectiveness of each key module and find the model not only works with the assistance of mobility data but also with purely cases and deaths, which implies its broad application scenarios.

CCS CONCEPTS

• **Applied computing** → **Forecasting**; • **Information systems** → *Spatial-temporal systems.*

KEYWORDS

datasets, neural networks, gaze detection, text tagging

ACM Reference Format:
Yue Cui, Chen Zhu, Guanyu Ye, Ziwei Wang, and Kai Zheng. 2021. Into the Unobservables: A Multi-range Encoder-decoder Framework for COVID-19 Prediction. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, QLD, Australia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3459637.3482356

*Kai Zheng is the corresponding author.

CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia

ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3482356

1 INTRODUCTION

Fifteen months after it was declared as a pandemic by World Health Organization (WHO), by May 1st, 2021, the ongoing Coronavirus Disease 2019 (COVID-19) had claimed the lives of more than 3.2 million people worldwide and more than 150 million had been reported as infected [1], making it one of the most serious public health events in human history. The pandemic has not only had a great influence on individual's daily life but also disrupted the economy and public relations of countries.

To prevent the spread of the disease, many countries enacted intervention strategies such as lock-down, stay-at-home order, and mandatory mask-wearing rules in public places. Effective these measures are, it is also important for governments and organizations to manipulate the strictness. A loose control that underestimates the region's infection risk might lead to the disastrous consequence of exposing a massive population to the virus. On the other hand, since the economic and social cost of such measures is high, overreaction at low-risk areas can also be problematic. For example, the closure of nonessential businesses can increase the unemployment rate and social anxiety. Therefore, to achieve the goal of smart policy making, it is of necessity to precisely understand the COVID-19 situation of a region and forecast the future trend.

The task of COVID-19 prediction is the most similar in spirit to time series forecasting. However, compared with typical time series forecasting settings, such as traffic prediction (e.g., [23, 29]) and stoke prize prediction (e.g., [2]), COVID-19 prediction poses nontrivial challenges of: 1) **Unknown and complicated epidemiological patterns**. As a novel disease, many of the epidemiological characteristics of the COVID-19 pandemic have yet to be fully observed and quantified. It is not sufficiently understood that, for example, how infectious one patient is when she is at the asymptomatic phase, how do environment and the susceptible population's background affect infection, and what is the best dosage of vaccine and re-vaccination interval. Moreover, a common belief is that the incubation period plays an important role in understanding and controlling a pandemic, which is defined as the time between exposure to the virus and symptom onset. But according to [13], the incubation period of COVID-19 varies significantly, which has a median of approximately 5 days but can be as long as 22 days. To model the development of the pandemic, it is desirable to take these uncertainties into consideration. 2) **Missing data**. Though the testing capacity of COVID-19 has been scaling up remarkably since the outbreak, limited by the time-sensitive nature of the test, there are always unobserved missing reports. It is concerning that

[1] based on the data collected by John Hopkins University https://coronavirus.jhu.edu/

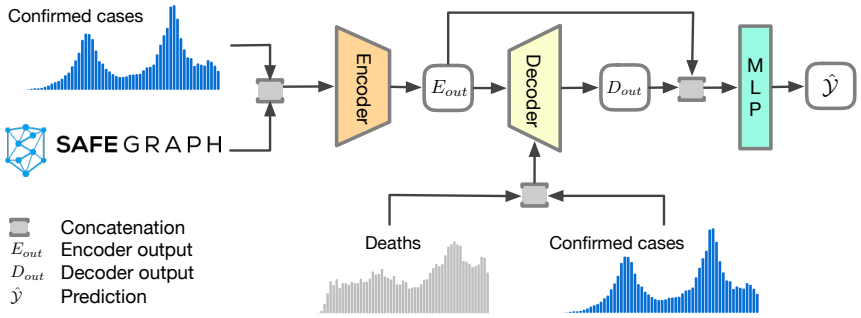


Figure 1: Pipeline of the proposed method.

these SARS-CoV-2 carriers may interact freely and unintentionally with their family and the community. Monitoring these "invisible" cases can be beneficial in making precise forecasting. 3) **Invasion of privacy.** Sharing and processing private data can be helpful in preventing the spread of the disease. For example, personal trajectories are often required for accurate epidemiological investigation, tracking, and transmission prevention. However, it could seriously compromise people's privacy. Thus, it is of importance to balance public interest and personal privacy.

To address these challenges, in this paper, we propose an approach for COVID-19 cases prediction based on an encoder-decoder framework. The pipeline of the proposed method is illustrated in Figure 1. The basic rationale behind our approach is that we implicitly model the spreading process of disease in a graph structure (the encoder) and fit observed cases with both missing cases and observed cases considered (the decoder). For components of the encoder, we first introduce the multi-range embedding layer, which takes reported COVID-19 cases and the human mobility data collected as census block group (CBG) visitor counts as time series and performs convolution in the time axis to obtain temporal embeddings with respect to multiple exposure-infection ranges. Based on the assumption that infection happens when the infected interact with the susceptible, in the graph-based within-range exposure-infection layer, we model the potential dependencies of CBG visitors and COVID-19 confirmed cases by taking embedded features as nodes of graphs and learning the message passing process between nodes. A cross-range fusion layer based on the attention mechanism is then applied to fuse information across embedding ranges. We argue that observed cases are inaccurate and purely using such incomplete data might downgrade the prediction performance. The proposed decoder receives COVID-19 cases and deaths as inputs, which stand for explicit observations and implicit factors. We use a prediction horizon-aware convolution operation to obtain temporal embeddings and then implement multi-head attention to decode outputs of the encoder. The final prediction is then made by linear transforming of the decoder outputs through multi-layer perceptions (MLPs).

The contributions of this paper are summarized as follows:

- We propose a novel encoder-decoder based approach to successfully enhance the prediction accuracy of COVID-19. To the best of our knowledge, this is the first study modeling COVID-19 spreading with graph neural networks and considering both observed and hidden cases.
- We present a propagation and fusion mechanism as the encoder with regard to multiple exposure-infection ranges to capture COVID-19's various incubation periods.
- We propose a decoder that is fed with confirmed cases and deaths data to jointly model the effect of observed cases and missing reports.
- Effectiveness of the proposed approach is analyzed and confirmed through extensive experiments on JHU US COVID-19 dataset. We also conduct comprehensive ablation studies and parameter analysis to verify the effectiveness of each key component.

2 RELATED WORK

Many efforts have been devoted by recent works to analyzing and forecasting COVID-19 statistics. According to the method used, existing models for COVID-19 cases prediction can be categorized into several groups. (1) *SEIR based* methods make predictions by modeling the dynamic transition of the susceptible, exposed, infectious and removed population [11, 15, 17, 20, 28]. Typical focus of these works is on the estimation of key parameters of the epidemiology SEIR model [11] or modifying the basic SEIR model to adapt to the novel disease [28]. (2) *Time-series based* methods take the COVID-19 data as general time series and learn to make predictions with sequential models, such as with hidden Markov models (HMM) [16], Bayesian hierarchical model (BHM) [4, 28, 31] and recurrent neural networks (RNNs) [19, 28]. (3) *Ensemble* methods hybrid prediction of different models [6, 12, 14]. For example, [14] is an ensemble of three different models: an auto-regressive model, long short term memory (LSTM) model and a SEIR model.

The above works mainly focus on how to make better predictions with only historical cases (or also with demographic characteristics). Although there exist some techniques to improve prediction accuracy by considering human mobility, which plays an important role in the spreading of COVID-19, they are not widely adopted because of complex design or narrow task settings. [7] proposes a mobility data-driven approach to estimate the epidemiological parameters of SEIR. Similarly, [5] achieves the same goal by combining epidemiological models but requires fine-coursed mobility networks that model transmission rates between locations and the locations' epidemiological data. [1, 10, 22] prove that those constraints on individual movements and social interactions are effective in controlling the spread of the disease. [26] constructs features from mobility data and then uses a cross-city transfer learning module

to detect high-risk neighborhoods. Besides, practitioners still use fixed epidemiological parameters and fail to take into account the missed data in reported cases.

3 PRELIMINARY

Before describing the proposed method, we first introduce some preliminary concepts and state the problem.

Definition 3.1. **JHU CSSE COVID-19 data**. We use the daily reports containing case and death data from the JHU CSSE group [2] as golden-standard COVID-19 data. For the task of weekly prediction, we sum over the daily counts by week to produce weekly counts. The case data is denoted as c and the death data demoted is as f.

Rather than modeling detailed human mobility patterns as proposed in [5], which estimates the number of individuals from CBG i to POI p at the t-th hour, we sorely use CBG visitor counts data provided by SafeGraph, which is defined as follows.

Definition 3.2. **CBGs visitor counts**. A census block group (CBG) is a geographical unit that typically has a population of 600 to 3,000 people [3]. Visitor counts of the i-th CBG, denoted by v_i, measures the daily number of visits to points of interest (POIs) that locate in the CBG.

Definition 3.3. **COVID-19 case prediction**. For a county j, at time t, given a L_x-length observed historical COVID-19 cases and CBG visitor counts, i.e., $\mathcal{X}^j(t) = [X_1^j(t), ..., X_{L_x}^j(t)] \in \mathbb{R}^{L_x \times d_x^j}$, where $X_i^j(t) = [c_i^j(t), v_{i,1}^j(t), ..., v_{i,d_x-1}^j(t)] \in \mathbb{R}^{d_x^j}, i \in \{1, ..., L_x\}$, the historical deaths $\mathcal{F}(t)^j = [f_1^j(t), ..., f_{L_x}^j(t)] \in \mathbb{R}^{L_x}$, the goal of COVID-19 case prediction is to predict the corresponds cases Δ steps ahead, i.e., $\mathcal{Y}^j(t) = c_{L_x+\Delta(t)}^j$.

4 METHODOLOGY

In this section, we detail two major components of the proposed method, the encoder for feature embedding and the decoder for prediction. Note that the following descriptions can be applied to any county and any input sequence we model. For simplicity, we omit the county identifier j and input sequence identifier t.

4.1 Encoder

As illustrated in Figure 2, the encoder has three major components: a temporal embedding layer to generate embeddings of different ranges, a graph-based within-range expose-infection (GRE) module to implicitly model the spreading process of SARS-CoV-2 in the crowd, and a cross-range fusion layer to aggregate information across ranges.

4.1.1 Temporal Embedding. The COVID-19 incubation period various widely from case to case, which has a median of 5 days and can be up to 22 days [13]. To model different durations of getting exposed to SARS-CoV-2 to being tested positive, we first use several multi-channel CNNs to obtain temporal features of the input time series w.r.t. multiple exposure-infection ranges. More specifically, given the input time series, i.e., $\mathcal{X} = [X_1, ..., X_{L_x}]$ we apply 1dConv operation along the time axis of the input with K different kernels. For the $i-th$ kernel, $i \in \{1, ..., K\}$, extracted temporal feature can be described as:

$$\mathcal{H}_i^0 = \sigma(W_i^0 * \mathcal{X} + b_i^0), \tag{1}$$

where $*$ is the convolution operation, W_i^0 is the weight of the i-th kernel, $\mathcal{H}_i^0 = [H_{i,1}^0, ..., H_{i,\widetilde{L}_x^i}^0] \in \mathbb{R}^{C \times \widetilde{L}_x^i \times d_x}$, C is the channels of convolution, $\widetilde{L}_x^i$ denotes the post-convoluted length of the sequence and 0 means this operation is implemented at the 0-th layer. The definition of layer will be describe in detail in subsequent part of the this section.

Temporal features of different exposure-infection ranges are informative. The obtained feature from the kernel of size k_i represents an observation of k_i-length window. For each observation window, i.e., a specific time slice from the $\widetilde{L}_x^i$-length sequence, there can be interaction and transmission between dimensions (CBG visitors and confirmed cases). For example, a susceptible interacts with an infected and then catches SARS-CoV-2. Moreover, considering that the incubation period of the disease is uncertain, people who are infected might be recognized as a case outside the observation window. Therefore, we can then make use of the obtained feature embeddings to capture dynamic transitions between groups.

4.1.2 Graph-based Within-range Exposure and Infection. Graph as a ubiquitous data structure has proven to be powerful in representing relationships between points and its generality [9]. There have been works that construct and utilize the graph structures of time series and make predictions [24, 27]. Inspired by these works, we propose a graph-based within-range exposure-infection (GRE) module to model the latent interactions between visitors in CBGs and infected people in graph formalism.

For each specific time slice in $\widetilde{L}_x^i$, we take the case and CBG-related features as nodes and the potential dependency between nodes as edges. There are two obstacles in modeling the message passing between nodes. First, relationships between the nodes are unknown. Second, it can be computationally expensive to model two kinds of message, i.e., message that passes between nodes in a specific time slice and across time slices, aka spatial information and sequential information.

We deal with the first challenge by learning graph structures from data. Self-attention [21] is performed over all nodes to compute the attention values, which are taken as entries of the adaptive adjacency matrix. As for the second challenge, it is an intuitive and typical way to separately model the two kinds of message passing [8, 29]. For example, using convolutional models (e.g., GCN, CNN) to propagate spatial information and then sequential models to capture temporal dependencies (e.g. LSTM, GRU) [24, 29]. However, due to the inherent sequential nature of RNN-based models, parallelization is disabled, and processing the time series in a serial manner can be extremely time-consuming as the accumulated time input sequence get longer. We here propose an effective and efficient way to jointly propagate the two kinds of message.

Consider the input sequence as a whole, then there are in total $d_x\widetilde{L}_x^i$ nodes. The extracted feature vectors ($\in \mathbb{R}^C$) can be used as embeddings of the nodes. The adjacency matrix $A^l \in \mathbb{R}^{d_x\widetilde{L}_x^i \times d_x\widetilde{L}_x^i}$ is calculated as:

[2] https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data
[3] https://en.wikipedia.org/wiki/Census_block_group

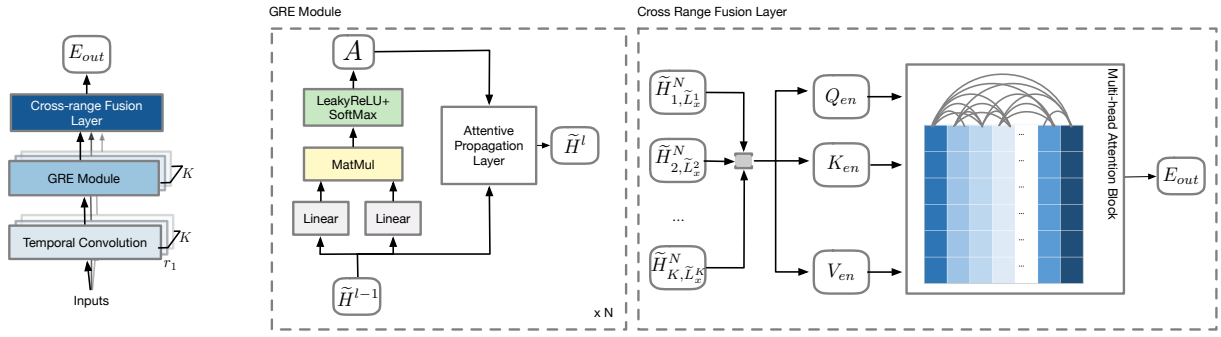


Figure 2: An overview of the encoder.

$$A^l = Softmax(LeakyReLU(\frac{\widetilde{H}^{l-1}W_Q^l(\widetilde{H}^{l-1}W_K^l)^T}{\sqrt{d_K}})), \quad (2)$$

where W_Q and $W_K \in \mathbb{R}^{C\times d_K}$ are parameter matrices d_K is the dimension of the self-attention layer, $\widetilde{H}^{l-1} \in \mathbb{R}^{\widetilde{L}_x^i d_x \times C}$, $l \geq 1$ is the 2D-reshaped version of $\mathcal{H}_i^{l-1}$, which is the embedding of nodes obtained from the previous GRE layer. Rather using the ReLU function used in [25], we adopt LeakyReLU as activation function to eliminate weak connections, where the negative slope is set close to $-\infty$. The embedding of nodes are updated as

$$\begin{aligned} \hat{h}_i^{l-1} &= \sum_{j\in N_i}(A_{ij}h_j^{l-1}), \\ h_i^l &= g_e([h_i^{l-1}; \hat{h}_i^{l-1}]; \theta_e), \end{aligned} \quad (3)$$

where N_j is the set of neighbors of the i-th node, $[\cdot;\cdot]$ is the concatenation operation, $g_e(;\theta_e)$ denotes the transforming block consisting of one MLP layer.

By stacking multiple layers, the GRE module is able to propagate information from different neighborhood levels.

4.1.3 Cross-range Fusion. Information within a fixed range is intensive and temporally characteristic. Instead of taking information across ranges as independent, we find it beneficial to mix it.

The cross-range fusion layer is implemented as multi-headed self-attention [21]. Obtaining updated node embeddings from the last layer of GRE, denoted as L, we feed the cross-range fusion model with the corresponding last time step, which is the closest step to our prediction target. Output of the encoder can be expressed as follows,

$$\begin{aligned} E_{out} &= MultiHeadAttn(Q_{en}, K_{en}, V_{en}), \\ Q_{en} &= K_{en} = V_{en} = [H^L_{1,\widetilde{L}_x^1}; ...; H^L_{i,\widetilde{L}_x^i}; ...; H^L_{K,\widetilde{L}_x^K}], \end{aligned} \quad (4)$$

where $H^L_{i,\widetilde{L}_x^i}$ is embedding of the last time step in the i-th exposure-infection range.

4.2 Decoder

The challenge of robust prediction arises when the statistic of observed data is incomplete. Though cruel, a common belief is that the reported death can be more reliable and serve as an indicator of the true process of the pandemic. It is promising that there might be strong merits in introducing deaths into the prediction model.

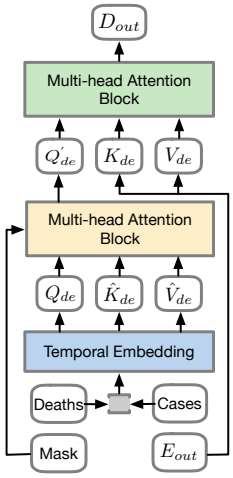


Figure 3: Decoder overview.

Encoder-decoder framework advances machine translation and has been widely used in other fields such as computer vision and time series processing. After encoding the infection process by modeling the mobility of the population, we here adopt a decoder to make final predictions by considering historical observed cases and hidden unobserved cases, which both contribute to future positive detections. We use a similar decoder structure as [21] and it mainly consists of an embedding layer and two identical multi-head attention layers. Figure 3 illustrates the structure of decoder.

4.2.1 Horizon-aware Embedding. The embedding layer is designed to temporally embed inputs with regard to the prediction horizon. Given historical data of cases and corresponding deaths, i.e., $C = [c_1, ..., c_{L_x}] = \mathcal{X}[:, 0]$ and $\mathcal{F} = [f_1, ..., f_{L_x}]$, prediction horizon Δ, a 1dConv layer with kernel size Δ is used to embed the sequence,

$$\mathcal{H}_{de} = \sigma(W_{de} * [C; \mathcal{F}] + b_{de}), \quad (5)$$

where $[:]$ denotes the slicing operation, W_{de} is the linear transforming matrix and b_{de} denotes the bias.

4.2.2 Attentive Encoding. Then, multi-head self-attention is performed over 2D-reshaped $\mathcal{H}_{de}$, with a mask that only allows information propagation within specific statistics, which is in order to keep variables' feature clean before introducing keys and values, i.e.,

$$\widetilde{H}'_{de} = MultiHeadAttn(Q_{de}, \hat{K}_{de}, \hat{V}_{de}, mask),$$
$$Q_{de} = [H_{de,1}; ...; H_{de,\widetilde{L}_x^{de}}] = \hat{V}_{de} = \hat{E}_{de}. \quad (6)$$

Next, we feed a multi-head cross-attention layer with the updated embeddings of the decoder input, which is denoted as $\widetilde{H}'_{de}$, and output of the encoder,

$$D_{out} = MultiHeadAttn(Q'_{de}, K_{de}, V_{de}),$$
$$Q'_{de} = \widetilde{H}'_{de}, K_{de} = V_{de} = E_{out}, \quad (7)$$

where $\widetilde{L}_x^{\Delta}$ is the post-convoluted sequence length of the decoder input, $D_{out} \in \mathbb{R}^{2\widetilde{L}_x^{de} \times C}$ is the computed output of the cross-attention layer.

4.3 Final Prediction

We then reshape D_{out} into $\mathbb{R}^{2\times\widetilde{L}_x^{de}\times C}$ and the final output is computed through a linear transformation of the concatenation of the encoder output and the corresponding last time step in D_{out}, i.e.,

$$\hat{\mathcal{Y}} = g_d([E_{out}; D_{out}[:, -1, :]]; \theta_d), \quad (8)$$

where $g_d(;\theta_d)$ is the transforming block.

We use mean square error (MSE) as the loss function. Algorithm 1 demonstrates the macro training process of the model.

5 EXPERIMENTS AND RESULTS

5.1 Datasets and Metrics

We use the monthly places patterns dataset [4] produced by SafeGraph for human mobility modeling. The dataset contains daily visitor and demographic aggregations of about 4.4MM POIs in the US. The duration of the dataset is from January 2018 to the present. We generate CBG-level human mobility by first aggregating POIs of a CBG and then making a summation of the "visits_by_day" column.

For COVID-19 statistics, we use the public JHU-CSSE COVID-19 dataset of the US [5] as gold-standard data. The dataset collects daily cases and deaths from when the county began reporting to present.

For evaluation, we produce two datasets by extracting counties in New York and California respectively. We considered the dataset duration of all counties to be between February 1-st, 2020 to December 31-st, 2020. For cases and deaths, since the start-reporting date varied from county to county, there can be missing data points at the beginning of the duration. We fill these missing data with 0s. For other kinds of missing data, i.e., the missing data in the mobility dataset and during the pandemic, we fill them with previous values.

New York: Contains data of 32 counties. The number of CBGs of each county ranges from 1 to 900 with an average number of 40.63, a median number of 4. The average daily CBG visitor count of a county between February 1-st, 2020 to December 31-st, 2020 is 42.77. The average daily incident confirmed cases of a county at the sampled date October 31-st, 2020 is 21.53 and weekly incident cases are 111.90.

[4]https://docs.safegraph.com/docs/places-schema#section-patterns
[5]https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_time_series

Algorithm 1 Network training of the model

Input:
County level historical case data and human mobility patterns measured by CBG visitor counts $\mathcal{X}$, historical death data $\mathcal{F}$, Δ;
Output:
Model parameters;
Initialize:
Hyper parameters: Input sequence length L_x, batch size b_n, kernel size and channel C of K convolutional kernels, self-attention related hyper parameters, GNN depth L, MLP layer number and embedding size;
for epoch in N epochs **do**
  for batch $\{[\mathcal{X}(t_1), \mathcal{F}(t_1)], ..., [\mathcal{X}(t_{b_n}), \mathcal{F}(t_{b_n})]\} \in$ training set **do**
    /* The encoder */
    Temporal embedding according to Equation 1;
    for layer l in L layers **do**
      Compute Adjacency matrix by Equation 2;
      Message passing and update node representations in Equation 3;
    end for
    Fuse information across ranges according to Equation 4;
    /* The decoder */
    Embedding decoder inputs by Equation 5 and making attentive prediction $\hat{\mathcal{Y}}(t_i + \Delta)$ by Equation 7, 8;
    Compute training loss as $\mathcal{L} = MSE(\hat{\mathcal{Y}}(t_i + \Delta), \mathcal{Y}(t_i + \Delta))$
    Perform gradient descend w.r.t. model parameters;
  end for
  /* Validation process (omitted)*/
  Save the better model according to RAE on validation set;
end for

California: Data of 34 counties is available. The number of CBGs of each county ranges from 1 to 1657 with an average number of 142.47, a medium number of 40. The average daily CBG visitor count of a county between February 1-st, 2020 to December 31-st, 2020 is 77.36. The average daily incident confirmed cases of a county at the sampled date October 31-st, 2020 are 107.76 and weekly incident cases are 647.15.

Key statistics of the two datasets are summarized in Table 1.

Table 1: Statistics of datasets.

	New York	California
# county	32	34
Ave. # CBGs	40.63	142.47
Ave. # daily cases	21.53	107.76
Avg. # weekly cases	111.90	647.15

As a common practice, we evaluate the model by three metrics, Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and Relative Absolute Error (RAE), which are computed as:

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=t_1}^{t_T}(\hat{y}_t - y_t)^2}, \quad (9)$$

$$MAE = \frac{1}{T}\sum_{t=t_1}^{t_T}|\hat{y}_t - y_t|, \tag{10}$$

$$RAE = \frac{\sum_{t=t_1}^{t_T}|\hat{y}_t - y_t|}{\sum_{t=t_1}^{t_T}|y_t - \overline{y_{t_1:t_T}}|}, \tag{11}$$

where $\hat{y}$ is the prediction output, y is the ground-truth value, $t \in [t_1, t_T]$ is the time instance in test set, $\overline{y}$ is the mean of set y.

5.2 Baselines

5.2.1 Weekly Prediction. The US CDC opened a project to collect, standardize, visualize and synthesize COVID-19 forecast data from global modeling groups [6]. The submitted forecasting files are week-ahead forecasts, which represent the total number of new cases reported during a given epiweek (from Sunday through Saturday, inclusive). We compare our model with the models submitted to CDC for the task of weekly prediction. Though only limited models are published as papers or open-sourced, their standardized prediction data is available [18]. Thus, instead of reproducing their models, we directly evaluate their prediction results. The submission update frequency, target prediction location, and granularity vary from group to group. Filtered by the availability of county-level prediction data lies between November 1-st, 2020 and December 31-st, 2020 (the "target_end_date" column), the compared models are:

- **CEID-Walk**: A statistical random walk model.
- **CMU-TimeSeries**: An autoregressive time-series model.
- **Google_Harvard-CPF**: An SEIR model fit with machine learning.
- **IowaStateLW-STEM**: A nonparametric spatiotemporal model.
- **JHUAPL-Bucky**: A metapopulation SEIR model.
- **LNQ-Ens1**: A machine learning model.
- **OliverWyman-Navigator**: A time-dependent SIR model.
- **UCLA-SuEIR**: An SuEIR model with machine learning.
- **UMass-MechBayes**: A mechanistic Bayesian compartment model.
- **USC-SI_kJalpha**: An SIR model.
- **UVA-Ensemble**: An ensemble of an auto-regressive model, a machine learning model, and an SEIR model.
- **COVIDhub-Baseline**: A baseline model by predicting with the most recent observation and historical difference.
- **COVIDhub-Ensemble**: An ensemble model of predictions meeting submission criteria of CDC.

A more detailed description of the compared models can be found at [3, 18].

5.2.2 Daily Prediction. It is also interesting to see if the model could perform well on forecasting COVID-cases daily. The task of daily prediction aims at forecasting the number of new cases reported for a given day. The test set is chosen as every day between November 1-st, 2020 to December 31-st (inclusive). We compare our model with state-of-the-art COVID-19 prediction model and time series forecasting models. The baselines are:

- **LSTM**: A 2-layer LSTM model, taking daily cases as input.
- **GRU**: A 2-layer GRU model, taking daily cases as input.
- **HMM**: A hidden Markov model by taking mean values, minimum values, and maximum values of cases of every five days as features.
- **Informer** [30]: An attention-based model for time series forecasting.
- **M-SEIR** [28]: A modified SEIR model integrating population migration data.
- **Mobility-SEIR** [5]: A SEIR model incorporating human mobility patterns.

5.3 Implementation Details

All experiments are implemented on 4 Intel(R) Xeon(R) E5-2690 v4 @ 2.60GHz CPUs and 2 TITAN XP GPUs.

5.3.1 Module Architecture. For the embedding layer of the encoder, the channel size is set as 32. For all MultiHeadAttn parts of encoder and decoder, the number of layers is set as 1, with queries, keys, and values of dimension 3, head number equals 3, and inner-layer of dimension 8. As for the predictor, embedding dimension of the MLP is optimized by the hyper-parameter optimization framework Optuna [7] within the range *[16, 64]*, step size 16.

5.3.2 Hyper-parameter Setting. For each dataset, we split it into 8:1:2 for training, validation, and test. The input length of each prediction is set as 64 for the California dataset and 48 for the New York dataset. The kernel sizes of the encoder embedding layer are set as [7, 14, 21]. The batch size is set as 8 for weekly prediction and chosen by Optuna in range *[8, 32]* with step size 8 for daily prediction. We choose the optimizer through Optuna, searching from *Adam, SGD, RMSprop*. We initialize the learning rate as 1e-5 and let it decay by 0.8 every 10 epochs. The weight decay rate is set as 1e-5. The total number of epochs is set as 500, through which we save the best model based on the RAE of the validation set and reload it for the final evaluation of the test set. We implement early stopping on validation RAE with patience 100. Unless explicitly stated, all reported results are on the test set. A dropout rate of 0.001 is applied to the attention and MLP layers. If not specified or tuned with Optuna, all parameters are the same on four datasets. For each test, we run the proposed model 10 times with different random seeds and report the mean values of metrics.

5.4 Main Results

5.4.1 Weekly Prediction. Results are reported in Table 2, with best results highlighted in bold, evaluated county number denoted as # county. All metrics are computed by county and we take the average value over counties to present final results. The same results preprocessing strategy is applied to the daily prediction results as illustrated in Table 3.

In most cases of the prediction horizon and datasets, the proposed method performs significantly better than baseline models. Almost all of the best results are achieved by the proposed method. In the New York dataset with prediction horizon 1 week, baseline prediction MAE is 70.679 and the proposed method improves the statistic to 51.404, bringing in up to 28% relative improvement. Note that though model UMass-MechBayes achieves the best results on

[6] https://www.cdc.gov/coronavirus/2019-ncov/cases-updates/forecasts-cases.html

[7] https://github.com/optuna/optuna

Table 2: Summary of weekly forecasting comparison results. Best results are highlighted in bold.

Dataset		CA			NY			Dataset		CA			NY		
Method	Metric	1 wk	2 wk	3 wk	1 wk	2 wk	3 wk	Method	Metric	1 wk	2 wk	3 wk	1 wk	2 wk	3 wk
Google_Harvard-CPF	RAE	0.9080	1.0852	1.1630	0.9460	1.6181	1.8954	UVA-Ensemble	RAE	1.2894	-	-	0.8100	1.1526	-
	MAE	4043.9	2013.3	2456.2	154.37	297.61	352.71		MAE	3601.1	-	-	147.12	196.63	-
	RMSE	9091.2	2579.6	3088.4	186.28	360.04	413.95		RMSE	6289.9	-	-	192.06	252.24	-
	# county	34			32				# county	34			32		
IowaStateLW-STEM	RAE	0.8323	1.1647	1.4303	0.8789	1.0065	1.2585	JHUAPL-Bucky	RAE	1.2445	1.5149	1.7876	1.0727	2.0222	2.8041
	MAE	1376.3	2267.4	2934.9	120.79	167.57	240.72		MAE	1331.8	1789.1	1851.6	182.57	313.28	384.05
	RMSE	1775.4	2799.9	3523.5	143.07	190.34	263.41		RMSE	1905.7	2475.5	2387.3	223.84	398.2	495.16
	# county	34			31				# county	34			32		
UCLA-SuEIR	RAE	0.6957	1.1266	1.4344	0.8649	1.232	1.5745	CEID-Walk	RAE	0.5733	0.8867	1.1421	0.5876	0.9090	1.2115
	MAE	1393.9	2554.7	3247.1	330.25	661.39	888.18		MAE	1100.4	1811.3	2424.5	106.53	189.16	264.36
	RMSE	1772.2	3000.2	3834.1	380.01	741.93	986.12		RMSE	1445.7	2322.2	2992.3	125.91	214.94	292.75
	# county	29			11	10			# county	34			32		
COVIDhub-Baseline	RAE	0.5736	0.8873	1.1463	0.5834	0.9029	1.2046	UMass-MechBayes	RAE	0.6112	0.7943	0.9998	**0.3887**	**0.6531**	**0.8576**
	MAE	1099.4	1814.8	2429.9	106.21	186.89	263.36		MAE	1374.9	1951.0	2323.3	288.84	477.63	635.11
	RMSE	1438.7	2322.3	2992.8	126.04	213.09	291.82		RMSE	1673.2	2463.6	2952.1	358.51	610.77	914.11
	# county	34			32				# county	26			8		
CMU-TimeSeries	RAE	0.6545	0.7487	0.7949	-	-	-	COVIDhub-Ensemble	RAE	0.5443	0.8433	1.0806	0.5275	0.7837	1.0876
	MAE	1834.2	2643.9	3134.9	-	-	-		MAE	989.18	1652.8	2160.1	91.719	151.64	228.93
	RMSE	2468.5	3646.0	3964.8	-	-	-		RMSE	1287.5	2107.5	2628.2	111.59	181.64	256.54
	# county	19			-				# county	34			32		
USC-SI_kJalpha	RAE	0.5681	0.8291	0.9973	0.6198	0.9300	1.1801	OliverWyman-Navigator	RAE	0.5162	0.6704	0.7775	0.5566	0.7804	1.0530
	MAE	964.01	1363.0	1778.7	97.191	150.30	217.08		MAE	897.25	1147.9	1536.6	81.148	118.25	160.66
	RMSE	1229.9	1815.2	2304.7	118.17	187.37	259.07		RMSE	1246.3	1669.1	1942.6	102.88	142.19	187.20
	# county	34			32				# county	34			32		
LNQ-Ens1	RAE	0.4916	0.7379	0.9363	0.4654	0.7201	0.9515	Ours	RAE	**0.4662**	**0.6617**	**0.7569**	0.4102	0.7200	0.9241
	MAE	731.30	1252.5	1832.9	70.679	117.32	171.38		MAE	**658.60**	**1108.2**	**1423.2**	**51.404**	**115.51**	**150.58**
	RMSE	968.74	1732.2	2408.0	89.876	142.73	192.52		RMSE	**909.68**	**1609.1**	**1886.4**	**67.194**	**138.94**	**169.71**
	# county	34			32				# county	34			32		

RAE of the New York dataset, the statistics are counted only among 8 counties.

Though for all models, the prediction accuracy drops as the horizon becoming farther, it is worth noting that the proposed approach is more robust compared to baselines: the relative declination is less significant.

5.4.2 Daily Prediction. Compared with the task of weekly prediction, daily prediction is an even harder task given that the noise of data is more evidential if looked into daily. Results are reported in Table 3, with best results highlighted in bold. Again, across all of the prediction horizons and all of the datasets, the proposed method performs significantly better than baseline models. More concretely, the proposed model gains 40%, 32%, 50%,42% relative improvements over the best baselines, in horizon 14, 21 California dataset and horizon 14, 21, New York dataset, respectively.

Another interesting observation is that the state-of-the-art time series forecasting model Informer [30] doesn't perform well on the task of COVID-19 prediction and the variance of prediction results over different horizons is small. This could result from that Informer is originally proposed for long sequence time-series forecasting and such a prediction of relatively short length and at long horizon is not its strength. Instead, some naive but general methods such as HMM and GRU perform much better. Moreover, it can also be observed that for the proposed model, the prediction accuracy does not always decline as the horizon gets larger. This is mainly owing to the weekly periodic pattern of data reporting in some counties, therefore easy for the model to learn.

5.5 Ablation Study

We conduct detailed ablation study in Table 4 to explore the effect of each component of the proposed approach. Results are calculated on New York dataset for the task of 1 week ahead weekly prediction. The original version of th proposed method **Ours** and **LNQ-Ens1** are listed for comparison. All variants are trained with the same batch size. We now discuss the results by variants.

5.5.1 Effect of Key Encoder Components. To evaluate the effect of the encoder, we create two variants.

Ours-w/o-mobility. Ours-w/o-mobility is a variant that human mobility patterns that are not modeled. We achieve so by excluding CBG visitor counts data in the input. The problem can thus be defined as given historical cases and deaths, predict future cases. Removing human mobility in inputs, the variant Ours-w/o-mobility can only mine the exposure-infection process from reported cases, thus inevitably lead to worse results. It is interesting to note that though the variant achieves inferior performance over the proposed method, it still outperforms state-of-the-art baseline LNQ-Ens1 significantly, which indicates that the model could still be effective even on datasets where human mobility patterns are not available.

Ours-w/o-cross. In this variant we directly utilize the features embedded in each defined range instead of fusing them. We maintain the structure before the cross-range fusion module discussed

Table 3: Summary of daily forecasting comparison results. Best results are highlighted in bold.

Dataset		California				New York			
Method	Metric	3 d	7 d	14 d	21 d	3 d	7 d	14 d	21 d
LSTM	RAE	2.1933	2.1914	2.2646	2.3420	2.3367	2.3346	2.4068	2.4899
	MAE	814.14	817.81	837.97	854.02	121.44	121.24	124.10	127.88
	RMSE	924.83	931.66	952.11	960.71	126.99	126.74	129.33	133.00
Informer	RAE	1.3481	1.3345	1.3528	1.3564	1.7764	1.7788	1.7842	1.7399
	MAE	581.45	580.65	582.14	581.79	72.644	72.781	72.774	72.582
	RMSE	755.97	755.22	756.76	756.38	83.908	84.013	84.015	83.856
M-SEIR	RAE	1.0321	1.1406	1.4288	2.5094	1.4797	1.5272	1.5271	1.5611
	MAE	486.68	543.29	578.51	937.85	56.745	55.696	53.311	53.966
	RMSE	668.74	721.99	811.88	1252.1	67.722	66.551	63.955	64.519
HMM	RAE	1.0642	1.0773	1.1487	1.2129	0.9771	1.0317	1.0828	1.1426
	MAE	418.34	427.50	507.23	534.15	35.185	34.474	38.580	38.404
	RMSE	664.51	670.70	732.75	780.32	46.241	43.658	48.068	47.912
Mobility-SEIR	RAE	1.0720	1.0836	1.0938	1.1266	1.0308	1.0678	1.1585	1.2406
	MAE	402.79	412.89	418.68	434.86	37.081	37.371	41.903	48.085
	RMSE	539.50	544.46	549.70	596.00	45.830	46.422	51.105	57.676
GRU	RAE	1.0345	0.9972	1.0728	1.1132	1.0661	1.0736	1.1733	1.2603
	MAE	364.06	375.51	409.50	425.81	37.669	39.471	45.399	50.839
	RMSE	516.47	522.44	567.78	586.41	46.382	47.352	53.100	58.658
Ours	RAE	**1.0174**	**0.7940**	**0.8229**	**0.8426**	**0.8979**	**0.7126**	**0.7540**	**0.7944**
	MAE	**266.50**	**215.86**	**245.80**	**289.12**	**21.620**	**18.415**	**19.272**	**22.192**
	RMSE	**383.04**	**313.18**	**358.29**	**410.41**	**27.037**	**24.545**	**26.316**	**28.880**

Table 4: Summary of ablation study results on the New York dataset, 1-week-ahead prediction.

Method	RAE	MAE	RMSE
LNQ-Ens1	0.4654	70.679	89.876
Ours	0.4102	51.404	67.194
Ours-w/o-mobility	0.4366	61.681	79.089
Ours-w/o-cross	0.6267	115.28	131.12
Ours-w/o-decoder	0.5439	85.866	103.87
Ours-w/o-death	0.5616	93.377	111.76
Ours-w/o-residual	0.4486	61.463	76.766

in Section 4.1.3 and simply concatenate the feature from the corresponding last time steps to obtain E_{out}. It can be observed in Table 5 that Ours-w/o-cross gives the worst results among all models and variants, which indicates that the fusion layer plays an important role in the model. The reason why cross-range fusion is important could be that when the incubation period of a data point is uncertain, it is necessary to assume multiple alternatives and allow information to be propagated among them.

5.5.2 Effect of Key Decoder Components. Three variants are created to evaluate the decoder.

Ours-w/o-decoder. In a neural machine translation task, decoders serve the role of matching source language to the target language. However, a decoder is usually not necessary for normal time series forecasting tasks. This is especially true when the encoder explicitly models the serial feature of an input sequence. So does the decoder in our model help? We remove the entire decoder to create the Ours-w/o-decoder variant. The output of the encoder, i.e., E_{out}, is fed to the final prediction layer, which contains a 2-layer MLP to compute the final prediction. The results show that the variant performs much worse than the proposed model and the baseline as well. Therefore, it can be concluded that it is effective to comprehensively decode the embedded message obtained from the encoder.

Ours-w/o-death. Besides verifying that a decoder works, it is also interesting to explore which specific part of the decoder contributes the most. In the Ours-w/o-death variant, we only use reported case data as decoder inputs. In other words, the model only has access to observed cases data. We can tell from the result that such a variant is actually a misuse of decoder, given it downgrades the performance and is even inferior to Ours-w/o-decoder. The key functionality of an encoder-decoder framework is that the encoder transforms an input into certain states while the decoder maps the states with processed queries. The proposed method can easily be trained to accommodate such functionality: using observables and unobservables to fit the detected.

Ours-w/o-residual. In this variant, we remove the residual connection in Equation 8. It can be told from the results that such a residual connection can contribute positively to the model. It is worth mentioning the variant itself also works well since the result is significantly superior to the state-of-the-art baseline LNQ-Ens1.

We conclude from the ablation study that not using mobility data is fine but is inferior to the originally proposed approach; cross-range fusion helps enhance prediction accuracy a lot because it jointly models various incubation periods; a proper decoder is necessary; using only the observed cases to decode the embedding information is a misuse of the decoder, which means death data plays a rather important role in the decoder; and residual connection makes a positive contribution to the decoder.

5.6 Effect of Model Parameters

We now evaluate the proposed framework on key hyperparameters.

5.6.1 Effect of Input Length and Kernel Size. The size of the convolutional kernel influences how the model observes the raw temporal data. To further investigate how the temporal embedding works

Table 5: Effect of input length and kernel size on the New York dataset, 1-week-ahead prediction.

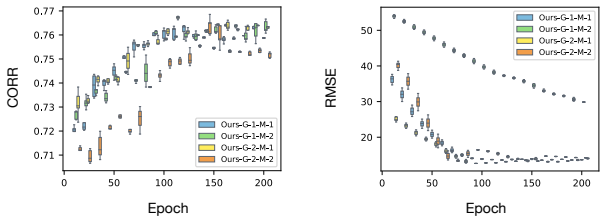
Metric	Ranges	32	48	64
RAE	(7)	0.5455	0.5494,	0.5542
	(7, 14)	0.4360	0.4366	0.4505
	(7, 21)	0.6102	0.6071	0.5966
	(14, 21)	0.6953	0.6846	0.6598
	(7, 14, 21)	0.5535	0.5218	0.4102
MAE	(7)	91.553	96.771	96.029
	(7, 14)	57.683	54.537	54.277
	(7, 21)	109.55	101.41	101.41
	(14, 21)	111.19	122.23	112.61
	(7, 14, 21)	91.443	81.355	51.404
RMSE	(7)	108.55	112.74	113.04
	(7, 14)	74.280	72.322	72.557
	(7, 21)	129.90	122.86	122.71
	(14, 21)	126.34	137.94	128.24
	(7, 14, 21)	109.21	98.617	67.194

and why cross-range fusion is effective, we test our model with different input sequence lengths and incubation ranges. (,) denotes the size(s) of the kernel(s) of the encoder, which represents the incubation range(s). For simplicity, we choose the kernels that are the integral multiplication of 7. Only kernel sizes no greater than 21 are considered because according to epidemic research [13] the incubation period of COVID-19 rarely exceeds 21 days.

Generally, a longer length of input enables the model to learn from long-term patterns and thus enhances the prediction accuracy. For instance, the proposed model equipped with the incubation ranges (7, 14, 21) and 64-length input sequence improves by 26%, 44%, and 39% on RAE, MAE, and RMSE respectively, compared with the one with an input sequence of length 32. This trend doesn't hold for all range combinations, for example, an inverse trend can be found on the proposed model equipped with range (7). Another observation is that the combination (7, 14, 21) achieves the best results among all other combinations across all evaluated input lengths. This could result from that the embeddings obtained from and fused across these perspectives are comprehensive enough.

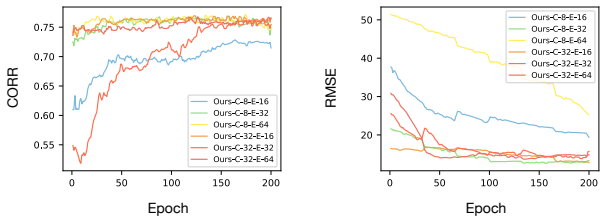
5.6.2 Effect of Model Depth. To evaluate the effect of GNN and MLP layers' depth, we test our model on the New York dataset w.r.t. RMSE and Empirical Correlation Coefficient (CORR) with different depth of GNNs and MLPs. CORR is computed as: $CORR = \frac{\sum_{t=t_1}^{t_T}(y_t-\overline{y_{t_1:t_T}})(\hat{y}_t-\overline{\hat{y}_{t_1:t_T}})}{\sqrt{\sum_{t=t_1}^{t_T}(y_t-\overline{y_{t_1:t_T}})^2\sum_{t=t_1}^{t_T}(\hat{y}_t-\overline{\hat{y}_{t_1:t_T}})^2}}$. We show the results on validation set of Ontario County in box-plots as demonstrated in Figure 4. G- in **Ours-G-{1,2}-M-{1,2}** indicates the GNN depth and M- denotes the MLP depth. We run models 200 epochs for all settings. It could be conclude that that 1-layer MLP is enough; larger depth of graph attention layer enables node to learn from a global perspective and thus enhances the expressive capacity of graph neural networks, which fastens the learning process.

5.6.3 Effect of Embedding Size. In order to study the impact of embedding size of the embedding layers and MLP layers, we compare six variants of the proposed model on the New York dataset and show validation curves w.r.t. CORR and RMSE. The name **Ours-C-{8, 32}-E-{16, 32, 64}** indicates a variant of the proposed model



(a) (b)

Figure 4: Effect of embedding size. CORR and RMSE are measured on 1-week-ahead prediction for Ontario County of the New York dataset.



(a) (b)

Figure 5: Effect of model depth. CORR and RMSE are measured on 1-week-ahead prediction of Ontario County in the New York Dataset.

with convolutional channel size chosen from {8, 32} and MLP embedding dimension chosen from {16, 32, 64}. We plot the learning curve on the validation set and the total number of epochs is fixed as 200, as illustrated in Figure 5. Generally, a larger embedding dimension achieves better results. Except for Ours-C-8-E-16, models equipped with other settings roughly converge at CORR=0.75 in the 200th epoch; and Ours-C-8-E-64 is the slowest learner on RMSE while Ours C-32-E-16 is the fastest. This gives a hint that a larger channel size of the embedding layer can model temporal features more thoroughly, which is beneficial to predict the overall pattern of a time series. The embedding size of the MLP layer together with the above-mentioned channel size can greatly influence the training process of predicting the exact number of COVID-19 cases. Therefore, it is suggested to jointly consider the effect of both two parameters when tuning the model.

6 CONCLUSIONS

In this paper, we propose a simple and effective multi-range encoder-decoder framework for COVID-19 infection prediction. It learns the temporal transmission between groups based on reported cases and human mobility data with multiple exposure-infection ranges considered, and makes predictions based on both observed and hidden cases. In this way, unobservable but vital factors affecting virus spreading can be fully exploited. Extensive experiments in weekly and daily prediction tasks validate the effectiveness of the approach across different prediction horizons and datasets.

ACKNOWLEDGMENTS

This work is partially supported by Natural Science Foundation of China (No. 61972069, 61836007 and 61832017).

REFERENCES

[1] Duygu Balcan, Vittoria Colizza, Bruno Gonçalves, Hao Hu, José J Ramasco, and Alessandro Vespignani. 2009. Multiscale mobility networks and the spatial spreading of infectious diseases. *Proceedings of the National Academy of Sciences* 106, 51 (2009), 21484–21489.

[2] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. 2015. *Time series analysis: forecasting and control*. John Wiley & Sons.

[3] cdcepi. 2020. COVID-19 Forecast Model Descriptions = https://github.com/cdcepi/COVID-19-Forecasts/blob/master/COVID-19_Forecast_Model_Descriptions.md.

[4] Texas Advanced Computing Center. 2020. The University of Texas COVID-19 Modeling Consortium. https://covid-19.tacc.utexas.edu/dashboards/us/.

[5] Serina Chang, Emma Pierson, Pang Wei Koh, Jaline Gerardin, Beth Redbird, David Grusky, and Jure Leskovec. 2021. Mobility network models of COVID-19 explain inequities and inform reopening. *Nature* 589, 7840 (2021), 82–87.

[6] B. Yu et al. 2020. COVID-19 Severity Prediction. https://covidseverity.com/.

[7] Salah Ghamizi, Renaud Rwemalika, Maxime Cordy, Lisa Veiber, Tegawendé F Bissyandé, Mike Papadakis, Jacques Klein, and Yves Le Traon. 2020. Data-driven simulation and optimization for covid-19 exit strategies. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3434–3442.

[8] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. 2019. Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting. (2019), 922–929.

[9] William L Hamilton. 2020. Graph representation learning. *Synthesis Lectures on Artifical Intelligence and Machine Learning* 14, 3 (2020), 1–159.

[10] Qianyue Hao, Lin Chen, Fengli Xu, and Yong Li. 2020. Understanding the Urban Pandemic Spreading of COVID-19 with Real World Mobility Data. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3485–3492.

[11] Gabriel G Katul, Assaad Mrad, Sara Bonetti, Gabriele Manoli, and Anthony J Parolari. 2020. Global convergence of COVID-19 basic reproduction number and estimation from early-time SIR dynamics. *Plos one* 15, 9 (2020), e0239800.

[12] The Johns Hopkins University Justin Lessler lab and Google Inc. 2020. Exploring methods for merging mechanistic and statistical models to forecast epidemics. https://github.com/HopkinsIDD/EpiForecastStatMech/.

[13] Stephen A Lauer, Kyra H Grantz, Qifang Bi, Forrest K Jones, Qulu Zheng, Hannah R Meredith, Andrew S Azman, Nicholas G Reich, and Justin Lessler. 2020. The incubation period of coronavirus disease 2019 (COVID-19) from publicly reported confirmed cases: estimation and application. *Annals of internal medicine* 172, 9 (2020), 577–582.

[14] Bryan Lewis, Srinivasan Venkatramanan, Madhav Marathe, and Christopher L. Barrett. 2020. COVID-19 Pandemic Response. https://biocomplexity.virginia.edu/project/covid-19-pandemic-response.

[15] Kevin Linka, Mathias Peirlinck, and Ellen Kuhl. 2020. The reproduction number of COVID-19 and its correlation with public health interventions. *Computational Mechanics* 66, 4 (2020), 1035–1050.

[16] Abdelghafour Marfak, Doha Achak, Asmaa Azizi, Chakib Nejjari, Khalid Aboudi, Elmadani Saad, Abderraouf Hilali, and Ibtissam Youlyouz-Marfak. 2020. The hidden Markov chain modelling of the COVID-19 spreading using Moroccan dataset. *Data in brief* 32 (2020), 106067.

[17] Nick H Ogden, Aamir Fazil, Julien Arino, Philippe Berthiaume, David N Fisman, Amy L Greer, Antoinette Ludwig, Victoria Ng, Ashleigh R Tuite, Patricia Turgeon, et al. 2020. Artificial intelligence in public health: Modelling scenarios of the epidemic of COVID-19 in Canada. *Canada Communicable Disease Report* 46, 8 (2020), 198.

[18] rechlab. 2020. COVID-19 Forecast Models = https://github.com/reichlab/covid19-forecast-hub/tree/master/data-processed.

[19] Yuan Tian, Ishika Luthra, and Xi Zhang. 2020. Forecasting COVID-19 cases using Machine Learning models. *medRxiv* (2020).

[20] Ashleigh R Tuite, David N Fisman, and Amy L Greer. 2020. Mathematical modelling of COVID-19 transmission and mitigation strategies in the population of Ontario, Canada. *Cmaj* 192, 19 (2020), E497–E505.

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *NIPS'17*.

[22] M Vollmer, S Mishra, H Juliette, et al. 2020. Using mobility to estimate the transmission intensity of COVID-19 in Italy: a subnational analysis with future scenarios. Imperial College London. 2020.

[23] Billy M Williams and Lester A Hoel. 2003. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. *Journal of transportation engineering* 129, 6 (2003), 664–672.

[24] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. 2020. Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks. In *KDD'20*.

[25] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Graph wavenet for deep spatial-temporal graph modeling. *arXiv preprint arXiv:1906.00121* (2019).

[26] Congxi Xiao, Jingbo Zhou, Jizhou Huang, An Zhuo, Ji Liu, Haoyi Xiong, and Dejing Dou. 2020. C-Watcher: A Framework for Early Detection of High-Risk Neighborhoods Ahead of COVID-19 Outbreak. *arXiv preprint arXiv:2012.12169* (2020).

[27] Haoyan Xu, Yida Huang, Ziheng Duan, Jie Feng, and Pengyu Song. 2020. Multivariate Time Series Forecasting Based on Causal Inference with Transfer Entropy and Graph Neural Network. *arXiv preprint arXiv:2005.01185* (2020).

[28] Zifeng Yang, Zhiqi Zeng, Ke Wang, Sook-San Wong, Wenhua Liang, Mark Zanin, Peng Liu, Xudong Cao, Zhongqiang Gao, Zhitong Mai, et al. 2020. Modified SEIR and AI prediction of the epidemics trend of COVID-19 in China under public health interventions. *Journal of thoracic disease* 12, 3 (2020), 165.

[29] Huaxiu Yao, Xianfeng Tang, Hua Wei, Guanjie Zheng, and Zhenhui Li. 2019. Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction. In *AAAI'19*, Vol. 33. 5668–5675.

[30] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021*. AAAI Press, online.

[31] Mingyuan Zhou. 2020. Discrete Dynamical Systems (DDS) for COVID-19 Forecast. https://dds-covid19.github.io/.