Distribution Agreement

Muran Qin April 10, 2023

Data-Driven Fine-Grained Epidemic Modeling via Graph Neural Networks

By

Muran Qin

Li Xiong
Advisor

Computer Science

Li Xiong
Advisor

Nosayba El-Sayed
Committee Member

Andreas Züfle
Committee Member

2023

Data-Driven Fine-Grained Epidemic Modeling via Graph Neural Networks

By

Muran Qin

Li Xiong
Advisor

An abstract of
a thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Science with Honors

Computer Science

2023

Abstract

Data-Driven Fine-Grained Epidemic Modeling via Graph Neural Networks
By Muran Qin

Fine-grained epidemic modeling is crucial for controlling the spread of diseases such as COVID-19. While many graph-based deep learning frameworks for pandemic forecasting achieved powerful performance, seldom use other relevant data sources besides the disease case surveillance data. This paper presents a framework for using the publicly available Social Connectedness Index (SCI) and Social Vulnerability Index (SVI) to enhance the baseline model. These datasets provide valuable insights into the social interactions and socioeconomic status of each location, both potentially significant factors for epidemic spreading dynamics. Experiments were conducted on the U.S. county-level granularity over three datasets with different time frames and geographical scales. We found that SCI and SVI both improve the performance over the original model on some prediction horizons while having comparative performance on other prediction horizons, demonstrating the promising effectiveness of using social-related data sources on pandemic forecasting. Finally, we suggested potential future research directions on data-driven pandemic forecasting.

Data-Driven Fine-Grained Epidemic Modeling via Graph Neural Networks

By

Muran Qin

Li Xiong
Advisor

A thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Science with Honors

Computer Science

2023

Acknowledgments

First and foremost, I would like to thank my advisor, Professor Xiong Li for guiding me through my study and introducing me to the Hyperlocal Risk Monitoring project. She was also the Professor who taught my first Machine Learning courses. Her engaging teaching made me decide to dive deep into the relevant studies.

I would also like to thank Professor Cyrus Shahabi, and Sepanta Zeighami who consistently provide numerous critical feedback and advice to my work.

In addition, I also want to thank my committee members Professor Nosayba El-Sayed and Professor Andreas Züfle who both taught me valuable knowledge through their expertise during my undergraduate studies.

Last but not least, I want to thank my parents who emotionally and financially supported me throughout my college years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Over the past three years, the COVID-19 pandemic has had a significant, if not devastating, impact on the world, causing widespread illness, death, and economic disruption. Scientists have made an extensive effort in the area of disease spread modeling to accurately forecast the spread of the virus, which could guide the communities to react, and is crucial for the policy-makers to make well-informed decisions on how to develop effective interventions and marshal limited healthcare resources to minimize the destruction brought by the disease. Even though COVID-19 is fading away, studies in relevant fields could help society to cope better and respond faster in future epidemics.

Traditional epidemic modeling models are compartmental based and estimate disease transmission dynamics at the population level [21, 37]. However, they make strong assumptions with respect to a stationary process and barely consider the spatial dependencies between locations, which is extremely important in epidemic modeling. More recent works consider both the spatial and temporal dependencies of the disease by learning simultaneously on multiple locations [24, 29]. Graph Neural Networks (GNNs) [41] are extremely good at capturing the spatial influences each location has on another location due to the message-passing mechanism where each location is represented as a node in the graph. As a result, most recent works use a GNN-based framework and achieve better performance on epidemic forecasting [7, 10, 13, 30]. We noticed that most effort for better performance is directed to building a more powerful model, but not much has been done to explore the effect of using additional data sources in the GNN-based models besides the epidemic cases time series. There are plenty of data sources available providing information on mobility patterns, social interactions, and socioeconomic status of the locations.

Although many previous works argued that additional data sources are hard to collect, the data sources we used in this study – Social Connectedness Index (SCI) and Social Vulnerability Index (SVI) – are both publicly available. As their names suggest, these datasets provide valuable insight into the social connection between locations and the location's vulnerability to an epidemic, respectively. Our goal is to find a suitable baseline model and incorporate these data sources into the model to improve its U.S. county-level COVID-19 forecasting performance.

Our key contributions can be summarized as follows:

- We summarize the most relevant state-of-the-art graph-neural-network-based COVID-19 forecasting models and provide both a quantitative comparison and a qualitative comparison between the models on the U.S.-county level case prediction.

- We identify two social data sources – SCI and SVI – that provide information regarding the social connections between counties and the social conditions of the counties. With these datasets, we study the effect of using social connections and socioeconomic data in GNN-based epidemic modeling.

- We provide a framework for incorporating SCI and SVI data into a state-of-the-art graph neural network model, Cola-GNN, and evaluated the performance. We enhance Cola-GNN's performance over a longer prediction horizon of 2 to

4 weeks by using SVI-aware attention. We also summarize the implications of using each data.

The rest of this paper is organized as follows: Section 2 summarizes the related works on GNNs and epidemic forecasting. Section 3 introduces the formulation of the research problem and the common data structures used. Section 4 provides a comparative study between four state-of-the-art models. Section 5 incorporates SCI and SVI into Cola-GNN and evaluates their performances and implications. Finally, Section 6 concludes the key findings in the paper and discusses potential future research direction.

Chapter 2

Related Work

2.1 Graph Neural Networks

A graph is a ubiquitous data structure that describes objects and their relations. It is useful in many scenarios such as modeling molecule and protein structures, social networks, traffic networks, etc [4, 11, 12, 23, 25, 32, 40]. Hence, tremendous efforts have been made in this area, aiming to learn from the graphical relationships in the data. Graph neural networks (GNNs) are neural models that capture the dependence of graphs via message passing between the nodes of graphs [41]. The message-passing operation in a GNN is typically defined using a graph convolutional operation, which is similar to a convolutional operation used in image processing. The convolutional operation is applied to the feature vectors of each node and its neighboring nodes, allowing each node to incorporate information about its neighbors into its own feature vector.

Some state-of-the-art GNN layers include Chebyshev Convolutional Network (ChebNet) [9] that uses Chebyshev polynomials to perform spectral convolution on graph-structured data; Graph Convolution Network (GCN) [16] consists of multiple layers that aggregate information from neighboring nodes and update the node embedding;

Graph Attention Network (GAT) [27] uses attention mechanisms to weigh the importance of neighboring nodes during message passing. Other variants such as Graph Isomorphism Network (GIN) [36] and Graph Convolutional Recurrent Network (GCRN) [22] also achieve powerful performance and are applied to different tasks.

2.2 Multivariate Time Series Forecasting

A multivariate time series (MTS) is a collection of time-dependent variables, where each variable is dependent on the values of other variables in the collection. MTS forecasting has many applications in areas such as traffic forecasting, electrocardiogram forecasting, and stock price prediction [2, 6, 7, 34]. The majority of the early methods follow a statistical approach. The autoregressive integrated moving average (ARIMA) captures both the autocorrelation (auto-regressive) and moving average patterns of a time series, while also taking into account any trends or seasonality in the data by differencing (integrated) the series. The vector autoregressive model (VAR) extends the autoregressive (AR) by capturing the linear interdependencies among the series. Despite the popularity of statistical models due to their simplicity and interpretability, they make strong assumptions with respect to a stationary process and are not easily scalable to multivariate time series data. Deep models are more effective to capture non-linearity in the data. Many deep MTS forecasting models employ convolutional neural networks to capture the inter-series correlations and recurrent neural networks to capture the intra-series temporal correlations [17, 33, 35].

In the recent literature, graph-based models are increasingly popular for MTS forecasting tasks because the relationship between series can be effectively modeled as graphs [15, 18, 28]. ST-GCN provides a framework for traffic prediction which integrate graph convolution and gated temporal convolution through spatio-temporal convolution blocks [38]. StemGNN [7] combines Graph Fourier Transform (GFT) and

Discrete Fourier Transform (DFT) to capture inter and intra-series dependencies on the spectral domain.

2.3 Epidemic Modeling

Compartmental models in epidemiology including SIR, SEIR, and SIRD divide a population into compartments such as susceptible (S), exposed (E), infected (I), recovered (R), and death (D) and model the transmission dynamics as a parameterized systems of equations. SuEIR [42] and modified SEIR [37] both extend on SEIR. The former takes into account the untested/unreported cases, while the latter accounts for the mobility of individuals. However, these compartmental models suffer from the assumption that all individuals within a compartment are homogeneous and cannot account for spatial dynamics. Deep models are generally more expressive and could more accurately model complicated epidemic dynamics. Epidemic forecasting can be viewed as an MTS forecasting problem where each node is a geographical region (State, county, etc) associated with its daily cases. While most general MTS forecasting models mentioned earlier can be applied, many models are designed specifically for epidemic forecasting. Cola-GNN [10] learns time-series embedding for long-term influenza-like illness (ILI) prediction by combining graph structures and time-series features in a dynamic propagation process. STAN [13] and CausalGNN [30] are both hybrid models that combine the graph input features with epidemiological context to predict the number of COVID-19 cases. By incorporating disease transmission dynamics into graph neural networks, the forecasts are regularized and achieve better performance.

Chapter 3

Problem Definition

We formulate the COVID forecasting problem as a multivariate time series prediction problem where the time series are the newly infected cases every day. $X = [X_1, \ldots, X_T] \in \mathbb{R}^{N \times T}$ denotes the multivariate time series input and N is the number of time series or locations. X_t denotes the observed value for every node at timestamp t. Each row of X represents a location and contains the corresponding univariate time series values. Given the historical timestamps, the objective is to predict X_{T+h} where h refers to the prediction horizon, or leadtime of the prediction.

Since there are strong spatio-temporal dependencies in epidemic propagation, all the models in this study model the input as either a static or dynamic graph, where each node represents a location. The static graphs are denoted as $\mathcal{G}_{Static}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of N nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. The graph $\mathcal{G}_{Static}$ is associated with a feature matrix $C \in \mathbb{R}^{N \times T}$ and a weighted adjacency matrix $A \in \mathbb{R}^{N \times N}$. C is usually the same as X. The dynamic graphs are denoted as $\mathcal{G}_{Dynamic}(\mathcal{V}, \mathcal{E}, \mathcal{T})$, where $\mathcal{T}$ is the set of T timestamps. $\mathcal{G}_{Dynamic} = \{\mathcal{G}_1(\mathcal{V}_1, \mathcal{E}_1) \ldots \mathcal{G}_T(\mathcal{V}_T, \mathcal{E}_T)\}$, where each $\mathcal{G}_t$ represents a different static graph at timestamp t. At every timestamp, the graph $\mathcal{G}_t$ is associated with a feature matrix $C_t \in \mathbb{R}^{N \times C}$ where C is the feature number and an weighted adjacency matrix $A_t \in \mathbb{R}^{N \times N}$.

Chapter 4

Comparative Study

In this section, four state-of-the-art graph-based COVID-forecasting models including StemGNN, Cola-GNN, STAN, and CausalGNN are introduced at a high level from the earliest to the most recent[1]. We then compared the models on both a qualitative and quantitative basis. The goal for the comparative study is to select the most suitable base model for the enhancement study in Section 5.

4.1 Background

4.1.1 StemGNN

Spectral Temporal Graph Neural Network (StemGNN) [7] improves the accuracy of MTS forecasting by capturing inter-series correlations and temporal dependencies jointly in the spectral domain using Graph Fourier Transform (GFT) [1, 9] and Discrete Fourier Transforms (DFT), respectively. StemGNN consists of a latent correlation layer that automatically learns the inter-series correlation through a gated recurrent unit (GRU) [8] encoding and a self-attention mechanism [26], two StemGNN blocks with residual connections where each block carries out the Spectral Graph Con-

[1] All the mathematical details and code for each of the models can be found in the original paper (except the code for CausalGNN)

volution [16], and an output layer. The StemGNN network uses both a forecast loss and a backcast loss during the training process.

4.1.2 Cola-GNN

Cross-location Attention based Graph Neural Networks (Cola-GNN) [10] learns time series embeddings for long-term influenza-like illness (ILI). It models the impact of one location on another location dynamically using a location-aware attention mechanism [3] and extracts important features from each time series using a multi-scale dilated convolutional module [39, 20]. The cross-location attentions and the local temporal features are then passed through a flu propagation model which consists of graph message-passing layers. The final representation for each location combines the graph node embedding and the last Recurrent Neural Network (RNN) hidden state of the original time series. An output layer consisting of a fully-conneced layer and an activation function is used for the downstream prediction task.

4.1.3 STAN

Spatio-temporal Attention Netork (STAN) [13] takes in a dynamic graph without edge weights as input where the node feature at each timestamp consists of the latest values from historical data within a sliding window. First, graph attention network [27] is applied to the graph at each timestamp to capture the spatio-temporal trends of the pandemic dynamics. Then a GRU [8] for each location takes in the embedding of the corresponding node at each timestamp to learn the temporal dependencies. The final GRU hidden state for each location is used to predict the number of cases in the future. In addition to the traditional forecast loss, a dynamics-based loss term based on the SIR model was added to enhance long-term predictions.

4.1.4 CausalGNN

Causal-Based Graph Neural Networks (CausalGNN) [30] learns spatio-temporal embedding where graph input features and epidemiological context are combined via a mutual learning mechanism using graph-based non-linear transformations. The transformations include a feature encoding that encodes the input at each timestamp, a temporal encoding that learns the temporal dependencies, and an attention-based dynamic GNN layer (AGCN) used to capture the spatio-temporal disease dynamics. On top of the GNN framework, a causal module including causal encoding, causal decoding, and susceptible(S)-infected(I)- recovered(R)-deceased(D) (SIRD) simulation [19] is added to provide epidemiological context for the time series embedding for each location.

4.2 Qualitative Comparison

Besides StemGNN, which learns on the spectral domain, all other models learn on the graph domain through message passing between nodes. For capturing temporal dependencies, StemGNN, Cola-GNN, and STAN all used an RNN. On top of RNN, StemGNN used DFT, and Cola-GNN used multi-scaled dilated convolution. CausalGNN uses a temporal encoding consisting of affine transformations at each timestamp to learn the temporal dependencies.

Since the eventual goal of the comparative study is directed at learning the potential incorporation of additional data sources. Therefore, the rest of the comparison is going to focus on the type of input, the type of graph, and the graph construction method of each model along with a discussion of the extensibility of each model.

Table 4.1 summarized the most relevant comparisons between the models. All the models use most of their input as the node features. StemGNN and Cola-GNN only use the original MTS as the node features. On top of the original MTS, STAN and CausalGNN incorporate other information such as the latitude, longitude, and population into the node features along with recovered and death time series that are also used for the SIR/SIRD simulation.

Model	**Node Features**	**Edge Weight**	**Graph Type**
StemGNN	Confirmed cases	Self Attention using GRU embedded features	Static
Cola-GNN	Confirmed cases	Additive Attention using RNN embedded features & geographical adjacency matrix	Static
STAN	Latitude, longitude, population density, population size, active cases, total cases, number of hospitalizations and ICU stays	Graph Attention Network	Dynamic
CausalGNN	Confirmed cases, recovered, death, population density, latitude, longitude	Additive Attention	Dynamic

Table 4.1: Comparison of the node features, edge weights, and type of graph used in StemGNN, Cola-GNN, STAN, and CausalGNN.

More on edge weight learning, all the models use some variation of the attention mechanism to learn the correlation between locations, while Cola-GNN uses geographical adjacency as an additional input source to compute the edge weights. The most significant difference in terms of the graphs is the graph type each model constructed. A dynamic graph, as defined in Section 3 is a graph that changes during each timestamp, whereas a static graph does not. StemGNN and Cola-GNN both construct a static graph. The edge weights are computed before graph convolution and do not change later. On the other hand, STAN and CausalGNN both implicitly construct a dynamic graph. At each timestamp, the model encodes the node features at time t $C_t \in \mathbb{R}^{N \times C}$. While the node features at each timestamp are encoded for each location, the temporal module of the models learns the temporal dependencies.

Overall, all the model has powerful learning ability because they all jointly learn both spatial and temporal patterns. However, since StemGNN operates on the spectral domain, it is more difficult to incorporate additional input sources into the model. Cola-GNN is more extensible because it operates on the original graph domains. Data regarding each location can be added as node features to the graph and data regarding the relationship between each location can be used to manipulate edge weights. If the data is dynamic, such as temperature and mobility data which changes with respect to time, they can be incorporated into STAN and CausalGNN because of their use of a dynamic graph. The additional data at each timestamp could be added to the graph at each corresponding timestamp.

4.3 Quantitative Comparison

For the quantitative comparison, we only compared StemGNN and Cola-GNN. Since STAN and CausalGNN both use a significant amount of additional data sources in the epidemiological simulation component such as the recovered and death cases etc, it is unfair to compare their performance. Although the original Cola-GNN uses geographical adjacency data, it is not a crucial part of the model and does not have a significant impact on the performance. Moreover, the goal of this study is to investigate the incorporation and effectiveness of additional data sources, it is better to use a simpler model that only takes the daily cases time series as the input node features.

4.3.1 Dataset

US-County data are collected from New York Times' COVID-19 data repository[2]. The data consist of the daily cumulative counts of COVID-19 cases in each U.S. counties. The raw data is preprocessed into three datasets of different lengths and scales. Using a varying time range allows us to study how the model performs for different disease trends. The description for each dataset is shown in Table 4.2. More details are recorded in Appendix A. Instead of predicting the cumulative count, we computed the daily newly infected counts and predict the change in COVID-19 cases.

Dataset	Begin Date	End Date	Days	#Locations
California	4/7/2020	10/6/2021	548	48
US-Long	4/7/2020	10/6/2021	548	753
US-Short	4/7/2020	4/6/2021	365	753

Table 4.2: Time range and the dimension of each COVID-19 cases time series datasets.

4.3.2 Setup

We used the original implementation and the default model parameters for both StemGNN[3] and ColaGNN[4]. Both implementations are in Pytorch. The experiments are run using Python 3.7 and PyTorch 1.13 with CUDA 11.7 on a Linux server with an Nvidia GeForce RTX 3090 GPU.

The time series data is smoothed using a 6-day moving average and normalized using the maximum and minimum value of the training set. The prediction is denormalized for evaluation. A 70%-15%-15% train-valid-test split along the time dimension was used. The splits are done chronologically so the model is only evaluated

[2] https://github.com/nytimes/covid-19-data
[3] https://github.com/microsoft/StemGNN
[4] https://github.com/amy-deng/colagnn

on future values. A 28-day input window is used for the predictions. All the seeds are set to 42 for a reproducible result.

4.3.3 Evaluation Metric

In the experiments, we denote the prediction to be $\{\hat{y}_1, \dots, \hat{y}_n\}$ and the ground truth to be $\{y_1, \dots, y_n\}$. For evaluation, locations are not distinguished during evaluation and the following metrics are adopted.

Root Mean Squared Error (RMSE) measures the difference between the prediction and the ground truth. The squared term penalizes larger errors more severely:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$$

Mean Absolute Error (MAE) measures the average magnitude of errors in the prediction:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|$$

Preason's Correlation (PCC) measures the strength of linear dependence between two variables and is scale invariant. The value ranges between -1 to 1 with 1 meaning a total positive linear correlation (the higher the better performance in our setting):

$$\mathrm{PCC} = \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

The above evaluation metrics are reported with varying **Leadtime**, which is the same as the prediction horizon, it is the number of timestamps that the model predicts in advance. For example, given $X_{N,T}$ as the input and a leadtime of 5, the ground truth value for the prediction is $X_{N,T+5}$.

4.3.4 Results

We evaluate StemGNN and Cola-GNN on a leadtime of 2, 5, 7, 14, and 28 days. A leadtime equal to 1 is ignored because the symptom monitoring data is usually delayed. 14 and 28-day (2-week and 4-week) horizon is common for evaluating COVID-19 forecasting models. Table 4.3 summarized the result for different leadtimes on all three datasets. The result for each dataset spans 5 columns. The top half shows the RMSE and the bottom half shows the PCC. The first row on each half is the leadtime in days.

	California					**US-Long**					**US-Short**				
RMSE (↓)	2	5	7	14	28	2	5	7	14	28	2	5	7	14	28
StemGNN	118	378	266	**253**	**258**	83	334	110	272	225	302	**100**	128	**250**	**307**
Cola-GNN	**78**	**206**	**227**	368	441	**70**	**89**	**79**	**138**	**177**	**26**	145	**99**	334	728
PCC (↑)	2	5	7	14	28	2	5	7	14	28	2	5	7	14	28
StemGNN	0.967	0.891	0.887	**0.887**	**0.887**	0.931	0.666	0.875	0.639	0.475	0.630	**0.844**	**0.736**	0.619	**0.732**
Cola-GNN	**0.983**	**0.908**	**0.903**	0.799	0.676	**0.951**	**0.917**	**0.935**	**0.808**	**0.663**	**0.978**	0.766	0.693	**0.708**	-0.109

Table 4.3: RMSE and PCC performance for StemGNN and Cola-GNN on different leadtimes and different datasets using a 28-day input. Models trained and tested using a 70%-15%-15% train-valid-test data split.

On the **California** dataset, Cola-GNN has better performance than StemGNN when the leadtime is shorter, while StemGNN has better performance for a leadtime of 2 and 4 weeks. On the **US-Long** dataset, Cola-GNN outperforms StemGNN over all leadtimes. Finally, on the **US-Short** dataset, StemGNN is having trouble predicting over a 2-day leadtime and Cola-GNN fails to make a decent prediction over a 28-day leadtime.

Looking at Table 4.3 row-wise, Cola-GNN's performance gets steadily worse as the leadtime increases, which is intuitive. It is hard to predict the future that is further ahead. However, it is not the same trend for StemGNN. The performance fluctuates as leadtime increases. For example, on the **California** dataset, StemGNN performed the worst on a 5-day leadtime, while having about the same performance for 7, 14, and 28-day leadtime.

4.4 Analysis

The overall performance of Cola-GNN is slightly better than StemGNN. Cola-GNN's performance is also more stable. This is somewhat expected as Cola-GNN is designed to forecast influenza influenza-like illness (ILI). ILI and COVID-19 are both contagious repository diseases and have similar spreading dynamics. On the other hand, StemGNN is designed as a general MTS forecasting model. StemGNN works especially well on tasks such as electrocardiogram forecasting and traffic forecasting where the data, unlike the available COVID-19 data, is more periodic and has a shorter period.

Qualitatively, since Cola-GNN learns on the original graph domain instead of the spectral domain, it is more interpretable and extensible than StemGNN. The graph message-passing module in Cola-GNN explicitly models the pandemic by considering both the spread from neighboring locations and the dynamics of similar locations.

Although STAN and CausalGNN are also very extensible COVID-19 forecasting models that use epidemiological context, they are not considered in the quantitative comparison in this study because they all require a significant amount of data sources besides the daily case data. Nonetheless, it would be interesting to compare STAN and CausalGNN in a future study.

Recall that the goal for this Section is to select the most suitable base model for the enhancement study in Section 5. Due to Cola-GNN's performance, interpretability, and extensibility over StemGNN, we are choosing Cola-GNN as the baseline method.

Chapter 5

Enhancement Study

In this Section, we introduce the geographical adjacency data used in the original Cola-GNN and the two additional social-related data sources that could be meaningful to the epidemic dynamics and incorporate them into Cola-GNN to study the significance of each data source.

5.1 Dataset

5.1.1 Geographical Adjacency

County Adjacency[1] data are collected from the National Bureau of Economic Research (NBER). The dataset contains the geographical adjacency information of all the U.S. counties.

5.1.2 Social Connectedness Index

Facebook's social connectedness index (SCI)[2] [5] uses Facebook friendship ties to measures the strength of social connections between two geographic regions. Assuming a large portion of friends on Facebook is also friends in the physical world, SCI

[1] https://www.nber.org/research/data/county-adjacency

[2] https://data.humdata.org/dataset/social-connectedness-index

can be an indicator of real-world mobility, which is highly correlated to the disease spreads.

The SCI uses an anonymized snapshot of active users on Facebook and assigns them to locations based on their information and activity on Facebook. Formally, the SCI between two regions i and j is defined as:

$$\text{SCI}_{i,j} = \frac{\text{FB\_Connectionss}_{i,j}}{\text{FB\_Users}_i \times \text{FB\_Users}_j}$$

Where FB_Users_i and FB_Users_j are the number of Facebook users in the region i and j, and $\text{FB\_Connectionss}_{i,j}$ is the total number of friendship connections between users from the two locations. The SCI essentially measures the relative probability of a Facebook friendship link between a user in location i and a user in location j.

The public release version of the data scaled SCI values between 1 to 1,000,000,000. A small amount of random noise is added to the values before rounding to the nearest integer to ensure the privacy of the data. The data are collected on October 2021.

5.1.3 Social Vulnerability Index

CDC/ATSDR social vulnerability index (SVI)[3] is created by Centers for Disease Control and Prevention (CDC) and Agency for Toxic Substances and Disease Registry (ATSDR) to help public health officials and emergency response planners identify and map the communities that will most likely need support before, during, and after a hazardous event.

SVI describes the resilience of a community to disasters such as an earthquake or pandemic like COVID-19. The vulnerability depends on the degree to which a community exhibits certain social conditions, including high poverty, low percentage of vehicle access, or crowded houses. The SVI dataset contains 16 social factors including unemployment, racial and ethnic minority status, and disability, many collected

[3] https://www.atsdr.cdc.gov/placeandhealth/svi/data_documentation_download.html

from the U.S. Census data. SVI also assigns each region an overall ranking.

In this study, we use the 2020 county-level SVI dataset. Instead of using the overall ranking, we selected a subset of features because we are interested in the similarity between each county in terms of demographics and social determinants. There are a total of 18 features, including estimated total population, estimated population density which is computed by dividing the estimated total population by the area in square miles, and estimated percentage of: people below 150% poverty, unemployment, housing cost burdened occupied housing units, people with no high school diploma (age 25+), uninsured, people ages 65 and older, people ages 17 and younger, disability, single-parent households with children under 18, people (age 5+) who speak English "less than well," minority, housing in structures with 10 or more units, mobile homes, occupied housing units with more people than rooms, households with no vehicle available, and people in group quarters. The estimated percentage are on a scale of 0 to 100.

5.2 Cola-GNN: A Closer Look

Although Cola-GNN is introduced briefly in Section 4.1.2, it is necessary to review some details of the Cola-GNN model before incorporating the new data sources. The Cola-GNN framework diagram from the original paper [10] is shown in 5.1. It has four parts: 1) Directed Spatial Influence Learning; 2) Multi-scale Dialated Convolution; 3) Graph Message Passing – Propagation; 4) Output Layer – Prediction. In the following paragraphs, we are going to introduce the relevant mathematical detail of each part.

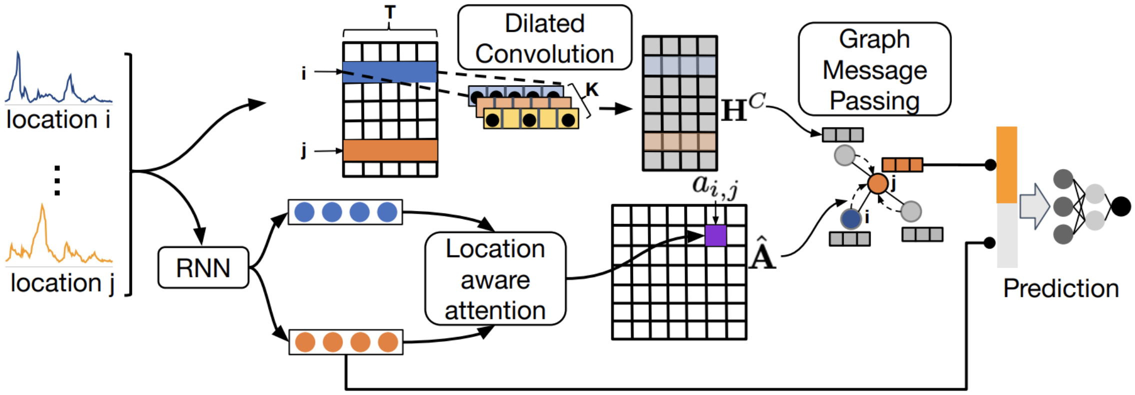

Figure 5.1: The overview of the proposed framework. The original time series for each location are copied to two components: (1) an RNN model (bottom) for learning directed spatial influence; and (2) a dilated convolution model (top) for learning multi-level temporal features. Figure from *Cola-GNN: Cross-location Attention based Graph Neural Networks for Long-term ILI Prediction* by Deng, et al [10].

5.2.1 Directed Spatial Influence Learning

Cola-GNN uses both the additive attention and the geographical adjacency matrix to dynamically model the impact of one location on other locations. This step is essentially learning the edge weights of the graph. Given the input data $X = [X_1, \dots, X_T]$, a global Recurrent Neural Network (RNN) [31] is used to capture the temporal dependencies of all locations. At each timestamp, RNN updates its hidden state. Let $h_i \in \mathbb{R}^D$ denote the last hidden state for each node i where D is the RNN hidden dimension. The last hidden states are used to learn the attention coefficient $a_{i,j}$ with an additive attention mechanism [3]:

$$a_{i,j} = v^T g(W^s h_i + W^t h_j + b^s) + b^v \tag{5.1}$$

where g is an activation function, $W^s, W^t \in \mathbb{R}^{d_a \times D}$, $v \in \mathbb{R}^d_a$, $b^s \in \mathbb{R}^d_a$, and $b^v \in \mathbb{R}$ are trainable parameters. d_a is a hyperparameter controlling the hidden dimensions of the attention. Given all pairs of $a_{i,j}$, an attention coefficient matrix A^{Att} is obtained where each row indicated the degree of impact by other nodes on the current node.

A^{Att} is then normalized row-wise.

Cola-GNN also considers the geographical proximity between nodes as closer nodes are likely to have a higher impact on each other due to population mobility. In the final step of learning edge weight, the geographical adjacency matrix A^{Geo} where $a^{Geo}_{i,j} = 1$ if node i and j are neighbors is combined with A^{Att} by an element-wise gate M adapted from the feature fusion gate [14]:

$$\tilde{A}^{Geo} = D^{-1/2} A^{Geo} D^{-1/2} \tag{5.2}$$

$$M = \sigma(W^m A^{Att} + b^m 1_N 1_N^T) \tag{5.3}$$

$$\hat{A} = M \odot \tilde{A}^{Geo} + (1_N 1_N^T - M) \odot A^{Att} \tag{5.4}$$

where D is the degree matrix defined as $d_{ii} = \sum_{j=1}^{N} a_{ij}$, $W^m \in \mathbb{R}^{N \times N}$, and $b^m \in \mathbb{R}$ are trainable parameters. Eq. 5.2 normalizes A^{Geo}, Eq. 5.3 learns the feature fusion gate M from A^{Att}, Eq. 5.4 computes the final edge weights $\hat{A}$ by using a weighted average between the normalized A^{Geo} and A^{Att} with weights from M.

5.2.2 Multi-Scale Dilated Convolution

Before the graph message passing, Cola-GNN embeds the time series of each location using a multi-scaled dilated convolution [20], to capture temporal dependencies at different levels of granularity:

$$d_s[i] = \sum_{l=1}^{L} x_s[i + k \times l] \times c[l] \tag{5.5}$$

where x_s is the time series of each location or the s^{th} row of the input X, d_s is the output feature vector, c is the convolutional filter of length L, and k is the dilation rate. To capture both short-term and long-term patterns, K filters with dilation rate k_s and k_l ($k_l > k_s$) were used. The convolution is applied to each location. The final convolution output for each location is obtained by concatenating the output feature vector from all the filters.

5.2.3 Graph Message Passing – Propagation

After learning the cross-location attentions and the local temporal features, graph message passing is used to model the pandemic propagation among all locations. Each location corresponds to a node in the graph where the initial node features are the convolution output and the adjacency matrix is equal to $\hat{A}$. In each graph message passing layer, the node features are updated as follows:

$$h_i^{(l)} = g\left(\sum_{j \in \mathcal{N}} \hat{a}_{i,j} W^{(l-1)} h_j^{(l-1)} + b^{(l-1)}\right) \tag{5.6}$$

where g denotes a nonlinear activation function, $W^{(l-1)} \in \mathbb{R}^{F^{(l)} \times F^{(l-1)}}$ is the weight matrix for hidden layer l with a dimension of $F^{(L)}$, and $b^{(l-1)} \in \mathbb{R}^{F^{(l)}}$ is the bias. $\mathcal{N}$ is the set of neighbors of node i. $h_i^l \in \mathbb{R}^{F^{(l)}}$ is the embedded node feature at the l^{th} layer with a dimension of $F^{(l)}$. After l message passing layers, the final embedding $h_i^{(l)}$ is obtained for all the locations.

5.2.4 Output Layer – Prediction

The final prediction $\hat{y}_i$ for each location is computed using both the RNN features and graph message passing features:

$$\hat{y}_i = \phi(\theta^T [h_{i,T}; h_i^{(l)}] + b^\theta) \tag{5.7}$$

where $h_{i,T} \in \mathbb{R}^D$ is the RNN final hidden state used to compute the attentions in Section 5.2.1, $h_i^l \in \mathbb{R}^{F^{(l)}}$ is the final node embedding from Section 5.2.3. ϕ is an activation function and $\theta \in \mathbb{R}^{D+F(l)}$, $b^\theta \in \mathbb{R}$ are trainable parameters.

5.2.5 Optimization

The model is trained by optimizing the $l1$-norm loss (MAE) via gradient descent:

$$\mathcal{L} = \sum_{i=1}^{N} \sum_{m=1}^{n_i} |y_{i,m} - \hat{y}_{i,m}| \tag{5.8}$$

where n_i is the number of samples for location i and is the same across all locations, $y_{i,m}$ is the ground truth value for location i in sample m, $\hat{y}_{i,m}$ is the corresponding predicted value.

5.3 Proposed Method

There are two ways to potentially use the additional data sources in Cola-GNN – to improve the edge weight learning or to improve the node feature learning before graph message passing. In this study, we incorporate SCI and SVI into the graph edge weight learning process. Since the goal of the model is to predict the number of new cases (which are represented as node features), adding SCI and SVI as node features in the graph message passing or the final embedding before prediction will likely hurt the performance. We expect the additional data sources to provide deeper insight into the relationship between nodes and guide the message-passing process. To do so, we proposed two replacements to the location aware attention module in the original Cola-GNN as shown in Fig 5.2. The SCI aware attention utilizes SCI data while the SVI aware attention utilized SVI data. This is achieved by replacing the geographical adjacency matrix in the location aware attention module with an SCI matrix and SVI similarity matrix.

For N locations $\{n_1, \dots n_N\}$, we define three types of matrices $A \in \mathbb{R}^{N \times N}$. First the geographical adjacency matrix A^{Geo} which is used in the original Cola-GNN. It is computed as[4]:

[4] By default, each node is adjacent to itself

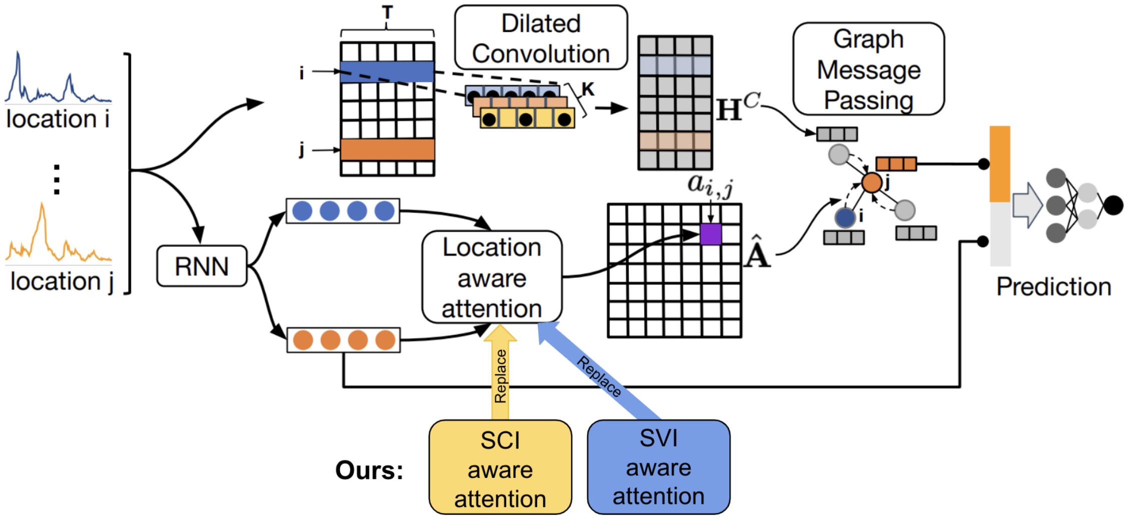

Figure 5.2: Our modification on the original Cola-GNN framework. The original location aware attention is replaced by 1) SCI aware attention and 2) SVI aware attention.

$$a_{i,j}^{Geo} = \begin{cases} 1 & \text{if } n_i \text{ and } n_j \text{ are adjacent} \\ 0 & \text{otherwise} \end{cases} \tag{5.9}$$

The SCI matrix A^{SCI} is defined as:

$$a_{i,j}^{SCI} = SCI_{i,j} \tag{5.10}$$

where $SCI_{i,j}$ is the scaled SCI value between n_i and n_j.

Finally, for the SVI similarity matrix A^{SVI}, each feature is first normalized to the range of 0 to 1. The SVI feature matrix $SVI \in \mathbb{R}^{N\times 18}$ contains the SVI features for n_i on the i^{th} row, denoted as SVI_i. A^{SVI} is defined as:

$$a_{i,j}^{SVI} = 1 - \frac{d(SVI_i, SVI_j)}{\|SVI_i\|} \tag{5.11}$$

where $d(SVI_i, SVI_j)$ is the Euclidean distance between the features of n_i and n_j and $\|SVI_i\|$ is the number of SVI features which is 18 in our study and the same for all locations. All the weights are guaranteed to be nonnegative because the distance is divided by the maximum distance.

Note that all three matrices are symmetric and measure the similarity between locations using different criteria. A^{Geo} represents the geolocational similarity between locations, A^{SCI} represents the social connectedness between locations, whereas A^{SVI} represents the demographical and socioeconomical similarity between locations.

To incorporate A^{SCI} and A^{SVI}, they are first normalized using the same method as in Eq. 5.2, denoted as $\hat{A}^{SCI}$ and $\hat{A}^{SVI}$. Then we replace the geographical adjacency in the original model with $\hat{A}^{SCI}$ and $\hat{A}^{SVI}$ in Eq. 5.4.

5.4 Experiment

The experiment setup for Cola-GNN is the exact same as in Section 4.3.2. The performance of location aware attention, SCI aware attention, and SVI aware attention are summarized in Table 5.1.

	California					**US-Long**					**US-Short**				
RMSE (↓)	2	5	7	14	28	2	5	7	14	28	2	5	7	14	28
Original	**78**	206	**227**	**368**	441	**70**	**89**	**79**	138	177	26	145	99	334	728
SCI	85	**191**	235	456	442	74	92	96	**108**	173	**23**	123	**57**	247	758
SVI	82	215	**227**	370	**426**	99	97	87	195	**172**	**23**	**121**	166	**210**	**317**
PCC (↑)	2	5	7	14	28	2	5	7	14	28	2	5	7	14	28
Original	**0.983**	0.908	0.903	0.799	**0.676**	**0.951**	**0.917**	**0.935**	0.808	0.663	0.978	0.766	0.693	**0.708**	-0.109
SCI	**0.983**	0.901	**0.914**	0.763	0.631	0.943	**0.917**	0.916	**0.881**	**0.685**	**0.983**	**0.829**	**0.890**	-0.048	0.522
SVI	**0.983**	**0.919**	0.905	**0.823**	0.659	0.901	0.904	0.922	0.615	0.679	0.982	0.767	0.791	0.685	**0.717**

Table 5.1: RMSE and PCC performance for original Cola-GNN, replacing geographical adjacency with SCI, and SVI on different leadtimes and different datasets using a 28-day input. Models trained and tested using a 70%-15%-15% train-valid-test data split.

On the **California** dataset, all three methods achieved similar performance. On the larger **US-Long** dataset, SCI aware attention slightly improved the 14 and 28-day leadtime prediction over the original model, whereas SVI aware attention showed comparable performance. Finally, on the **US-Short** dataset, all methods had a comparable performance on shorter horizons (2, 5, and 7 days). However, on longer horizons (14 and 28-day), SVI aware attention outperforms both SCI aware attention and the original method. More specifically, the original model had trouble forecasting for the 28-day leadtime, but SVI aware attention achieved a decent prediction with both a significantly lower RMSE (reduced by more than a half) and higher PCC.

5.5 Analysis

Although SCI and SVI aware attention slightly improved longer-term forecasting, we cannot conclude that using SCI or SVI is always better than using geographical adjacency. More extensive experiments are needed to verify the results. However, we can conclude that both our new methods have at least comparable results to the original method. At the end of this section, we are going to qualitatively compare the matrices used in all three methods.

To compare the difference between each adjacency matrices, we used a heatmap to visualize the $\hat{A}^{Geo}$, $\hat{A}^{SCI}$, and $\hat{A}^{SVI}$ for the **California** dataset as shown in Fig 5.3, 5.4, and 5.5.

For all three matrices, the diagonal has a relatively high weight. This is desirable because a location's future cases should be heavily dependant on itself. $\hat{A}^{SCI}$ has an outstanding high weight on the diagonal and diminishing weights elsewhere. Social connections and friendship often indicate real-world friendship, therefore the number of connections within a node could be significantly higher than connections across the nodes. $\hat{A}^{SVI}$ has the opposite pattern, although the diagonal entries have relatively higher weights, it is not very distinguished from other entries. The heatmap of the $\hat{A}^{SVI}$ looks a lot denser than SCI. This could be due to the computation of the SVI matrix resulting in the majority of weight before normalization being in a close range

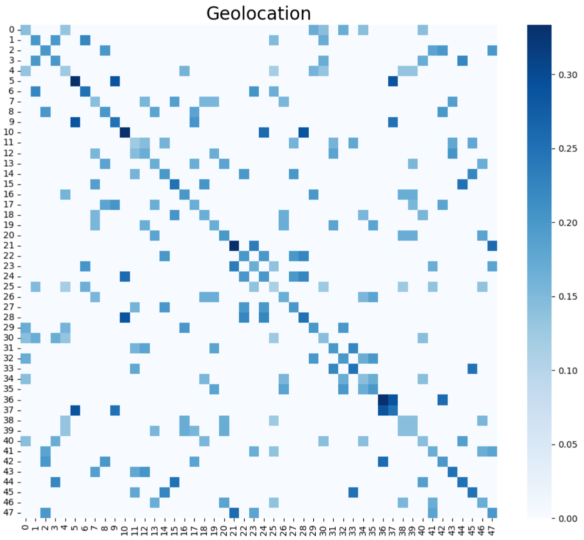

Figure 5.3: Heatmap for the normalized geographical adjacency matrix of 48 California counties.

so they are not as distinguishable after normalization. Finally, $\hat{A}^{SCI}$ has the highest range of 0 to 1, $\hat{A}^{Geo}$ ranges from 0 to 0.35, and $\hat{A}^{SVI}$ value are on a smaller magnitude form 0 to 0.024.

One advantage of SCI and SVI over geographical adjacency is that they can model more complex relations beyond geographical boundaries. For example, in the U.S. dataset, non of the counties in New York is connected to California via the geographical adjacency matrix. But both the SCI matrix and SVI similarity matrix connect counties that are not geographically adjacent. This is desirable in pandemic modeling since people travel around and a portion of cases in New York have certainly spread to California during the COVID-19 pandemic. The remote connections SCI makes can indicate real-world long-distance friendships and mobility. For distant locations, two counties with similar SVI features have similar socioeconomic environments which may indicate similar spread patterns.

Overall, the SCI and SVI aware attention showed promising performance. Further study to modify the matrix computation and normalization methods or the

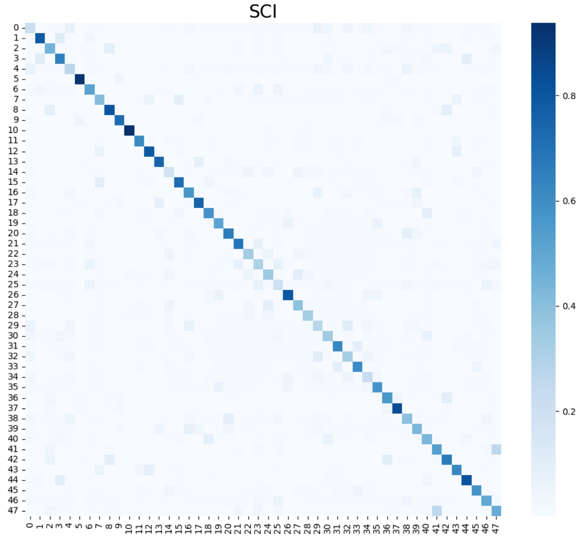

Figure 5.4: Heatmap for the normalized SCI matrix of 48 California counties.

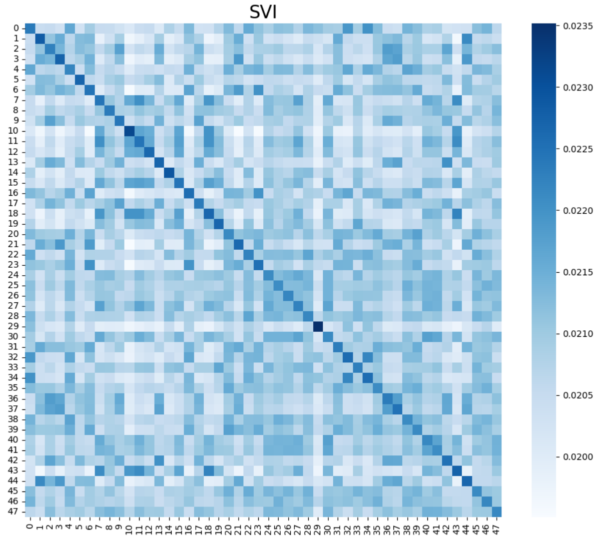

Figure 5.5: Heatmap for the normalized SVI similarity matrix of 48 California counties.

incorporation method could potentially unveil the benefits these social connections and socioeconomic data sources provide to county-level epidemic forecasting.

Chapter 6

Conclusion

In this study, we compared four state-of-the-art graph-based epidemic forecasting frameworks regarding the data they use as input and the way they model epidemic dynamics among locations as graphs. We further compared the performance of two of them, StemGNN and Cola-GNN, on the U.S. county-level dataset. Furthermore, we identified two data sources, SCI and SVI that could provide insights for epidemic forecasting. We proposed methods for incorporating them into the Cola-GNN framework and conducted an experiment to study the performance. Finally, we analyzed the insight each data source provides and its advantages. By using SCI and SVI, We improved the long-term forecasting performance of Cola-GNN. However, the overall performance improvement is not consistent and further study is needed to verify the empirical effectiveness of SCI and SVI data.

We also acknowledge some of the improvements that could be built on this study. For example, a more careful design of the SCI and SVI matrices so they are on the same scale. We could also try out different combinations of the three similarity matrices using a weighted average so more information is passed into the model. Another potentially useful improvement is to make the graph less dense by thresholding the edge weights. A sparser graph could reduce computational complexity and reduce over-fitting.

For future works on data-driven epidemic modeling, there are many other data sources that could be useful for learning the epidemic dynamics. Some of those data sources include: 1) SafeGraph's point of interest visit (POI) data[1], which provide detailed real-world mobility information for a sample of the population, 2) USC's Understanding America Survey (UAS) data[2], which provide a sample of behavior for people in each location, and 3) New York Times's mask-use data[3] which similarly provides an indicator for the level of cautiousness for the people in a region. Since COVID-19 is short compared to influenza and flu data, future studies could also incorporate additional data on influenza and flu modeling to further study the importance of various data sources.

[1] https://docs.safegraph.com/docs/weekly-patterns#section-weekly-patterns-schema

[2] https://covid19pulse.usc.edu/

[3] https://github.com/nytimes/covid-19-data/tree/master/mask-use

Appendix A

Dataset

For all U.S. data, we only included data from the 50 states[1]. To obtain the dataset presented in Table 4.2 from the raw cases data, we first computed the daily change in cases. Then, we filtered out counties with NA values or with a total change in case less than a threshold during the 548-day period from 4/7/2020 to 10/6/2021.

For the **California** dataset, we used a threshold of 3,000. The resulting 48 counties are: *Alameda, Amador, Butte, Calaveras, Contra Costa, Del Norte, El Dorado, Fresno, Glenn, Humboldt, Imperial, Kern, Kings, Lake, Los Angeles, Madera, Marin, Mendocino, Merced, Monterey, Napa, Nevada, Orange, Placer, Riverside, Sacramento, San Benito, San Bernardino, San Diego, San Francisco, San Joaquin, San Luis Obispo, San Mateo, Santa Barbara, Santa Clara, Santa Cruz, Shasta, Siskiyou, Solano, Sonoma, Stanislaus, Sutter, Tehama, Tulare, Tuolumne, Ventura, Yolo*, and *Yuba*. Their alphabetical order corresponds to the counties visualized in the heatmap in Fig 5.3, 5.4, and 5.5.

For **US-Long** and **US-Short** datasets, we limit the number of counties by using a higher threshold of 10,000, resulting in 753 counties.

[1] hrefhttps://state.1keydata.com/https://state.1keydata.com/

Bibliography

[1] Rie Ando and Tong Zhang. Learning on graph with laplacian regularization. *Advances in neural information processing systems*, 19, 2006.

[2] Adebiyi A Ariyo, Adewumi O Adewumi, and Charles K Ayo. Stock price prediction using the arima model. In *2014 UKSim-AMSS 16th international conference on computer modelling and simulation*, pages 106–112. IEEE, 2014.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[4] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in neural information processing systems*, 33:17804–17815, 2020.

[5] Michael Bailey, Rachel Cao, Theresa Kuchler, Johannes Stroebel, and Arlene Wong. Social connectedness: Measurement, determinants, and effects. *Journal of Economic Perspectives*, 32(3):259–80, 2018.

[6] Khac-Hoai Nam Bui, Jiho Cho, and Hongsuk Yi. Spatial-temporal graph neural network for traffic forecasting: An overview and open research issues. *Applied Intelligence*, 52(3):2763–2774, 2022.

[7] Defu Cao, Yujing Wang, Juanyong Duan, Ce Zhang, Xia Zhu, Congrui Huang,

Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, et al. Spectral temporal graph neural network for multivariate time-series forecasting. *Advances in neural information processing systems*, 33:17766–17778, 2020.

[8] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[9] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.

[10] Songgaojun Deng, Shusen Wang, Huzefa Rangwala, Lijing Wang, and Yue Ning. Cola-gnn: Cross-location attention based graph neural networks for long-term ili prediction. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 245–254, 2020.

[11] Frederik Diehl, Thomas Brunner, Michael Truong Le, and Alois Knoll. Graph neural networks for modelling traffic participant interaction. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 695–701. IEEE, 2019.

[12] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for adacare: Explainable clinical health status representation learning via scale-adaptive feature extraction and recalibrationsocial recommendation. In *The world wide web conference*, pages 417–426, 2019.

[13] Junyi Gao, Rakshith Sharma, Cheng Qian, Lucas M Glass, Jeffrey Spaeder, Justin Romberg, Jimeng Sun, and Cao Xiao. Stan: spatio-temporal attention network for pandemic prediction using real-world evidence. *Journal of the American Medical Informatics Association*, 28(4):733–743, 2021.

[14] Yichen Gong and Samuel R Bowman. Ruminating reader: Reasoning with gated multi-hop attention. *arXiv preprint arXiv:1704.07415*, 2017.

[15] Amol Kapoor, Xue Ben, Luyang Liu, Bryan Perozzi, Matt Barnes, Martin Blais, and Shawn O'Banion. Examining covid-19 forecasting using spatio-temporal graph neural networks. *arXiv preprint arXiv:2007.03113*, 2020.

[16] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[17] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks, 2018.

[18] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2017.

[19] Elena Loli Piccolomini and Fabiana Zama. Monitoring italian covid-19 spread by a forced seird model. *PloS one*, 15(8):e0237417, 2020.

[20] Liantao Ma, Junyi Gao, Yasha Wang, Chaohe Zhang, Jiangtao Wang, Wenjie Ruan, Wen Tang, Xin Gao, and Xinyu Ma. Adacare: Explainable clinical health status representation learning via scale-adaptive feature extraction and recalibration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 825–832, 2020.

[21] Sen Pei and Jeffrey Shaman. Initial simulation of sars-cov2 spread and intervention effects in the continental us. *MedRxiv*, pages 2020–03, 2020.

[22] Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. Structured sequence modeling with graph convolutional recurrent networks. In *Neural Information Processing: 25th International Conference, ICONIP 2018,*

Siem Reap, Cambodia, December 13-16, 2018, Proceedings, Part I 25, pages 362–373. Springer, 2018.

[23] Jonathan Shlomi, Peter Battaglia, and Jean-Roch Vlimant. Graph neural networks in particle physics. *Machine Learning: Science and Technology*, 2(2): 021001, 2020.

[24] Arash Sioofy Khoojine, Mahdi Shadabfar, Vahid Reza Hosseini, and Hadi Kordestani. Network autoregressive model for the prediction of covid-19 considering the disease interaction in neighboring countries. *Entropy*, 23(10):1267, 2021.

[25] Alexey Strokach, David Becerra, Carles Corbi-Verge, Albert Perez-Riba, and Philip M Kim. Fast and flexible protein design using deep graph neural networks. *Cell systems*, 11(4):402–411, 2020.

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[27] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[28] Chenyu Wang, Zongyu Lin, Xiaochen Yang, Jiao Sun, Mingxuan Yue, and Cyrus Shahabi. Hagen: Homophily-aware graph convolutional recurrent network for crime forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4193–4200, 2022.

[29] Li Wang, Guannan Wang, Lei Gao, Xinyi Li, Shan Yu, Myungjin Kim, Yueying Wang, and Zhiling Gu. Spatiotemporal dynamics, nowcasting and forecasting of covid-19 in the united states. *arXiv preprint arXiv:2004.14103*, 2020.

[30] Lijing Wang, Aniruddha Adiga, Jiangzhuo Chen, Adam Sadilek, Srinivasan Venkatramanan, and Madhav Marathe. Causalgnn: Causal-based graph neural networks for spatio-temporal epidemic forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 12191–12199, 2022.

[31] Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[32] Oliver Wieder, Stefan Kohlbacher, Mélaine Kuenemann, Arthur Garon, Pierre Ducrot, Thomas Seidel, and Thierry Langer. A compact review of molecular property prediction with graph neural networks. *Drug Discovery Today: Technologies*, 37:1–12, 2020.

[33] Yuexin Wu, Yiming Yang, Hiroshi Nishiura, and Masaya Saitoh. Deep learning for epidemiological predictions. In *The 41st International ACM SIGIR Conference on Research Development in Information Retrieval*, SIGIR '18, page 1085–1088. Association for Computing Machinery, 2018.

[34] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 753–763, 2020.

[35] Yuteng Xiao, Hongsheng Yin, Yudong Zhang, Honggang Qi, Yundong Zhang, and Zhaoyang Liu. A dual-stage attention-based conv-lstm network for spatio-temporal correlation and multivariate time series prediction. *International Journal of Intelligent Systems*, 36(5):2036–2057, 2021.

[36] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

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

[38] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*, 2017.

[39] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[40] Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks. *Advances in neural information processing systems*, 31, 2018.

[41] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020.

[42] Difan Zou, Lingxiao Wang, Pan Xu, Jinghui Chen, Weitong Zhang, and Quanquan Gu. Epidemic model guided machine learning for covid-19 forecasts in the united states. *MedRxiv*, pages 2020–05, 2020.