A

Presented to
the faculty of the School of Engineering and Applied Science
University of Virginia

in partial fulfillment
of the requirements for the degree

by

APPROVAL SHEET

This

is submitted in partial fulfillment of the requirements
for the degree of

Author:

Advisor:

Advisor:

Committee Member:

Committee Member:

Committee Member:

Committee Member:

Committee Member:

Committee Member:

Accepted for the School of Engineering and Applied Science:

Jennifer L. West, School of Engineering and Applied Science

Abstract

Infectious diseases, such as seasonal influenza, Zika, Ebola, and the ongoing COVID-19, can be spread, directly or indirectly, from one person to another leading to an outbreak, an epidemic, or a pandemic. Infectious diseases place a heavy social and economic burden on our society. Producing timely, well-informed, and reliable spatiotemporal forecasts of the epidemic dynamics can help inform policymakers on how to provision limited healthcare resources, develop effective interventions, rapidly control outbreaks, and ensure the safety of the general public.

Traditional approaches are mainly based on theory-based mechanistic models (e.g., an agent-based SEIR model) and statistical time series models (e.g., autoregressive models). Recent advances in deep learning have significantly improved the state of the art in computer vision, natural language processing, and many other fields. Although deep learning-based predictive models have gained increased prominence in epidemic forecasting, they are far from being well explored. One challenge is the lack of sufficient good-quality training data, particularly during new emerging epidemics. Another challenge is that existing models are seldom designed to consider both spatial and temporal correlations dynamically for capturing disease spread dynamics. A further challenge is that such models rarely consider epidemiological context as prior. Models in the aforementioned cases are prone to be overfitting and are unlikely to provide explanatory power for the underlying phenomena due to the black box nature. Given the challenges, my research focuses on deep learning-based methods that incorporate spatiotemporal features and theory-based mechanistic models for a better understanding of disease spreading and improving forecasting *accuracy* and *explainability*. The aims are 1) improving epidemic forecasting accuracy by proposing graph neural network-based frameworks that consider temporal and spatial signals using a novel large scale mobility dataset, 2) improving explainability and accuracy of deep learning-based forecasting models by combining deep learning models with theory-based mechanistic models to incorporate epidemiological context.

First, we proposed a mobility informed graph neural network-based framework to capture cross-location co-evolving disease dynamics for better spatiotemporal epidemic forecasting. The proposed frameworks leverage priors from domain knowledge and mobility data. The priors are employed to instruct the model learning with the aim to allow for easier interpretation of the model and forecasting results. We incorporated

large-scale aggregated spatiotemporal mobility data into graph neural networks. The proposed model provides a natural representation of disease and human mobility dynamics to develop spatially explicit forecasts thus leading to better forecasting accuracy.

Second, we proposed TDEFSI that works towards enhancing deep learning models with theory-based mechanistic models with the aim of providing accurate forecasts and gaining a mechanistic understanding from a learned model. TDEFSI combines deep learning models and mechanistic models in a sequential learning process. In TDEFSI, mechanistic models are used to generate context-specific synthetic training data and then deep neural networks are trained with that synthetic data. Accurate high geographical resolution forecasting was achieved by using high-performance computing simulations. Furthermore, the explainable power of the proposed framework was explored by what-if scenario analysis.

Third, we further proposed CausalGNN that uses a causal module to mutually provide and embed causal features to get epidemiological context. CausalGNN adopts a joint learning process that learns a latent space to combine the spatiotemporal and causal embeddings using graph-based non-linear transformations. The learned model employs a causal mechanistic model to provide epidemiological context thus leading to better forecasting accuracy and better understanding of the underlying phenomena. In addition, the learned model can generate meaningful disease model parameters leading to explainable forecasts.

Acknowledgements

I am most grateful to my advisors, Professor Madhav Marathe and Professor Jiangzhuo Chen, for their ongoing guidance and support. They are exceptionally knowledgeable and extremely professional. I consider myself incredibly fortunate to have them as my advisors. They always provided useful advice and perspective, forming me as a researcher.

I would like to thank my committee members Professor Anil Vullikanti, Professor Stephen Eubank, Professor Jundong Li, and Dr. Adam Sadilek for their helpful comments on the thesis and their valuable suggestions for my research.

I am also grateful for all my collaborators Professor Jin-Hee Cho, Dr. Dipanjan Ghosh, Dr. Maria Gonzalez Diaz, Dr. Ahmed Farahat, Dr. Mahbubul Alam, Dr. Chetan Gupta, Professor Milind Tambe, Dr. Xue Ben, Dr. Alok Talekar, Dr. Ashish Tendulkar, Benjamin Hurt, Akhil Peddireddy, Dr. Arindam Fadikar, Dr. Ting Hua, Professor Yue Ning and many more whose names are not included here (in no particular order), for the many inspiring research conversations.

My research work is supported by Network Systems Science and Advanced Computing (NSSAC), a division of the Biocomplexity Institute (BI) at UVA. I would like to thank members of BI, Professor Christopher L. Barrett, Professor Achla Marathe, Professor Bryan Lewis, Dr. Srinivasan Venkatramanan, Dr. Aniruddha Adiga, Erin Raymond, Lisa Shifflett, Kimberly Lyman, and many others not listed here, for their thoughtful comments and suggestions related to epidemic modeling and interdisciplinary supports, useful discussion and suggestions on my research, and their ongoing support and encouragement on my life.

I am also grateful to UVA for the support and care I received from all of the faculty and staff, and in particular, from Professor Alfred Weaver, Professor Kevin Skadron, Tyler Miller, and Richard Tanson, who provided immense advice, support, and encouragement to me during my Ph.D. study, making my study at UVA a joyful journey.

I would also like to thank Virginia Tech for becoming my first home in a foreign country, for the wonderful friends I made, and for the support and care I received from all of the faculty and staff, in particular, from Professor Ing-Ray Chen, Professor Chang-Tien Lu and Roxanne Paul.

Lastly, my gratitude goes to my parents Shihe Wang and Xiaoqin Lan, my husband

Yupeng Sun, my little cute son Lucas Sun, who always believed in me, encouraged me, and supported me unconditionally, and to all my friends who have been there for me throughout this journey.

The work in this dissertation was partially supported by NSF Expeditions in Computing Grant CCF-1918656, CCF-1917819, US Centers for Disease Control and Prevention 75D30119C05935, DTRA subcontract/ARA S-D00189-15-TO-01-UVA, and National Institutes of Health (NIH) Grant 1R01GM109718.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, which is based on Wang et al. (2021a), we will present an overview of epidemic forecasting in Section 1.1, which includes introducing infectious disease and disease spreading dynamics in Section 1.1.1, discussing epidemic forecasting and its problems, challenges, and evaluation methods in 1.1.2, and providing a brief overview of theory-based mechanistic methods, statistical time series methods, and deep learning methods for epidemic forecasting in 1.1.3. A more specific discussion of related works will be made in each separate chapter.

1.1 An Overview of Epidemic Forecasting

1.1.1 Infectious Disease and Its Spread Dynamics

Human infectious diseases are caused by pathogenic microorganisms, such as bacteria, viruses, parasites, or fungi; and can spread directly or indirectly, from one person to another. An infectious disease spread can lead to an outbreak, an epidemic, or a pandemic. While most of the chapter will be focused on diseases that can be spread directly through human contact, similar methods can be developed for environment- or vector-mediated spread.

An *outbreak* can be defined as a sudden emergence of a localized cluster of disease occurrences in a sub-population. While it usually starts in a small community or a geographical area, it may lead to case exportation to other regions or countries. It may last for a few days to weeks, or even for multiple months. Some outbreaks could be seasonal such as those caused by environmental or vector abundance-based risk factors (e.g., Lyme disease). Others could be caused by exposures to zoonotic reservoirs (e.g., Ebola) or due to incidence in under vaccinated clusters (e.g., Measles). If not quickly controlled, an outbreak can become an epidemic causing significant health burden.

An *epidemic* occurs when an infectious disease spreads rapidly to many people within a community, population, or region. While they share several characteristics of

an outbreak, the spatial, temporal, and social scales are usually larger in magnitude. For example, in 2003, the severe acute respiratory syndrome (SARS) epidemic spread to about 8000 confirmed cases and led to nearly 800 deaths. Likewise, the Ebola epidemic ravaged West Africa between 2014 and 2016, with 28,600 reported cases and 11,325 deaths.

A *pandemic* is an epidemic that spreads over multiple countries or continents and could last multiple years. For instance, the influenza (flu) pandemic of 1918-1919 killed between 20 and 40 million people. While more devastating pandemics have been recorded (e.g., Bubonic Plague in the 14th century), the 1918 pandemic remains the most severe in recent history. The 2009 H1N1 influenza was a more recent global pandemic that led to an estimated 151K to 575K deaths worldwide during the first year the virus circulated. Since the 2009 H1N1 pandemic, the H1N1 flu virus along with other types has circulated seasonally in the U.S. causing significant illnesses, hospitalizations, and deaths. As of this writing, the ongoing COVID-19 pandemic is the most acute public health emergency since the 1918 influenza pandemic. As of April 2021, it has accounted for nearly 140 million reported cases and resulted in at least 3 million deaths worldwide[1].

While these distinctions help in characterizing the scale, they also reflect the difficult in obtaining data and the various factors involved in the dynamics. For instance, the control measures may vary between these scales, and hence such adaptations might make the task of forecasting more challenging. For the purposes of this thesis, we will mainly focus on epidemics, although the techniques outlined herein can be used interchangeably in many different scenarios.

The dynamics of an epidemic are usually characterized by: 1) when and where it started, 2) the scope and pervasiveness, 3) the duration of spread, and 4) overall severity (how it impacts individuals, communities, countries, and the whole society). For example, the 2014-2016 West Africa Ebola epidemic was the largest Ebola outbreak in history since the virus was first discovered in 1976. The World Health Organization (WHO) reported cases of Ebola Virus Disease (EVD) in the forested rural region of southeastern Guinea on March 23, 2014. It spread between countries, starting in Guinea then moving across land borders to Sierra Leone and Liberia. The average EVD case fatality rate was around 50% and has varied from 25% to 90% in past outbreaks[2]. In Guinea, Liberia, and Sierra Leone, the Ebola epidemic resulted in devastating effects on the healthcare workforce, the provision of healthcare services, children, and the national economy[3].

Numerous factors affect the disease spreading dynamics of an epidemic. Human factors such as activity (mobility, daily activities, mixing patterns) and demographics (age, gender, social status, economic status, etc.) are crucial because they determine

[1]Source: https://covid19.who.int/

[2]Source: `https://www.who.int/health-topics/ebola/#tab=tab_1`

[3]Source: `https://www.cdc.gov/vhf/ebola/history/2014-2016-outbreak/index.html`

how the disease transmits in a community. Environmental factors such as sanitation facilities, water supply, food, and climate account for an estimated 24% of the global disease burden and 23% of all deaths (by WHO), which includes epidemics and sporadic outbreaks. Public health interventions are the most effective way to control the disease spread. These interventions, both pharmaceutical (prophylactics, antivirals, vaccines) and non-pharmaceutical (stay-at-home orders, mask wearing, social distancing, safe burials) could be targeted at altering the spread dynamics. For example, during the ongoing COVID-19 pandemic, the virus is thought to spread mainly through close contact from person to person. Older adults and people of any age who have certain underlying medical conditions might be at higher risk for severe illness from COVID-19[4]. Certain jobs such as healthcare providers, school teachers, and supermarket workers are at higher risk of getting infected. The governments across the world had to rely on behavioral interventions (such as social distancing, wearing face masks in public, hand washing, monitoring and self-isolation for people exposed or symptomatic, etc.) at the beginning phase of the pandemic. With the development of multiple high efficacy vaccines that are authorized for emergency use, current measures include a combination of NPIs and vaccinations to drive down case rates, along with test/trace/isolation-based infection control.

1.1.2 Epidemic Forecasting

Infectious diseases have placed a heavy social and economic burden on our society. Producing timely, well-informed, and reliable spatiotemporal forecasts of the epidemic dynamics can help inform policymakers on how to provision limited healthcare resources, develop effective interventions, rapidly control outbreaks, and ensure the safety of the general public. In this section, we will first show an example of epidemic forecasting. Then we introduce the reference data used for forecasting, followed by a description of spatial and temporal epidemic forecasting. Next, we discuss the challenges of epidemic forecasting. Finally, we briefly introduce the epidemic forecasting metrics for evaluation.

Flu Forecasting - An Example of Epidemic Forecasting

A general idea of epidemic forecasting is to use observed data sources as the reference data to make spatial and temporal forecasts of an epidemiological target. For instance, take the "Predict the Influenza Season Challenge" – a flu forecasting project hosted by the Centers for Disease Control and Prevention (CDC) as an example. CDC's efforts with seasonal influenza forecasting began in 2013 with a competition that encouraged outside academic and private industry researchers to forecast the timing,

[4]Source: `https://www.cdc.gov/coronavirus/2019-ncov/need-extra-precautions/index.html`

peak, and intensity of the flu season, along with its short-term trajectory. The CDC provides data, relevant public health forecasting targets, and forecast accuracy metrics evaluated against actual flu activity. Each forecasting team submits their forecasts based on a variety of methods and data sources each week. The CDC has provided the Outpatient Illness Surveillance[5] report weekly at US national and HHS region[6] level since 1997 and at the state level since 2010 in FluView[7]. The historical records collected information on health care providers and patient visits for influenza-like illness (ILI). The CDC surveillance data is one type of reference data used to make forecasts, while researchers have used other datasets such as Google Flu Trends (GFT), Google Trends, twitter data, weather data to improve forecast accuracy. The epidemiological targets being forecast included season onset, peak week, peak intensity, and short-term activity. These target definitions rely on the percent of visits to health-care providers that are for ILI, also called ILI intensity. For instance: season onset is defined as the first week when ILI intensity is at or above baseline and remains there for at least two more weeks; peak week denotes the week when ILI intensity is the highest for the whole season; peak intensity is the highest value of ILI intensity during the season; short-term ILI activity means ILI intensity of one, two, three, and four weeks ahead of the date that they are available in FluView[8].

Researchers need to provide weekly forecasts at national, HHS region, and state level. These are provided in a probabilistic format, thus allowing uncertainty quantification. In recent times, there have been concerted efforts to build trained ensembles of these multiple methods to provide better forecasts. This effort has led to multi-team, multi-year collaborations (Reich et al., 2019) and has become increasingly prominent in public health communication and decision-making during influenza seasons. Each week during the influenza season, the CDC now displays the forecasts received through the Epidemic Prediction Initiative (EPI)[9]. Chakraborty et al. (2018) presented a set of considerations for flu forecasters to take into account prior to applying forecasting algorithms.

Reference Data

Reference data is extremely important in epidemic forecasting because it provides meaningful information about the disease spreading dynamics. In general, one could use various derived metrics, but the most common reference data is the traditional *surveillance data* which capture some measure of disease incidence for a given region

[5]Source: `https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html`

[6]The 10 Regional Offices that directly serve state and local organizations are hosted by the Office of Intergovernmental and External Affairs. HHS denotes U.S. Department of Human and Health Services.

[7]`https://www.cdc.gov/flu/weekly/fluviewinteractive.htm`

[8]`https://www.cdc.gov/flu/weekly/flusight/how-flu-forecasting.htm`

[9]`https://predict.cdc.gov/`

over a particular time period. Some examples include cases, hospitalizations, and deaths on a daily resolution at the county level for COVID-19. Usually such data is generated by regularly collecting reports from local public health laboratories and health care providers and then aggregating the collected information to form spatial and temporal data streams. The surveillance data is simply another indicator of total disease burden and could have various lead or lag time with respect to policy goals and actions. For a well observed disease, it is usually stable and reliable and is thus used as the main reference data. However, it is delayed due to the surveillance and reporting process, and may not be at a high resolution.

Another important reference data is the *mobility data*. Infectious diseases transmit directly or indirectly from person to person via contact networks. A contact network is formed when individuals come in geographic proximity to each other for a reasonable time duration. Human mobility behavior determines the formation of a contact network thus is crucial for modeling disease spreading dynamics especially for models based on social contact networks, which will be introduced in Section 2.4. For example, at an aggregate level, a region's COVID-19 dynamics can potentially be affected by regions where frequent travel occurs between them. Human mobility can be modeled or estimated using mathematical models such as the gravity model[10], or real world collected data, such as aggregate mobile phone data, air traffic data, commute data, etc.

Finally, *social media data* is often used as auxiliary information when making forecasts. Social media data is the collected information from social networks that show how users share, view, or engage with the epidemiological information, including behaviors such as searching, tweeting, or engaging in participatory surveillance (i.e., filling out surveys). There are many types of social data, such as tweets from Twitter and posts on Facebook. These data can be updated daily at finer geographic resolution but are not representative of the overall population. Furthermore, it requires large scale data collection and curation efforts.

Other reference data pertaining to environment and policy are also used for epidemic forecasting since they can provide reference information on any of the factors discussed in Section 1.1.1.

Temporal and Spatial Forecasting

Temporal forecasting. In the first two decades of the 21st century, multiple public health emergencies have occurred globally, highlighting the need to understand real-time epidemic science. During these emergencies, diseases cause rapid spread within a community and invade new regions in the span of just a few weeks to months, leaving a critical window of opportunity during which real-time warning is crucial.

[10]Wikipedia: A gravity model provides an estimate of the volume of flows of, for example, goods, services, or people between two or more locations. This could be the movement of people between cities or the volume of trade between countries.

Real-time forecasting is a type of forecasting that occurs concurrently to an event, such as an epidemic, using the most recent data available. Note that this does not mean forecasting only into the future, since in some cases, like seasonal influenza, the latest data available might be lagged by 1-2 weeks, thus requiring *forecasting* to the present (aka *nowcast* or even of the past (aka *hindcast*).

Retrospective forecasting is conducted by sequentially removing the data in the latest time from the full data set with the aim of evaluating and improving a model's forecasting performance retrospectively. Further, for an ongoing epidemic, the methods are often refined in real-time, and hence it is valuable to evaluate them across past data to check their out-of-sample performance.

The forecasting target as discussed above, could be short-term or long-term. While one could make forecasts of any incidence metric (say, number of K-12 outbreaks that will occur in the next 3 months), it is often useful to look at the aggregate epidemic trajectory (time series of number of cases, for example) and forecast its short-term trend and long-term characteristics. Given the process of data collection and surveillance lag, accurate statistics for epidemic warning systems are often delayed by some time, making long-term forecasting imperative without sacrificing on forecast performance.

There is no clear definition on what is considered short-term and long-term forecasting. *Short-term forecasting* typically refers to forecasting anywhere from one to six weeks ahead while *long-term forecasting* is usually used to predict the long term objectives such as time of peak, peak intensity, total number of deaths, etc.

With respect to the temporal forecasting, I'd like to introduce a commonly used definition *lead time* or *horizon*. Lead time or horizon in epidemic forecasting domain is the latency between the forecast of the epidemic dynamics (i.e., current time point) and its actual presentation (i.e., future time point). For example, if we are currently at time point t when making forecasts of an epidemiological target at time point $t + h$, then h is the horizon value.

Spatial forecasting. It is well established that the aggregate characteristics of epidemic incidence are being driven by spatial aspects of transmission. Thus, accurate forecast of the spatial spread of a disease could provide valuable insights into epidemic control. Spatial epidemic forecasting can be done at multiple geographical resolutions such as national, state/province, county, and city depending on forecasting models as well as the resolution of the available data. In this thesis, we adopt terminologies defined in Wang et al. (2020b) that " *flat-resolution* forecasting to denote the forecast of an epidemiological target with the same resolution as the reference data, *high-resolution* forecasting to denote the forecast with a higher geographical resolution than provided in reference data, and *coarse-resolution* forecasting to denote the forecast with a coarser geographical resolution than provided in reference data." Purely data-driven models can make flat-resolution forecasting. A coarse-resolution forecast can be obtained by (a) aggregating the flat-resolution forecasts into a coarse-resolution

based on their geographical attributes, or (b) aggregating the reference data to coarse resolution and running through the same forecasting methods. For theory-based mathematical models, forecasting can be made at any resolution depending on how detailed a computational model is, as long as they encode the resolution at which the forecasts are required. For example, individual level forecasts can be made if an agent-based model is used in epidemic simulations. Then the individual level incidence can be aggregated to the case count at any resolution based on the geographical location of each individual in the simulation.

Challenges of Epidemic Forecasting

Challenges with reference data. For recurring epidemics, such as seasonal influenza, the surveillance data could be at a coarse resolution and delayed in time. Other reference data is not as reliable and stable and requires extra data collecting and refining efforts. During an emerging epidemic, the forecasting problem could be particularly complicated as the training data 1) is sparse for each region (unlike seasonal flu there is no historical data); 2) noisy due to reporting bias, testing prevalence, etc.; 3) is a resultant of rapidly co-evolving dynamics of individual behavioral adaptations, government policies, and disease spread. Further such reference data could be retro-updated (referred to in the field as *backfill*) or change definitions mid-way. Difficulty obtaining real-time, reliable, and finer resolution information on disease dynamics have limited the predicting power of existing infectious disease forecasting techniques which heavily rely on this information.

Challenges with spatial and temporal forecasting. Designing a model that can capture both spatial and temporal patterns from data is crucial yet challenging. First, real-time forecasting is challenging for systems that are compute- and data-intensive due to the need for regular and frequent updates. Thus, a model with less computational cost is more suited for real-time forecasting systems, as long as it does not sacrifice much in terms of overall accuracy. Second, a challenge in long-term epidemic forecasting is that the temporal dependency is hard to capture with short-term input data. Particularly, limited availability of reference data during emerging epidemics has resulted in failure to capture long-term patterns from the data. Models that can capture short- and long-term patterns from limited input data are required for accurate long-term forecasting. Third, as spatial data becomes available, the influence from other locations should be explored while making forecasting. However, it is difficult to investigate data from systems with models that do not represent space in some way. Models considering cross-location signals can capture spatial patterns from the data, which can lead to better forecasting performance, but could be impacted by model mis-specification biases (especially the level of connectivity). Finally, difficulty in accessing high-resolution data often fail the spatial forecasting at a finer resolution.

Epidemic Forecasting Evaluations

In statistics, *point estimation* involves the use of sample data to calculate a single value (known as a point estimate since it identifies a point in some parameter space) which is to serve as a "best guess" or "best estimate" of an unknown parameter. In infectious disease epidemiology, point forecasts are often served as the best guess of an unknown target. More often, *probabilistic forecast* is necessary to properly reflect forecasting uncertainty. It is an estimation of the distribution of an unknown target. For example, in the CDC FluSight Challenge (see Section 1.1.2), for peak week forecasting, the point forecast could be the week the peak is most likely to occur during the current flu season and the probabilistic forecast is the probabilities that the peak will occur on each week during the season (e.g., 50% peak will occur on week 1; 30% chance on week 2; 20% chance on week 3).

Evaluation of forecasting performance is crucial for model improvement (Tabataba et al., 2017a). Popular metrics for evaluating point forecasts in epidemic forecasting are: (1) Mean Absolute Error (MAE), (2) Mean Squared Error (MSE), (3) Root Mean Squared Error (RMSE), (4) Mean Absolute Percentage Error (MAPE). Particularly, (5) Pearson Correlation (PCORR). PCORR is used to measure the model performance on predicting disease trends. Common scoring rules to evaluate full predictive distributions in epidemic forecasting are: (1) Logarithmic Score (logS) (Gneiting and Raftery, 2007) and its variation multibin logS (MBlogS) (Centers for Disease Control and Prevention, 2018), (2) Continuous Ranked Probability Score (CRPS) (Gneiting et al., 2007). The logS and MBlogS are used for scoring flu forecasting in the FluSight Challenge. In addition, (3) Interval Score (IS) (Gneiting and Raftery, 2007) and (4) Weighted Interval Score (WIS) (Gneiting and Raftery, 2007) are designed specifically for forecasts in a quantile/interval format. The IS has recently been used to evaluate forecasts of Severe Acute Respiratory Syndrome Coronavirus 1 (SARS-CoV-1) and Ebola (Chowell et al., 2019) as well as SARS-CoV-2 (COVID-19) (Bracher et al., 2021). WIS has been recently widely used for COVID-19 forecasting evaluation (Bracher et al., 2021). Among these metrics, PCORR ranges in $[-1, +1]$ that larger values are better; MAE, MSE, RMSE, and MAPE range in $[0, +\infty]$ that smaller values are better; logS and MBlogS range in $[-\infty, +\infty]$ that larger values are better; CRPS, IS, and WIS are negatively oriented so that smaller values are better.

1.1.3 Methodologies for Epidemic Forecasting

Forecasting the spatial and temporal evolution of infectious disease epidemics has been an area of active research over the past couple of decades. The existing works rely on theory-based mechanistic methods and data-driven methods for forecasting. Note that in this thesis, theory-based mechanistic methods or causal methods refer to forecasting methods employing epidemic models for simulating disease transmission processes between individuals. Data-driven methods refer to forecasting methods

employing statistical and time series-based methodologies without causal mechanism. In this section, we will briefly introduce these methodologies and some important related works. Refer to (Reich et al., 2019; Zhang et al., 2013; Nsoesie et al., 2014a; Chowell et al., 2016; Philemon et al., 2019; Adiga et al., 2020a; Zeroual et al., 2020) for more related works. A more specific review of related works will be presented in each chapter.

Theory-based Mechanistic Methods

The theory of epidemic spread is inherently tied to the progression of disease within an individual, and the processes of transmission between individuals. Historically, in epidemiology, within host disease progression has been encapsulated into compartmental models for embedding into population models. For instance, individuals get assigned various disease states such as susceptible, infectious, recovered, etc. and the host-pathogen characteristics are used to define the durations and likelihoods of various transitions in this finite state machine. These models have various extensions depending upon the disease being studied (e.g., existence of a latent phase, infectiousness after death) and the interventions being employed (e.g., hospitalization, treatments, vaccinations). See (Bailey et al., 1975; Kuznetsov and Piccardi, 1994; Lofgren et al., 2014) for some examples. Finally, to capture the transmission process, some representation of social/environmental contact is expressed in terms of mixing assumptions, contact networks, etc.

Forecasting methods employing these models are called mechanistic methods (or causal methods) because they are based on the causal mechanisms of infectious diseases. At a fairly high level, the underlying epidemic model can be either a compartmental model (CM) (Flahault et al., 2006; Lee et al., 2012; Lunelli et al., 2009) or an agent-based model (ABM) (Parker and Epstein, 2011; Chao et al., 2010; Tabataba et al., 2017b). In a compartmental model, a population is divided into compartments (e.g., S, E, I, R) and no distinction is made among individuals within a compartment. Further the entire population is assumed to be homogeneously mixing, a simplifying assumption, but reasonable for capturing large-scale dynamics. A differential equation system characterizes the change of the sizes of each compartment due to disease propagation and progression. Depending on the underlying assumptions, the rate of contact could be density-dependent or frequency-dependent (Keeling and Rohani, 2011). This class of models can further be extended to the class of meta-population models where spatial connectivity is explicitly accounted for, and the disease compartments are tracked per spatial region. While these models, inspired from population ecology, are widely used in understanding human disease dynamics, they suffer from lack of fidelity to represent essential structures such as households, schools, etc. which may play a role in disease spread dynamics and control. See (Ball et al., 2015) for an overview of challenges in using such models for disease dynamics.

In an agent-based model, disease spreads among heterogeneous agents through an

unstructured network (Eubank et al., 2004). Dynamics with individual behavior change exhibit significant impact on epidemic and dynamic forecast models (Eksin et al., 2019), which can be implemented using a high-performance computing model (Bisset et al., 2009). The individual level details in an agent-based model can be easily aggregated to obtain epidemic data of any resolution, e.g., number of newly infected people in a county in a specific week. Such models have been used extensively to study diseases in significant detail, including Ebola (Venkatramanan et al., 2018), Influenza (Nsoesie et al., 2013b), and more recently COVID-19 (Hoertel et al., 2020; Talekar et al., 2020). There are multiple ongoing efforts to understand the relationship between these different class of models (see (Ajelli et al., 2010), for example). Many forecasting methods have been developed based on either CM or ABM (Tuite et al., 2010; Shaman and Karspeck, 2012; Nsoesie et al., 2013a; Yang et al., 2014, 2015a; Zhao et al., 2015a; Morita et al., 2018). Taking seasonal or pandemic influenza as an example, we list a few notable exercises that have used either of these approaches for forecasting or allied tasks. Shaman and Karspeck (2012) developed a framework for initializing real-time forecasts of seasonal influenza outbreaks, using a data assimilation technique commonly applied in numerical weather prediction. Tuite et al. (2010) used an SIR CM to estimate parameters and morbidity in pandemic H1N1. Yang et al. (2014) applied various filter methods to model and forecast influenza activity using an SIRS CM. Nsoesie et al. (2013a) proposed a simulation optimization approach based on the SEIR ABM for epidemic forecasting. Venkatramanan et al. (2021) factored mobility map into a metapopulation SEIR model to retrospectively forecast influenza in the USA and Australia. The COVID-19 Scenario Modeling Hub[11] convened six modeling teams (including both CM and ABM methods) in an open call to provide long-term, 6-month (April–September 2021) COVID-19 projections in the United States (Borchering et al., 2021). Causal methods are generally computationally expensive as they require the parameter estimation over a high dimensional space. As a result, the use of such methods for real-time forecasting is challenging. Furthermore, forecasting performance depends on the assumed underlying disease models.

Statistical Time Series Methods

Statistical time series methods are data-driven methods that employ statistical and time series-based methodologies to learn patterns in historical epidemic data and leverage those patterns for forecasting.

These methods assume that the observed data is the outcome of a random process with an unknown probability distribution, typically a parametric distribution. In addition, the observed data is considered to be a function of explanatory variables or covariates which enables inference of distribution parameters through a likelihood function. A popular class of models are the autoregressive models (e.g., AR, ARMA,

[11] `https://covid19scenariomodelinghub.org/`

ARIMA) that assume that the observed time series current time step can be expressed as a linear combination of past samples and error terms. In the context of ILI forecasting, in addition to the ARIMA terms of the ILI time series, other exogenous variables such as search trends, social media data, weather data, etc. can be used as exogenous regressors to enhance nowcast performance.Yang et al. (2015b), Rangarajan et al. (2019), Kandula et al. (2017), Soebiyanto et al. (2010), and Paul et al. (2014) assumed a Gaussian distribution on the data when modeling ILI rates and activity level. AR models for count data are modeled using Poisson (Wang et al., 2015) and negative binomial distribution (Dugas et al., 2013; Radin et al., 2020) and result in a class of generalized linear models. Owing to a large number of explanatory variables, techniques such as LASSO (Tibshirani, 1996), log-likelihood ratio test (Rangarajan et al., 2019), and block coordinate descent methods (Tseng, 2001) are employed to select a sparse subset of most relevant variables. In the presence of sufficient seasonal data, a Bayesian weighted average of trajectories from past seasons to model current season assuming a mixture of Gaussian models is shown to perform reasonably well in the case of influenza (Viboud et al., 2003; Brooks et al., 2015) and dengue (Van Panhuis et al., 2014). Under non-parametric models, method of analogues which attempts to find the most relevant historical segments of data or nearest neighbors with respect to the observed data and use a weighted average of the nearest neighbors to produce forecasts (Viboud et al., 2003). Dirichlet process model was explored to match the current influenza activity to simulated and historical patterns, identify epidemic curves different from those observed in the past, and enable forecasts of the expected epidemic peak time (Nsoesie et al., 2014b). Exponential smoothing is another class of non-parametric regression models which employ exponentially decaying weights on historical samples (e.g., Petropoulos and Makridakis (2020)).

Statistical methods rely on the stationarity assumptions of the data. To some extent, the nonstationarity of time series data addressed through differencing and retraining over short observation windows but performance of statistical methods is inversely related to deviation from historically observed distribution. As observed in the case of COVID-19 forecasting, most statistical methods were employed during the initial phase of the pandemic to capture exponential growth phase, but with the pandemic undergoing rapid fluctuations, these methods were not effective in accurate forecasting.

Deep Learning Methods

Deep neural networks (DNNs) have gained increasing prominence in epidemic forecasting due to their ability to learn non-linear relationship between the inputs and the outputs without prior domain knowledge. Some of the common structure of such networks include: feedforward neural networks (FNNs), recurrent neural networks (RNNs), convolutional neural networks (CNNs), and graph neural networks (GNNs). A feedforward neural network is an artificial neural network wherein connections

between the nodes do not form a cycle. It was the first and simplest type of artificial neural network devised (Schmidhuber, 2015). Forecasting prevalence of epidemics using feedforward neural networks is a widely accepted approach. For example, dengue forecasting (Wahyunggoro et al., 2013; Aburas et al., 2010), and FNNs were first applied for influenza forecasting (Xu et al., 2017). Adhikari et al. (2019) proposed EpiDeep for seasonal ILI forecasting by learning meaningful representations of incidence curves in a continuous feature space. The FNNs, due to their ability to inherently capture temporal dynamics, have become a natural choice for time series forecasting. Popular RNN modules are gated recurrent unit (GRU) (Chung et al., 2014) and long-short term memory (LSTM) (Hochreiter and Schmidhuber, 1997). Volkova et al. (2017) built an LSTM model for short-term ILI forecasting using CDC ILI and Twitter data. Venna et al. (2019) proposed an LSTM-based method that integrated the impacts of climatic factors and geographical proximity. Zhu et al. (2019) proposed attention-based LSTM model for epidemic forecasting. Chimmula and Zhang (2020) used LSTM networks to predict COVID-19 transmission. The CNNs are usually used to deal with image data with regular grid data structure. The idea is to sum the neighboring node features around a center node, specified by a filter with parameterized size and learnable weight. CNNs can be used for epidemic forecasting because multi-variate time series (e.g., spatial regions) of an epidemic can be treated as an image with regular grid. Wu et al. (2018) constructed CNNRNN-Res combining RNN and convolutional neural networks to fuse information from different sources. The GNNs are the generalized version of CNN that can work on data with non-regular structures like a graph. The basic idea is to generate node embeddings based on local network neighborhoods through message passing. The neighborhoods are defined using an adjacency matrix, which can be any type of relationship between graph nodes. GNNs are famous for their ability to capture cross-spatial effects in dynamic environments thus leading to an increased prominence in epidemic forecasting. Deng et al. (2020) designed cola-GNN which was a cross-location attention-based graph neural network for forecasting ILI. Regarding COVID-19 forecasting, Kapoor et al. (2020) and Wang et al. (2021b) examined GNNs for COVID-19 daily case forecasting using mobility data. Wang et al. (2020a) examined a wide range of deep learning models for forecasting COVID-19 weekly confirmed cases. Ramchandani et al. (2020) presented DeepCOVIDNet to compute equi-dimensional representations of multivariate time series.

Training deep learning models usually require a large training dataset which is usually not available particularly for novel and emerging epidemics. Another well-known limitation of deep learning methods is the lack of interpretability for model forecasts due to their black box nature. Furthermore, since they are purely data-driven, they do not explicitly incorporate the underlying causal mechanisms. As a result, epidemic dynamics affected by behavioral adaptations are usually hard to capture, even for mechanistic models. However, with additional data becoming available and the surveillance systems maturing, these models are becoming more promising.

Ensemble Methods

Ensemble modeling aims to boost the forecasting performance by systematically integrating the predictive accuracy across individual models. It is widely used for epidemic forecasting. Ren et al. (2016) presented a comprehensive review. Popular ensemble modeling techniques used in epidemiological domain include bagging (Breiman, 1996), boosting (Freund et al., 1996), Ensemble Kalman Filter (EnKF) (Burgers et al., 1998), and Bayesian Model Averaging (BMA) (Hoeting et al., 1999). Ensemble in deep learning also leverages dropout (Hinton et al., 2012) and snapshot ensemble (Huang et al., 2017; Wang et al., 2020c) techniques. Thomson et al. (2006) discussed a system to forecast probabilities of anomalously high and low malaria incidence with dynamically based, seasonal-timescale, multi-model ensemble forecasts of climate. Chakraborty et al. (2014) proposed ensemble modeling schemes with different parametric bootstrapping procedures. Recently, Chowell and Luo (2021) adopted a simple bootstrap ensemble method to make epidemic forecasting. Adiga et al. (2021) presented a COVID-19 forecasting pipeline which incorporated probabilistic forecasts from multiple statistical, machine learning and mechanistic methods through a Bayesian ensembling scheme, and had been operational for nearly 12 months serving local, state and federal policymakers in the United States.

Ensemble methods involve designing and implementing various forecasting models and often require extra computational cost for ensemble model training.

1.2 The Importance of Reliable Deep Learning-based Epidemic Forecasting Methods

In the previous section, we reviewed the methodologies for epidemic forecasting and briefly discussed the advantages and disadvantages of these methods. Among these methods, deep learning-based methods are relatively new emerging research directions which are far from well explored. Producing reliable spatial and temporal epidemic forecasting using deep learning models is crucial yet challenging. First, the training data is sparse, especially during new emerging epidemics; in addition, it's noisy due to reporting bias, testing prevalence and so on. Models trained with such data are prone to be overfitting thus reducing the forecasting accuracy. Second, the disease spreads over a contact network which depends on human mobility behavior. Thus, a GNN-based model that considers both spatial and temporal signals is potential to better capture the disease spread dynamics. Geographical adjacency is one of the common types of information used in GNNs, however, it fails to reflect the dynamic nature of human mobility. On the other hand, human mobility data is a promising way to provide dynamic spatiotemporal correlations. Third, deep learning models barely consider epidemiological context. As a result, they are prone to be overfitting and it is therefore difficult to explain the learned model and its forecasts. In the

epidemiological domain, we are not only interested in generating correct forecasts but also understanding the causal mechanism behind the forecast. An ideal model should provide *accurate* and *explainable* forecasts.

Accuracy–Policymakers use predictive models to make public health decisions, and more accurate model outcomes result in better decisions. The cost of errors can be immense, such as loss of financial resources or life during an epidemic; however, optimizing model accuracy can mitigate those cost. Recent advances in deep learning have significantly improved the state of the art in computer vision, natural language processing, and many other fields. Based on such advances, we are eager to explore their use and demonstrate their predictive power in epidemic forecasting tasks. We have discussed in Section 1.1 that the aggregate characteristics of epidemic dynamics are being driven by both temporal and spatial aspects of transmission. Thus, a model considering both spatial and temporal signals is potentially better able to capture the disease spread dynamics. In this thesis, we will explore GNN-based methods that leverage spatiotemporal signals in the aim of capturing dynamic disease spread patterns.

Explainability–Unlike theory-based mechanistic methods, deep learning models are purely data-driven, they do not explicitly incorporate the underlying causal mechanisms. As a result, epidemic dynamics affected by behavioral adaptations are usually hard to capture. When training a deep learning model for forecasting tasks, it is crucial to answer the following questions: If we can learn a model from data, can that model provide both correct inferences and also an explanation for the underlying phenomena? In the epidemiological domain, answering these questions is particularly important for better understanding and controlling of the disease spread. In this thesis, we will explore reliable spatial and temporal deep learning-based forecasting methods combining theory-based mechanistic models in the aim of gaining a mechanistic understanding from a learned model.

1.3 Research Questions and Outline

Given the background and challenges outlined previously, my research focuses on deep learning-based methods that incorporate spatiotemporal features and theory-based mechanistic models for a better understanding of disease spreading and improving forecasting accuracy and explainability. The aims are 1) improving forecasting accuracy by proposing deep learning models that consider temporal and spatial signals, and 2) improving explainability of deep learning models using theory-based mechanistic models. We investigate the following questions and answer the questions with novel methods and techniques.

Q1. How to leverage mobility information to provide spatial and temporal spreading context for deep learning-based epidemic forecasting?

The assumption is that spatial and temporal signals can reflect disease spread dynamics. When modeling infectious diseases mechanistically, it is useful to note that in the human population the spread is facilitated by social contacts, which are in turn influenced by the movement of individuals. Researchers have leveraged this fact and have used information on human mobility to predict and explain the dynamics of disease spread. Geographical adjacency failed to reflect the dynamic of human mobility. We proposed GNN-based frameworks to capture cross-location co-evolving disease dynamics caused by human mobility. The proposed frameworks leverage domain knowledge and mobility data to instruct the model constructing and learning with the aim being better able to interpret the model and forecasting results. We incorporated new large-scale aggregated spatiotemporal mobility data into GNNs. The mobility informed GNNs account for dynamic spatiotemporal signals leading to a better understanding of the forecasting results. These contributions will be presented in Chapter 2.

Q2. How to leverage theory-based mechanistic models to provide epidemiological context for deep learning-based epidemic forecasting?

Prior works of physics, biology, and epidemiology (Karpatne et al., 2017) have shown evidence that incorporating domain knowledge into data-driven models can improve spatiotemporal forecasting algorithms. However, existing deep learning models barely consider epidemiological context for epidemic forecasting. Such models are prone to be overfitting leading to failures in long-term forecasting, especially when the data is noisy and sparse such as COVID-19 surveillance data at the US county level. Theory-based mechanistic models can capture the diffusion patterns of disease spread through detailed simulations using various disease models thus can provide meaningful epidemiological context. We proposed frameworks that combine deep learning models with theory-based mechanistic models for better spatiotemporal forecasting. We first proposed the framework TDEFSI that trains a RNN-based model with theory generated synthetic data. Computing-based simulations were used to generate context specific training data. The model trained with such data can capture the underlying causal processes and mathematical theories leading to an ability to make context specific forecasts and capture the unique properties of a given region. Accurate high geographical resolution forecasting was also achieved by using this framework. We further proposed a novel learning framework CausalGNN that jointly learns a latent space to combine the spatiotemporal and causal embeddings using graph-based non-linear transformations. The learned model employs a causal mechanistic model to provide epidemiological context thus leading to better forecasting accuracy and better understanding of the underlying phenomena. These contributions will be presented in Chapter 3.

Chapter 2

Spatial and Temporal Epidemic Forecasting Using Graph Neural Networks

Disease dynamics, human mobility, and public policies co-evolve during pandemics such as COVID-19. Understanding dynamic human mobility changes and spatial interaction patterns are crucial for understanding and forecasting COVID-19 dynamics. In this chapter, which is based on Wang et al. (2021b), we will present one of the early works on the use of GNNs to forecast COVID-19 dynamics. The motivation and major contributions are introduced in Section 2.1. A brief overview of related work is presented in Section 2.2, including using mobility data to understand COVID-19 spread, forecasting COVID-19 dynamics, and spatiotemporal forecasting. In Section 2.3, we introduce a large-scale mobility dataset which will be used to build graphs as well as graph node features. In Section 2.4, we formulate the problem and present a GNN-based epidemic forecasting framework using mobility maps. The experiment settings and results are shown in Section 2.5. Finally, Section 2.6 concludes the chapter and discusses directions for future work.

2.1 Motivation

The COVID-19 pandemic is arguably the most acute public health emergency since the 1918 influenza pandemic. It has already infected over 198 million people and resulted in 4.2 million deaths across the globe[1]. The global economy contracted by 3.5 percent in 2020 according to the April 2021 World Economic Outlook Report published by the IMF, a 7 percent loss relative to the 3.4 percent growth forecast back in October 2019[2]. The pandemic has affected almost every country in the world and

[1]Source: `https://covid19.who.int/` as of August 2, 2021.

[2]Source: `https://www.brookings.edu/research/social-and-economic-impact-of-covid-19/`

has resulted in an unprecedented response by governments across the world to control its spread. Pharmaceutical interventions are not generally available at the stage when we start this work (with the exception of remdesivir under FDA's expanded access Rome and Avorn (2020)) and thus, countries have had to rely exclusively on behavioral interventions that involve some form of social distancing. The rapid spread of the pandemic has forced countries to institute strict social distancing measures. Each social distancing policy is characterized by: (*i*) when and how gradually it started, (*ii*) the length of time for which it was enforced, (*iii*) the scope and pervasiveness, and (*iv*) the stringency (total lockdown versus stay-at-home advisories).

The social distancing measures have led to significant change in human mobility that have in turn affected the disease dynamics. To better understand COVID-19 dynamics and help to control the disease spread, it is crucial and challenging to (*i*) evaluate the level of public response to the US state and county level restrictions and (*ii*) provide accurate and timely spatiotemporal forecasting of epidemic dynamics. Using aggregate mobility data to understand COVID-19 dynamics has recently become popular. There have been a number of recent studies along these lines, for example, in China using Baidu data (Chinazzi et al., 2020), in the US using mobility data (Kraemer et al., 2020b; Adiga et al., 2020c), and at a global scale using airline traffic (Adiga et al., 2020b). On the other hand, a number of COVID-19 forecasting methods have been proposed since the initial outbreak early 2020, such as mechanistic methods (Yang et al., 2020; Anastassopoulou et al., 2020; Kai et al., 2020), and time series methods using statistical regression models (Ribeiro et al., 2020) or deep learning models (Ramchandani et al., 2020). However, existing deep learning-based methods barely considered cross-location effects in long term disease propagation. As GNNs have been successful in many domains, we investigate their potential applicability to forecast infectious disease dynamics.

There are several challenges in spatiotemporal epidemic forecasting using GNNs. First, the temporal dependency is hard to capture with short-term (e.g., less than 5 time points) input data because long-term trends (e.g., seasonal trends) are usually not reflected in a short-time duration. Second, the disease spreads over a contact network which depends on human mobility behavior. Geographical adjacency failed to reflect the dynamic of human mobility. Dynamic spatial effects have not been exhaustively explored with limited data input. Spatiotemporal effects have been studies in (Senanayake et al., 2016; Li et al., 2017; Yu et al., 2017; Ning et al., 2018). However, they usually require adequate data sources to achieve decent performance in epidemic forecasting. In this work, we focus on GNN-based methods to solve the above challenges by leveraging a new large-scale aggregated spatiotemporal mobility data which will be introduced in Section 2.3. The major contributions are:

- We analyze the joint effects of social-distancing guidelines and mobility patterns at the US state levels using an integrated map of mobility flows (MF), COVID-19 Surveillance data, and data on social distancing guidelines;

- We design a dynamic mobility informed GNN that considers both temporal dynamics and cross-location co-evolution dynamics using a recurrent message passing (RMP) module to recurrently embed information from a node's neighbors;
- We also design multiple variants of the proposed model which use a static mobility graph, geographical adjacency graph, and attention-based trainable graph;
- We evaluate the proposed model on forecasting the US state level daily new cases and demonstrate that the dynamic spatial and temporal mobility informed GNN allows for better forecasting performance compared with its variants as well as several existing classic and state-of-the-art time series methods.

2.2 Related Work

Using mobility data to understand COVID-19 spread. Mobility data plays a central role in most studies related to the spread of COVID-19. Important sources of data used include: mobility data from Google, Apple, Cuebiq, Safegraph, Descartes Labs and X-mode (UMD, 2020; Lasry et al., 2020; Apple, 2020; Gao et al., 2020; Klein et al., 2020). During the initial phase of the outbreak much of the analysis involved the use of airline data to determine the global spread of the virus and case importations (Bogoch et al., 2020; Chinazzi et al., 2020; Adiga et al., 2020b). Later phases saw the use of data to model disease dynamics (Kraemer et al., 2020b; Liu et al., 2020; Lai et al., 2020), counterfactual analysis and forecasting, see (Bassolas et al., 2019; Wellenius et al., 2020; Gao et al., 2020; Adiga et al., 2020c) for examples of such analysis. The data used in this work is *unique* and *has not been used for COVID-19 analysis until now*. The detailed description of the dataset will be presented in Section 2.3.

Forecasting COVID-19 dynamics. Researchers have used mechanistic models, time series models and deep learning models for COVID-19 forecasting and in general, this is a highly active area of research. Mechanistic methods have been a mainstay for COVID-19 forecasting due to their ability to represent the underlying disease transmission dynamics as well as incorporating diverse interventions. They enable counterfactual forecasting which is important for future government interventions to control the spread. Forecasting performance depends on the assumed underlying disease model. See (Yang et al., 2020; Anastassopoulou et al., 2020; Giordano et al., 2020; Kai et al., 2020; Yamana et al., 2020; Talekar et al., 2020) for examples of such approaches. As additional data becomes available and the surveillance systems mature, data-driven models, including statistical time series models (Harvey and Kattuman, 2020; Petropoulos and Makridakis, 2020; Ribeiro et al., 2020) and deep learning models (Hu et al., 2020; Chimmula and Zhang, 2020; Arora et al., 2020;

Magri and Doan, 2020; Dandekar and Barbastathis, 2020; Kapoor et al., 2020; Wang et al., 2020a; Ramchandani et al., 2020), are becoming more promising for COVID-19 forecasting. Among these works, Kapoor et al. (2020) was the first trying of applying GNNs with mobility data. However, only one day ahead forecasting was examined and there was no comparison with the state of the art. To the best of our knowledge, our model is among the first significant GNN-based models that incorporate a large-scale mobility dataset for forecasting COVID-19 dynamics.

Spatiotemporal forecasting. With the increasing growth of spatiotemporal data, mining valuable knowledge from spatiotemporal data is essential for many real-world applications In the forecasting of social events (Zhao et al., 2015b; Ning et al., 2018), text data such as news articles and tweets are often used as features, which is usually a weak auxiliary feature for epidemic forecasting as some epidemics, e.g., influenza has historically, and continues to, occur periodically at the population level. Collecting and processing relevant external data such as news or tweets is also expensive. In recent studies of air quality forecasting (Li et al., 2016; Gao et al., 2019) and traffic forecasting (Li et al., 2017; Wu et al., 2019; Lai et al., 2018; Yu et al., 2017), researchers have modeled spatiotemporal dependence between different sensors by integrating graph convolutional networks (GCNs) into RNNs or CNNs. However, data sampling for epidemic data is different than air or traffic data. For instance, traffic sensors transmit data at 5-minute intervals. COVID-19 data collection usually shows a larger granularity (e.g., days) with a delay. Traffic forecasting models tend to overfit in epidemic data when the model complexity is relatively high. It is of great significance to introduce an effective model for spatiotemporal epidemic forecasting given limited data. In this work, we design a flexible GNN-based model architecture to account for a variety of static and dynamic spatiotemporal signals. Our model can better understand the forecasting results by incorporating real-time aggregated spaiotemporal mobility maps and can be easily extended to forecast other disease dynamics.

2.3 A Large-Scale Mobility Dataset

2.3.1 Data Description

The **Google COVID-19 Aggregated Mobility Research Dataset** (Kraemer et al., 2020a) contains anonymized mobility flows aggregated over users who have turned on the Location History setting (in the default settings this is turned off). This is similar to the data used to show how busy certain types of places are in Google Maps — helping identify when a local business tends to be the most crowded. The dataset aggregates flows of people from region to region, which is here further aggregated at multiple geographical resolutions weekly. Figure 2.1 presents a snapshot of multiple scales of mobility maps. It shows progressively the mobility volume at various S2 cell

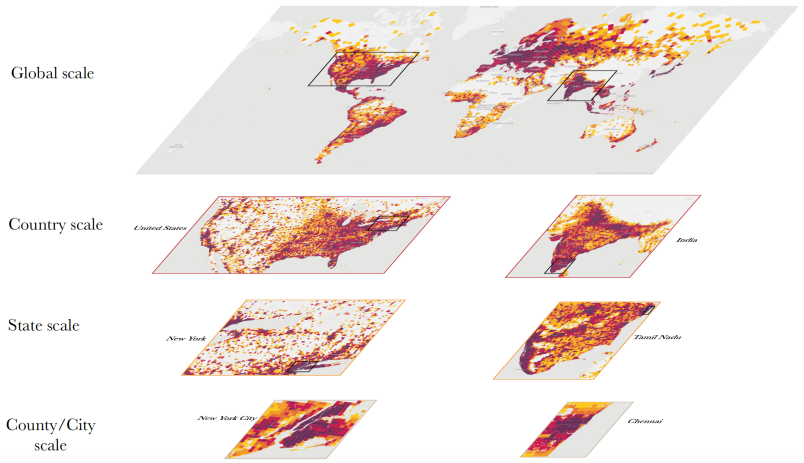


Figure 2.1: Multiple scales of mobility: mobility flows between 5 km^2 grid cells are aggregated to appropriate spatial resolution for the analyses. The figure shows progressively the mobility volume at various S2 cell levels (L12 to L6) and geographical scales (county, state, country, and globe).

levels (L12 to L6) and geographical scales (county, state, country, and globe).

To produce this dataset, machine learning is applied to logs data to automatically segment it into semantic trips (Bassolas et al., 2019). To provide strong privacy guarantees, all trips were anonymized and aggregated using a differentially private mechanism (Wilson et al., 2019) to aggregate flows over time. This research is done on the resulting heavily aggregated and differential private data. No individual user data was ever manually inspected, only heavily aggregated flows of large populations were handled. All anonymized trips are processed in aggregate to extract their origin and destination location and time. For example, if users traveled from location i to location j within time interval t.

2.3.2 Exploratory Data Analysis

Quantifying social distancing. We construct a mobility map G^{MF} from G by weighting the edge $e_{ij} \in \mathcal{E}$ with $f_{ij}(t)$ which represents the MF from v_i to v_j during week t. We choose a weekly scale because of the mobility flow data resolution. In order to quantify the effects of social distancing through changes in mobility, we introduce two metrics namely the Flow Reduction Rate (FRR) and Social Distancing Index (SDI) computing using G^{MF} . While FRR measures the reduction in connectivity of a region to the outside world, SDI measures the change in mixing within the region.

Flow Reduction Rate (FRR): It measures the impact of social distancing by comparing the levels of connectivity before and after the stay-at-home orders. Given a region v_i, we first compute the average outflows during the pre-pandemic period

for $v_i \in \mathcal{V}$ as $\bar{f}_i = \frac{1}{|T_p|} \sum_{t_p \in T_p} \sum_{j \in \mathcal{V}} f_{ij}(t_p)$ over first T_p weeks of year 2020. $\text{FRR}_i(t)$ is then defined as

$$\text{FRR}_i(t) = \frac{\sum_{j \in \mathcal{V}} f_{ij}(t) - \bar{f}_i}{\bar{f}_i}. \tag{2.1}$$

This defines a unit-less relative change in outflows from node v_i for any given week t with respect to $\bar{f}_i$. Henceforth, we omit i and t in the notation.

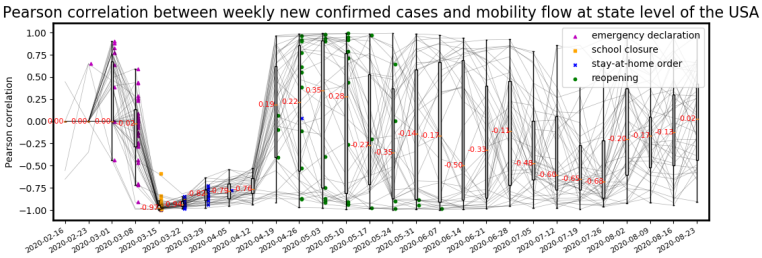
Social Distancing Index (SDI): It quantifies the mixing or movement within a county, we consider the MF between the 5 km^2 cells in it. Let $\boldsymbol{F}(t)$ denote the normalized flow matrix of the county at week t where $F_{ij}(t) = \frac{f_{ij}(t)}{\sum_{j \in \mathcal{V}} f_{ij}(t)}$, we compare $\boldsymbol{F}(t)$ to the uniform matrix $\boldsymbol{U}$ and the identity matrix $\boldsymbol{I}$. The SDI quantifies the closeness of $\boldsymbol{F}(t)$ to $\boldsymbol{U}$ and $\boldsymbol{I}$ and is defined as

$$\text{SDI}(t) = \frac{||\boldsymbol{F}(t) - \boldsymbol{U}||_2}{||\boldsymbol{F}(t) - \boldsymbol{U}||_2 + ||\boldsymbol{F}(t) - \boldsymbol{I}||_2}. \tag{2.2}$$

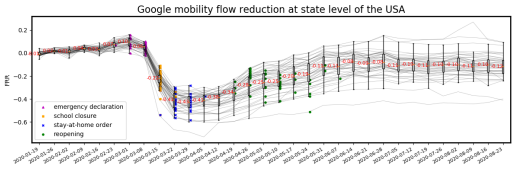
$\text{SDI}(t)$ value close to one indicates less mixing within a county while a value close to zeros indicates more mixing within a county.

Case count growth rate (CGR): Denoting the new confirmed case count at week t as n_t, the CGR of week $t+1$ is computed as $log(n_{t+1} + 1) - log(n_t + 1)$, where we add 1 to smooth zero counts.

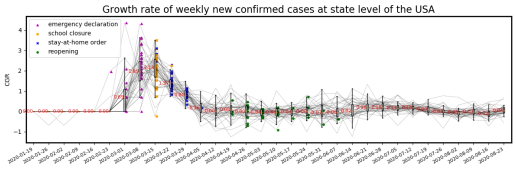
Spatiotemporal analysis of MF patterns and COVID-19 dynamics in the US. For MF analysis, we collected COVID-19 daily new confirmed case count data (see Section 2.3.1 for more details). The mobility data and confirmed data are processed to be weekly and end on Saturday. It starts from Week ending March 7, 2020 and ends at Week ending August 29, 2020 (27 weeks). The analysis is conducted in US 53 states and 3085 counties. In order to analyze the human mobility and COVID-19 dynamics during different phases of the pandemic, we use a 4-week window and moving one week ahead each time to compute Pearson correlation between new confirmed cases and MF within the window. Figure 2.2a shows the Pearson correlation along the weeks, Figure 2.2b shows FRR, and Figure 2.2c presents the timeline CGR, all together with social distancing orders. We observe that mobility flow and new confirmed cases show highly negative correlation (median -0.97) for almost all states during March and the correlation stays high until the mid-April. The high negative correlation, down trends of FRR (decreasing up to 41%) and up trends of CGR (increasing up to 2.11) during March indicates that as new confirmed cases increase the mobility flows decrease. Starting from the mid-April when the states started to reopen to some degree, there is a large variation in correlation values, where some are close to positive 1 while some are staying close to negative 1. However, we observe there has been a continuous rebound in the flows with flow reduction as the states reopening while the CGR of all the states remain in the range of [-0.04,0.22], which indicates that COVID-19 dynamics varies a lot due to that it is affected by multiple complicated



(a) Pearson correlation



(b) Weekly FRR



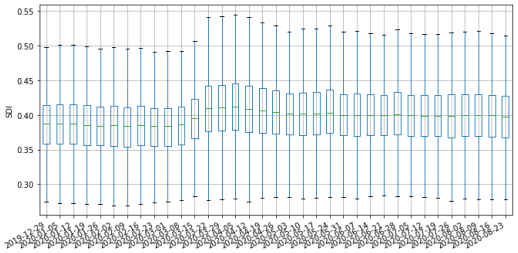
(c) Weekly CGR

Figure 2.2: Impact of state level social distancing policies on human mobility and COVID-19 dynamics. The time of state level social distancing mandates including emergency declaration (purple), school closure (orange), and stay-at-home (blue) are marked. The spaghetti plots showing time series of (a) Pearson correlation between MF and new confirmed cases, (b) the weekly FRR and (c) CGR in 53 states, respectively. A boxplot is used to display variation in samples of 53 states each week. The median value is shown along with the median line.

factors like local population size, individual behaviors (e.g., wearing a mask or not in public location), and government reopening guidelines.

Furthermore, Figure 2.2b shows a 41% ($Q1$: -50% and $Q3$: -30% in FRR) mobility reduction compared to baseline flows across the states during the week 2020/03/28 where most states had declared stay-at-home orders. There is a 27% reduction in flows during the week of the most school closure order, and 18% flow reduction in the week of the most stay-at-home orders. In the subsequent three weeks after the declaration of stay-at-home orders the flows remain nearly constant, but since then there has been a continuous rebound in the flows with flow reduction as the states reopening. The relative timing of the reduction in flows indicates that the population complied to the social distancing guidelines and reduced mobility.

We quantify mixing within counties using SDI. Figure 2.3 represents the variation in $SDI(t)$ across the various weeks of 2020. We observe $SDI(t)$ to be nearly constant until the implementation of national emergency and state-level orders after which we start to observe an increase in SDI (10%) indicating reduction in mixing within counties. As a general observation of variations in SDI across counties, we consider five states which have experienced the highest number of cases. We observe the overall shading moving towards yellow indicating reduction in mixing within counties. Since



(a) SDI at US counties

(b) SDI in the counties of New York, New Jersey, Massachusetts, Illinois, California pre-pandemic (2020/01/26).

(c) SDI in the counties of New York, New Jersey, Massachusetts, Illinois, California post-pandemic (2020/03/29)

(d) SDI in the counties of New York, New Jersey, Massachusetts, Illinois, California re-opening (2020/05/24)

Figure 2.3: Variation in SDI across counties of the US at different weeks of 2020. Increase in median SDI during March-April indicates overall reduction in mixing across counties. There is a slow but steady decrease in SDI since many states started reopening since early May 2020.

the last week of April, we observe a decrease in SDI indicating some degree of *social distancing fatigue* and possibly movement due to essential services. Importantly, the SDI has remained nearly constant over the month of May and median SDI nearly 5% higher than the median baseline values.

2.4 Graph Neural Network-based Epidemic Forecasting Framework Using Mobility Maps

2.4.1 Problem Formulation

We assume there are N regions and each region is associated with time series of multiple observed features, e.g., surveillance cases, in a time window T, where T is the observation duration. It could be of weekly or daily granularity depending on the data resolution. We define a dynamic graph of N regions as $G(\mathcal{V}, \mathcal{E}, \mathcal{T})$, where $\mathcal{V}$ is the set of N nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges, and $\mathcal{T}$ is the set of T time points. A node v_i at time t is attributed with $\boldsymbol{d}_{v_i,t} \in \mathbb{R}^{D_v}$ where D_v is the node feature numbers. An edge $e_{ij} \in \mathcal{E}$ at time t connecting nodes v_i and v_j is weighted by either adjacency matrix or mobility flow matrix, is attributed with $\boldsymbol{d}_{e_{ij},t} \in \mathbb{R}^{D_e}$ where D_e is

the edge feature numbers. Notations and their descriptions used in the work is shown in Table 2.1.

Table 2.1: Notations and their descriptions.

Notation	Description
$G(\mathcal{V}, \mathcal{E}, \mathcal{T})$	Dynamic graph with a set of nodes $\mathcal{V}$, a set of nodes $\mathcal{E}$ and a set of time steps $\mathcal{T}$
N	Number of nodes in $\mathcal{V}$
T	Number of time steps $\mathcal{T}$
v_i	A node i in a graph
e_{ij}	An edge from node v_i to node v_j
$\boldsymbol{d}_{v_i,t}$	Node feature vector at time t
D_v	Node feature number
$\boldsymbol{d}_{e_{ij},t}$	Edge feature vector at time t
D_e	Edge feature number
$f_{ij}(t)$	Mobility flow from node v_i to node v_j during time t
H	History window size
N_j^{active}	Cumulative cases exclude deaths in region v_j
N_j^{popu}	Population size in region v_j
H^f	Hidden dimension of node and edge feature encoding
p, f, g, o	Feature encoding, message passing, node update, and output functions
$\boldsymbol{h}_{i,t}$	Hidden state of node features $\boldsymbol{d}_{v_i,t}$
$\boldsymbol{a}_{ji,t}$	Hidden state of edge features $\boldsymbol{d}_{e_{ji},t}$
$\boldsymbol{m}_{i,t}$	Passing message to node v_i
η, σ	Nonlinear functions w.r.t attention coefficients
θ^v, θ^e	Trainable parameters in feature encoding modules
$\theta^f, \theta^g, \theta^\eta$	Trainable parameters in RMP modules
θ^o	Trainable parameters in the output module

2.4.2 Mobility Informed Graph Neural Networks

Constructing the graph. We construct a dynamic mobility graph $G(\mathcal{V}, \mathcal{E}, \mathcal{T})$, where each node feature $\boldsymbol{d}_v$ includes a sequence of dynamic observations regarding the region in a history window $H \leq T$. We include daily new case count, new death count and intra-region mobility flow ($f_{ii}(t)$ which represents the MF from v_i to v_i during time t) as the node features, i.e., $\boldsymbol{d}_{v_i,t} \in \mathbb{R}^{H\times 3}$. The graph edge features are derived from the inter-region mobility by aggregating Google mobility data to the state or county level. At a certain time point t, if there is any human movement from region j to region i in the past H days, we add a directed edge e_{ji} that connects region j and i, and associate it with the inter-region mobility flow $f_{ji}(t)$ and flow of active cases from source region (defined as $f_{ji}^{active}(t) = \frac{N_j^{active}(t)}{N_j^{popu}} * f_{ji}(t)$) as the edge feature i.e., $\boldsymbol{d}_{e_{ji},t} \in \mathbb{R}^{H\times 2}$ where $N_j^{active}(t)$ is the number of active cases (cumulative cases minus recovered cases and deaths) and N_j^{popu} is the population of region j.

Feature encoding. In the graph, node feature vectors and edge feature vectors include temporal information from the past. We encode the vectors using an LSTM module. At time step $t \in \mathcal{T}$, for each feature vector $\boldsymbol{d}_{v_i,t}$ or $\boldsymbol{d}_{e_{ij},t}$, the LSTM module encodes the vector into hidden representations as:

$$\begin{aligned} \boldsymbol{h}_{i,t} &= p(\boldsymbol{d}_{v_i,t}, \theta^v) \in \mathbb{R}^{H^n}, \\ \boldsymbol{a}_{ji,t} &= p(\boldsymbol{d}_{e_{ji},t}, \theta^e) \in \mathbb{R}^{H^e} \end{aligned} \tag{2.3}$$

where p denotes LSTM cell computation, H^n and H^e are hidden dimension of node feature and edge feature, θ^v and θ^e are parameters to be learned.

Spatiotemporal message passing. A region's COVID-19 dynamics can potentially be affected by regions where frequent travels occur between them. This resembles the core insight behind GNNs, i.e., the transformation of the input node's signal can be coupled with the propagation of information from a node's neighbors in order to better inform the future hidden state of the original input. This is most evident in the unified message passing framework proposed by Gilmer et al. (2017). In our model, we design a *Recurrent Message Passing* (RMP) module to recurrently pass the hidden representations from a node's neighbors to the current node. As shown in Figure 2.4, the RMP module has two phases: the message passing (MP) phase and the update phase (UP). It runs for L rounds, so any node in the graph is taking into account of neighbors that are L hops away. To be more specific, at time t, given a node v_i at a certain round $(l+1)$: In the MP phase, for each node pair (v_i, v_j) that $v_j \in \mathcal{N}(i)$ where $\mathcal{N}i$ denotes neighbors of v_i, we first combine the node hidden states $\boldsymbol{h}_{i,t}$, $\boldsymbol{h}_{j,t}$ and in-edge hidden representations $\boldsymbol{a}_{ji,t}$ from previous round l using a message passing function f to get a hidden state $\boldsymbol{a}_{ji,t}$ at current round $l+1$. It will later be aggregated (we use mean operation but can be sum, max, etc.) together over all pairs to obtain a message $\boldsymbol{m}_i$ for node v_i. In UP phase, we use a node upate function g to update the the node hidden states. The hidden states of the node v_i at the $(l+1)$th round $\boldsymbol{h}_{i,t}^{l+1}$ are updated in RMP module as :

$$\begin{aligned} \boldsymbol{a}_{ji,t}^{(l+1)} &= f\big(\boldsymbol{h}_{i,t}^{(l)}, \boldsymbol{h}_{j,t}^{(l)}, \boldsymbol{a}_{ji,t}^{(l)}, \theta^f\big) \in \mathbb{R}^{H^e}, \\ \boldsymbol{m}_{i,t}^{(l+1)} &= \sum_{j \in \mathcal{N}(i)} \boldsymbol{a}_{ji,t}^{(l+1)} \in \mathbb{R}^{H^m}, \\ \boldsymbol{h}_{i,t}^{(l+1)} &= g\big(\boldsymbol{h}_{i,t}^{(l)}, \boldsymbol{m}_{i,t}^{(l+1)}, \theta^g\big) \in \mathbb{R}^{H^n} \end{aligned} \tag{2.4}$$

where θ^f and θ^g are module parameters to be learned, $\mathcal{N}(i)$ denotes the neighbors of v_i where there exists e_{ji}, f is the message passing function that uses a multilayer perceptron (MLP) and g is the node update function that uses GRU, $\boldsymbol{m}^{(l+1)}$ is the messages passed between nodes.

Algorithm 1: MF informed GNN forward passing

Input: Time series of COVID-19 surveillance data for N regions, Google mobility flow data among N regions.

for *each time step* t **do**
  for *each region* i **do**
    $\boldsymbol{h}_{i,t} \leftarrow \text{FtrEncode}(\boldsymbol{d}_{v_i,t})$ `// Node features encoding`
  for *each region pair* (i, j) **do**
    $\boldsymbol{a}_{ji,t} \leftarrow \text{FtrEncode}(\boldsymbol{d}_{e_{ji},t})$ `// Edge features encoding`
    ▷ `Simultaneous calculations for all regions`
  for *each region* i **do**
    for l *in* $0, \ldots, L-1$ **do**
      $\boldsymbol{h}_{i,t}^{l+1} \leftarrow \text{RMP}(\boldsymbol{h}_{i,t}^{(l)}, \boldsymbol{h}_{j,t}^{(l)}, \boldsymbol{a}_{ji,t}^{(l)})$ `// Spatiotemporal message passing`
    $\hat{y}_{i,t} \leftarrow \text{Output}(\boldsymbol{h}_{i,t}^{(H)})$ `// Predicting`

Output layer. We feed the hidden representations to the output layer for the final forecasts:

$$\hat{y}_{i,t} = o\big(\boldsymbol{h}_{i,t}^{(L)}, \theta^o\big) \in \mathbb{R} \tag{2.5}$$

where θ^o is the parameters to be learned, o is the output function which is a MLP in our model.

Forward passing process. As shown in Figure 2.4, we first feed the sequences of temporal node and edge features through the feature encoding module to obtain node and edge embedding, which are utilized as the initial node and edge hidden representations for the RMP module. Then we perform MP and UP computations for L rounds. This is the core step to allow one region to leverage information from its neighbors and their connectivity in between. The output module will output the final forecasts.

Proposed models. The proposed model aims to examine the effect of dynamic mobility on understanding and forecasting COVID-19 dynamics. Thus we design several variants of the proposed model using dynamic mobility graph denoted as GNN-dmob , using a static mobility graph denoted as GNN-smob , using a geographical adjacency graph denoted as GNN-adj , and using an attention-based matrix denoted as GNN-att . The details are listed below.

- **GNN-dmob** the proposed model with dynamic mobility graph.
- **GNN-adj** uses the same graph structure with GNN-dmob but remove intra-region mobility flow from node features i.e., $\boldsymbol{d}_{v_i,t} \in \mathbb{R}^{H\times 2}$, and construct an

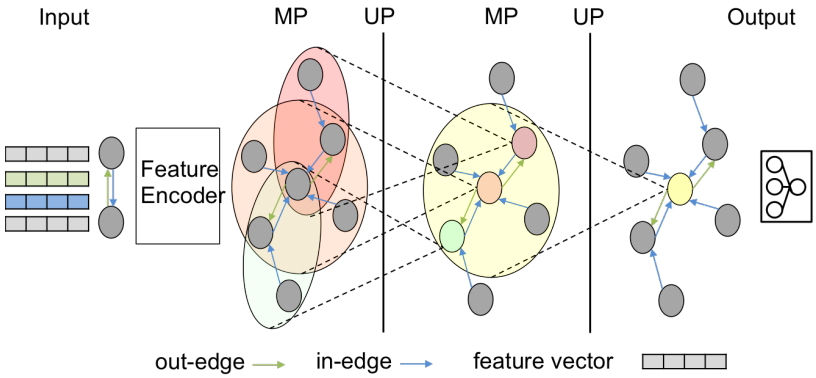


Figure 2.4: An example of two-hop RMP architecture. Temporal node feature and edge feature vectors are encoded using the feature encoder module. A two-hop RMP module is used to further embed spatiotemporal information to hidden representations. The output module makes the final forecasts.

adjacency matrix by adding an edge e_{ij} if region j is a neighbor of region i with edge weight 1. The adjacency matrix is normalized by row summation. It is static across time steps, thus the edge feature is $\boldsymbol{d}_{e_{ij},t} \in \mathbb{R}$.

- **GNN-smob** is similar to the GNN-adj but obtained by replacing the adjacency matrix with a static mobility graph which is an average of mobility graphs from March 1, 2020, to August 2, 2020.

- **GNN-att** is inspired by cola-GNN proposed in (Deng et al., 2019b). Instead of using a physical matrix in our model, we implement an attention-based model that allows the model to learn an attention matrix of all the regions. In MP phase, we update the message between nodes as:

$$\boldsymbol{m}_{i,t}^{(l+1)} = \eta\Big(\sum_{j\in\mathcal{N}} a_{ji,t}^{(l)} \boldsymbol{h}_{j,t}^{(l)}, \theta^{\eta}\Big) \tag{2.6}$$

where θ^{η} is the trainable parameters, a_{ji} is the attention coefficient defined as: $a_{ji,t}^{(l)} = \boldsymbol{v}^T\sigma(\boldsymbol{W}^i\boldsymbol{h}_{i,t}^{(l)} + \boldsymbol{W}^j\boldsymbol{h}_{j,t}^{(l)} + \boldsymbol{b}) \in \mathbb{R}$ where $\boldsymbol{v} \in \mathbb{R}^{H^a}$, $\boldsymbol{W} \in \mathbb{R}^{H^a \times H^f}$, $\boldsymbol{b} \in \mathbb{R}^{H^a}$ are trainable parameters, σ is Rectified Linear Units (ReLU) applied at element-wise.

2.5 EXPERIMENTS

In this section, we will introduce datasets, evaluation metrics, and baselines that are used for demonstration. Then the epidemic forecasting performance is presented and discussed. Sensitivity analysis on hyperparameters is shown at last.

2.5.1 Settings

Datasets. Google COVID-19 Aggregated Mobility Research Dataset (Kraemer et al., 2020a) (details are presented in Section 2.3). **COVID-19 surveillance data (CSD)** via the UVA COVID-19 surveillance dashboard (UVA, 2020). It contains daily confirmed cases and death count worldwide. The data is available at the level of a county in the US. Daily case counts and death counts are further aggregated to weekly counts.

Metrics. The metrics used to evaluate the forecasting performance are: *root mean squared error (RMSE)* and *Pearson correlation (PCORR)*. Assuming we have n testing data points and $n = N \times m$ means N locations by m weeks. We denote the true value and forecast for the ith testing data point to be z_i and $\hat{z}_i$. We do not distinguish locations in calculating RMSE. PCORR is calculated by locations and the final value is the average of all locations. RMSE evaluate forecasting accuracy, PCORR evaluates linear correlation between the true curve and the predicted curve.

- **Root mean squared error (RMSE)**:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(z_i - \hat{z}_i)^2} \tag{2.7}$$

  RMSE ranges in $[0, +\infty]$ and smaller values are better.

- **Pearson correlation (PCORR)** is calculated per location:

$$\text{PCORR} = \frac{\sum_{i=1}^{m}(\hat{z}_i - \bar{\hat{z}})(z_i - \bar{z})}{\sqrt{\sum_{i=1}^{m}(\hat{z}_i - \bar{\hat{z}})^2}\sqrt{\sum_{i=1}^{m}(z_i - \bar{z})^2}} \tag{2.8}$$

  PCORR ranges in [-1, +1] and larger values are better. The final PCORR values shown in our results are the average PCORR of N locations.

Baselines. We implemented several classic and state-of-the-art methods as the comparison methods.

- **Naive** uses the observed value of the most recent time point as the future forecast.

- **Autoregressive (AR)** uses observations from previous time steps as input to a regression equation to predict the value at the next time step. We train one model per region using AR order 28.

- **Autoregressive Moving Average (ARMA)** (Contreras et al., 2003) is used to describe weakly stationary stochastic time series in terms of two polynomials

for the autoregression (AR) and the moving average (MA). We set AR order to 28 and MA order to 2.

- **CNNRNN** (Wu et al., 2018) uses RNNs to capture the long-term correlation in the data and uses CNNs to fuse time series of other regions. A residual structure is also applied in the training process. We train one model per region. We set the residual window size as 28 and all the other parameters are set as the same as the original paper.

- **cola-GNN** (Deng et al., 2020) uses a graph message passing framework to combine graph structures and time series features in a dynamic propagation process. We set the RNN window size as 28 and all the other parameters are set as the same as the original paper.

Settings and implementation details. For forecasting tasks, we conduct model training and forecasting using daily (rather than weekly) new confirmed cases at US state level regarding the limited number of observations. The weekly mobility graph is expanded to daily by dividing the weekly values by 7. The training data set is from March 1 to August 1 (125 days), the testing set is from August 2 to August 29 (28 days). We make 2, 7, 14, 21, and 28 days ahead forecasting for each data point in the testing set. For all models, the historical window $H = 28$. For GNN-dmob , we use a single layer LSTM for feature encoding with 16 units, a two-layer MLP in MP phase with 32 and 16 units, and a single layer GRU in UP with 16 units. The same settings are used for GNN-smob , GNN-att , and GNN-adj . AR and ARMA use AR order 28 and ARMA uses MA order 2. CNNRNN and cola-GNN set with their best parameter settings in the original paper. We set batch size as 32, epoch number as 1000. The mean squared error (MSE) loss function and Adam Optimizer with default settings and early stopping with patience of 100 epochs are used for all model training. All results are average of 5 random runs. We did not demonstrate our model at the US county level because daily new cases at county level is much noisier and most of them are around zeros during March and April. Through our preliminary experiments, we see that GNN-based models (including GNN-based baselines) trained on such datasets make arbitrary forecasts for some of the counties. Note that we are not trying to convince others that the proposed model is the best among all existing models, but to provide a possible good way to utilize human mobility information for COVID-19 forecasting using GNNs.

2.5.2 Results and Analysis

Forecasting performance. The proposed GNN-dmob model is evaluated w.r.t. forecasting daily new confirmed cases at US state level for 2, 7, 14, 21 and 28 days ahead. Table 2.2 presents the RMSE, MAE, and PCORR performance averaging

Table 2.2: RMSE and PCORR performance of different methods on the US state dataset with horizon = 2, 7, 14, 21, and 28. The values are average of 5 runs. Bold face indicates the best results of each column.

	US state				
RMSE(↓)	2	7	14	21	28
Naive	411	389	445	496	525
AR	376	634	866	975	978
ARMA	400	637	944	1107	1186
LSTM	368	504	517	567	605
CNNRNN	416	432	512	606	659
cola-GNN	320	502	451	530	714
GNN-adj	**310**	411	407	412	513
GNN-att	319	479	985	457	745
GNN-smob	313	405	410	**406**	510
GNN-dmob	313	**330**	**350**	445	**465**
PCORR(↑)	2	7	14	21	28
Naive	0.106	0.361	0.310	0.261	0.157
AR	0.283	0.282	0.044	-0.017	-0.097
ARMA	0.281	0.260	0.071	-0.078	-0.133
LSTM	0.227	0.305	0.287	0.226	0.204
CNNRNN	0.211	0.250	0.265	0.248	0.029
cola-GNN	0.366	0.341	0.247	0.175	0.176
GNN-adj	0.293	**0.395**	0.319	0.315	**0.280**
GNN-att	**0.376**	0.256	0.087	0.136	0.238
GNN-smob	0.321	0.382	0.297	**0.294**	0.226
GNN-dmob	0.298	0.344	**0.320**	0.287	0.216

across 53 states and 28 days. In general, we observe that GNN-dmob has better RMSE and MAE performance than the comparisons for long-term forecasting. GNN-adj performs the best for 2 days ahead forecasting. The best performances on PCORR are evenly distributed among the proposed models. The results indicate that our proposed methods can capture the disease dynamic in both short-term and long-term. Naive baseline outperforms the other baselines for 7, 14, 21, 28 days ahead forecasting. This is not surprising because the testing data is from August during when the time series of all states are showing downward trends with small changing rates. The Naive assumes certain level of regularity in the time series leading to good forecasting performance on the testing data. DNN-based models perform better than AR-based models especially on long-term forecasting which indicates that DNN-based models have better generalization capability for forecasting unseen data. In our experiments, attention-based models cola-GNN and GNN-att are not outstanding for both short and long-term forecasting. A possible reason is that the learned attention coefficients are

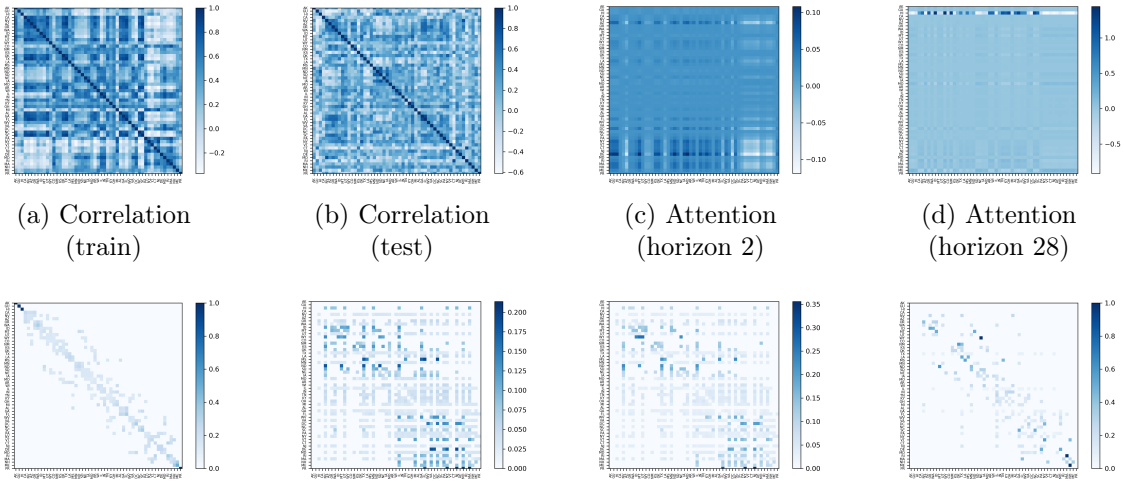
outdated due to the fast evolution in COVID-19 dynamics between training time period and testing time period, which leads to false attention by regions while predicting (refer to MF analysis w.r.t Figure 2.2).

Major observations and discussion: In Figure 2.5 we show heatmap of correlation matrix of training data and testing data, learned attention matrix of GNN-att at 2020/07/15 for horizon 2 and 28, geographical adjacency matrix, static MF matrix, and dynamic MF matrix at two different dates i.e., pre-MF (2020/03/02) and post-MF (2020/03/28) stay-at-home orders. The correlation matrix is the Pearson Correlation between two time series of daily new confirmed cases between two regions. Self-loops are suppressed from MF matrices. Except the learned attention matrix, all the other matrices are normalized by row summation values which are consistent with values used in the model training. We can observe that attention matrix of horizon 2 captures a similar pattern with correlation matrix of training data while it does not capture correlation patterns in testing data. The attention matrix of horizon 28 shows a different pattern with correlations. This indicates that the learned attention matrix may not capture disease dynamics of testing dataset for long-term forecasting. It explains why we observe that cola-GNN and GNN-att achieve comparable performance for short-term forecasting but worse performance for long-term forecasting compared with other proposed models. Furthermore, it involves more model parameters thus may mitigate its power when there is no sufficiently good quality training data. Static MF matrix show normal mobility flows between regions regardless of social distancing orders, while dynamic MF capture mobility change before and after government interventions. The pre-MF is close to static MF matrix while post-MF is close to adjacency matrix. Thus, in Table 2.2 we observe that GNN-adj performs similarly with GNN-smob , while GNN-dmob explicitly projects recent mobility patterns to the future leading to better performance.

Sensitivity analysis. We show sensitivity analysis on some of the hyperparameters: number of hops L (Figure 2.6) and historical window size K (Figure 2.7). Except the varying hyperparameter, all the other settings are the same with parameter setting in Section 2.5.1. We report RMSE and PCORR performance on the US-State dataset with horizon=28. The other results show similar observations thus are omitted for the sake of brevity.

Number of hops L. The L values are 2, 3, 4, and 6. Results are shown in Figure 2.6. In general, we observed that models trained with $L = 4$ achieves relatively better RMSE and PCORR performance than models with other settings. However, models with $L = 3$ outperform others in 7, 21, and 28 days ahead forecasting on PCORR while performing the worst in 14, 21, and 28 days ahead forecasting on RMSE. In the main experiment, we use $L = 4$ for the proposed models.

History window H. The H varies among 7, 14, 21, and 28. Results are shown in Figure 2.7. We can observe that as the window size increases, the model performance increases. However, there is an exception for the model with $H = 21$ which has larger



(a) Correlation (train) (b) Correlation (test) (c) Attention (horizon 2) (d) Attention (horizon 28)

(e) Geographical adjacency (f) Static MF (Average) (g) Dynamic MF 2020-03-02 (h) Dynamic MF 2020-03-28

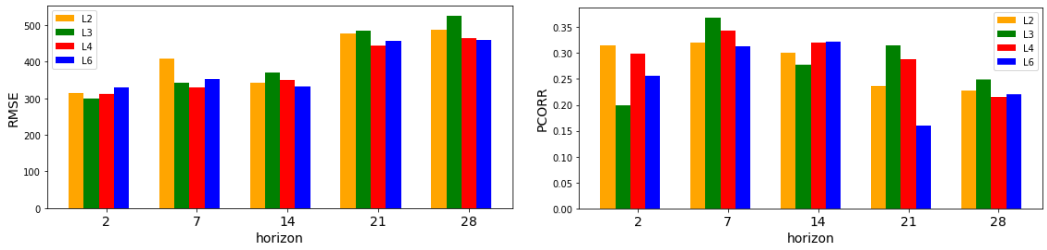
Figure 2.5: Heatmap of new cases correlation matrix (on training and testing datasets) (2.5a,2.5b), learned attention matrix at July 15, 2020 (horizon 2 and 28) (2.5c,2.5d), geographical adjacency matrix (2.5e), static MF matrix (2.5f), dynamic MF matrix (2.5g) and (2.5h) between states. States are ordered as per the health and human services grouping to obtain a sense of adjacency, self-loops are suppressed from MF matrices.

RMSE but smaller PCORR.

Major observations and discussion: we observed that some models like the one with $L = 3$ in Figure 2.6 and the one with $H = 21$ in Figure 2.7 perform inconsistently on different metrics. The likely reason is that all models may be overfitting to training dataset in some degree due to the small size of training dataset. In addition, the ground truth curves have large variation due to the noise in the surveillance data. Thus, the correlation (i.e., measuring the trend similarity) between the ground truth curves and the predicted curves may vary more across timeline and regions compared with average RMSE errors (i.e., measuring the forecast error).

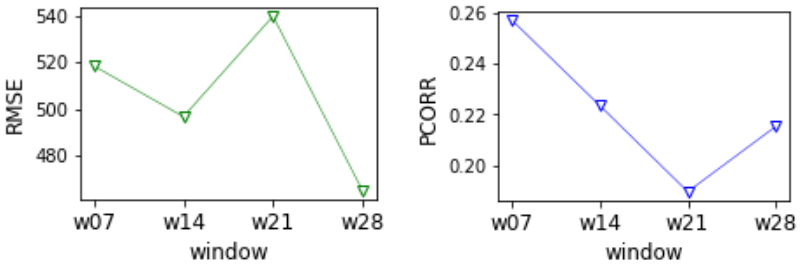
2.6 Conclusions and Open Questions

In this chapter, we introduce a novel GNN-based framework to incorporate aggregated mobility flows for better understanding the impact of human mobility on COVID-19 dynamics as well as better forecasting of disease dynamics. We propose an RMP module to embed spatiotemporal disease dynamics (COVID-19 surveillance data) and human mobility dynamics (MF data) while making forecasting. The experimental results of forecasting daily COVID-19 new cases for each state in the US demonstrate



(a) RMSE (b) PCORR

Figure 2.6: Sensitivity analysis on the number of hops L.



(a) RMSE (b) PCORR

Figure 2.7: Sensitivity analysis on history window H.

the additional improvements obtained by using the mobility data. The use of GNNs for COVID-19 is just beginning and our results are some of the first results in this area to yield good performance. The proposed model is flexible to account for a variety of static and dynamic spatiotemporal signals and can be extended to forecast other diseases dynamics. However, the proposed model forecasts at daily new confirmed cases but using weekly mobility graphs, which may degrade model performance. Furthermore, as we have discussed in Section 2.3.2 that COVID-19 dynamics varies a lot due to multiple factors like local population size, individual behaviors (e.g., wearing a mask or not in public location), and government reopening guidelines. Including these factors in the model can further improve the forecasting accuracy.

The use of MF in this work should be interpreted in consideration of several important limitations. First, there are limitations due to lack and non-uniformity of testing data. Second, the Google mobility data is limited to smartphone users who have opted into Google's Location History feature, which is off by default. These data may not be representative of the population as whole, and furthermore their representativeness may vary by location. Importantly, these limited data are only viewed through the lens of differential privacy algorithms, specifically designed to protect user anonymity and obscure fine detail. Moreover, comparisons across locations are only descriptive since these regions can differ in substantial ways.

Chapter 3

Combining Theory and Deep Learning Models for Reliable Epidemic Forecasting

In this chapter, which is based on Wang et al. (2019b, 2020b) and a work[1] that is under submission. We will present two works that aim to enhance deep learning models with theory-based mechanistic models for improving the forecasting accuracy and the interpretability of a learned model. Section 3.1 introduces the general background of combining theory and deep learning models for epidemic forecasting. The two frameworks TDEFSI and CausalGNN are presented in Section 3.2 and 3.3 respectively. Finally, the conclusions and open questions are discussed in Section 3.4.

3.1 Background

Given the challenges discussed in Section 1.1.2 and 1.1.3, existing deep learning-based epidemic forecasting models (Wu et al., 2018; Deng et al., 2020) barely considered epidemiological context during their learning process. Such models are prone to be overfitting leading to failures in epidemic forecasting, especially when the data is noisy and sparse. Furthermore, they lack explainability of the underlying causal mechanism. On the other hand, theory-based mechanistic models can capture the diffusion patterns of disease spread through detailed simulation use a realistic representation of the underlying social contact network formed by human mobility behavior and population demographics. Thus, a framework that combines deep learning with theory-based mechanistic models is promising. Current efforts on combining theory-based mechanistic models with data-driven methods are along two lines. One direction is using machine learning techniques to enhance theory-based mechanistic models. The efforts in this research direction include but not exhausted to (Zhao et al., 2015a;

[1] CausalGNN: Causal-based Graph Neural Networks for Spatio-Temporal Epidemic Forecasting

Hua et al., 2018) which used social media mining techniques to enhance theory-based mechanistic models for influenza forecasting and (Dandekar et al., 2020) which mixed first-principles epidemiological equations and a data-driven neural network when calibrating mechanistic model parameters. The other direction is using mechanistic causal theory to enhance data-driven models. A limited number of works have been made towards this direction. Wang et al. (2019b, 2020b) trained an LSTM-based model with theory-generated training data for forecasting influenza at high geographical resolution. Recently for COVID-19 forecasting, Gao et al. (2021) used a transmission dynamics loss term to regularize model forecasts in a GNN training. My works introduced in this chapter are among the first efforts in this direction. Specifically, my works focus on enhancing deep learning models with theory-based mechanistic models with the aim of providing accurate forecasts as well as gaining a mechanistic understanding from a learned model.

In the rest of this chapter, we will present two frameworks TDEFSI in Section 3.2 and CausalGNN in Section 3.3. TDEFSI shows a sequential learning process where where mechanistic models are used to generate high-resolution synthetic data then RNN-based models are trained with synthetic training data, while CausalGNN adopts a jointly learning process that learns a latent space to combine the spatiotemporal and causal embeddings using graph-based non-linear transformations.

3.2 TDEFSI

3.2.1 Motivation

Influenza-like illness (ILI) poses a serious threat to global public health. Worldwide, annually, seasonal influenza causes three to five million cases of severe illness and 290,000 to 650,000 deaths (WHO, 2019). Since 2010 in the USA, seasonal influenza has resulted in 10-50 million cases annually, 140,000 to 960,000 hospitalizations, between 12,000 and 79,000 deaths, and is responsible for approximately $87.1 billion in economic losses (CDC, 2019a; Molinari et al., 2007). Producing timely, well-informed, and reliable forecasts for ILI of an ongoing flu epidemic is crucial for preparedness and optimal intervention (Doms et al., 2018). Traditionally, ILI surveillance data from the Centers for Disease Control and Prevention (CDC) has been used as reference data to predict future ILI incidence. The surveillance data is updated weekly but often delayed by one to four weeks and is provided at an HHS region (i.e., the ten regions defined by the United States Department of Health & Human Services) level and recently at the state level. Considering the heterogeneity between different subregions, accurate forecasts with a finer resolution, e.g., at county or city level in the USA, are crucial for local public health decision making, optimal mitigation resource allocation among subregions, and household or individual level preventive actions informed by neighboring prevalence (Yang et al., 2016). Given spatially coarse-grained surveillance

data, it is challenging to forecast at a finer spatial level. In this work, we formally define *flat-resolution* forecasting and *high-resolution* forecasting. The definitions are shown in Section 1.1.2 "Spatial forecasting" paragraph. We do not repeat them for the sake of brevity.

To address this problem, we proposed a novel epidemic forecasting framework, called **T**heory Guided **D**eep Learning Based **E**pidemic **F**orecasting with **S**ynthetic **I**nformation (**TDEFSI**). TDEFSI uses theory generated synthetic data to train a neural network. This is necessitated by the fact that disease surveillance data is sparse. Furthermore, the data is noisy and incomplete. We overcome the limitations by training TDEFSI using data generated by high-performance-computing-based simulations of well accepted causal processes that capture epidemic dynamics. These simulations are based on decades of work and have been extensively validated. The simulations allow us to: (*i*) use a realistic representation of the underlying social contact network that captures the multi-scale spatial, temporal and social interactions, as well as the inherent heterogeneity of social networks (individual demographic attributes, heavy tailed nature of social contacts, etc.), leading to *forecasts that are context specific and capture the unique properties of a given urban region*; (*ii*) produce multi-resolution forecasts even though observational data might only be available at an aggregate level, leading to *an ability to forecast disease incidence at a county or a city level as well as forecasts for desired demographic groups*; and (*iii*) capture the underlying causal processes and mathematical theories leading to *explainable and generalizable AI* – the combination of theory and data driven machine learning is an important and emerging approach to scientific problems that are data sparse.

TDEFSI produces accurate weekly high-resolution ILI forecasts from flat-resolution observations. This is achieved by using a two-branch neural network for ILI forecasting. It combines within-season observations (observed data points of the previous weeks that characterize the ongoing epidemic) and between-season historical observations (observed data points from similar weeks of the past seasons that characterize general trends around the current week). It can generate probabilisitic forecasts by using Monte Carlo Dropout (MCDropout) technique (Gal and Ghahramani, 2016).

To the best of our knowledge, TDEFSI is the first to use a realistic causal high-resolution model to train a DNN for epidemic forecasting. The basic approach is general and points to the potential utility of the approach to study other problems in social and ecological sciences. Unlike physical systems, encoding system level constraints is often possible only via simulations; the theories are largely local rules of interactions. In this sense, training a DNN using simulations provides a natural way to place constraints on the concept class that the DNN effectively learns.

A natural question that arises is: *why does one need to use a DNN when simulations are available?* There are multiple reasons to do this: (*i*) computational efficiency (ability to rapidly produce forecasts, (*ii*) generalizability (often simulation parameters might end up overfitting to the data), and (*iii*) ability to incorporate additional data sources. In this sense, *DNN+simulations* appears to be a promising approach for

forecasting rather than using either of them individually.

3.2.2 Related Work

Influenza forecasting. In Section 1.1.3, we have introduced methodologies for epidemic forecasting in three categories: theory-based mechanistic methods, statistical time series methods, and deep learning methods. We also discussed the limitation of each type of methods. In this section, we will discuss forecasting methods specific for high-resolution ILI. For the sake of brevity, we omitted the related works which have been introduced in Section 1.1.3.

Theory-based mechanistic methods for ILI forecasting employing within-host progression models (e.g., SIR or SEIR) can determine the casual mechanisms of influenza. The underlying epidemic model can be either a CM or an ABM (more details in Section 1.1.3). To get county level epidemics in a CM, one needs to create compartments in each county, where county population sizes and between county travel data become crucial. On the other hand, the individual level details in an ABM can be easily aggregated to obtain epidemic data of any resolution, e.g., number of newly infected people in a county in a specific week. Causal methods are generally computationally expensive as they require the parameter estimation over a high dimensional space. As a result, the use of such methods for real-time forecasting is challenging.

Popular statistical time series methods for ILI forecasting include e.g., generalized linear models (GLM), autoregressive (AR) models, and autoregressive integrated moving average (ARIMA) (Bardak and Tan, 2015; Benjamin et al., 2003; Dugas et al., 2013). Wang et al. (2015) proposed a dynamic Poisson autoregressive model with exogenous input variables (DPARX) for flu forecasting. Yang et al. (2015b) proposed ARGO, an autoregressive-based influenza tracking model for nowcasting incorporating CDC ILI data and Google search data. The extensive work based on ARGO was discussed by Yang et al. (2017). Statistical methods are fast but they crucially depend on the availability of training data and as such can only produce flat-resolution forecasts. High-resolution forecasts must be calculated by multiplying the flat-resolution forecasts with high-resolution population proportions. The trained models could not capture the heterogeneous dynamics between high-resolution regions.

DNNs have gained increased prominence in ILI forecasting. However, just like statistical time series methods, DNN-based forecasting methods are data driven and have similar limitations. In addition, the model performance usually depends on the availability of a very large training dataset. Another well known limitation of DNNs is their ability to explain the resulting forecasts.

Hybrid methods combining data-driven and mechanistic methods are attractive as they can borrow the best from both worlds (Kandula et al., 2018). The authors in (Osthus et al., 2019) proposed a dynamic Bayesian model for influenza forecasting

which combined the machine learning approach and a compartmental model to explicitly account for systematic deviations between mechanistic models and the observed data. Such methods have shown promise as evidenced in recent papers on the study of physical and biological systems (Faghmous et al., 2014; Fischer et al., 2006; Hautier et al., 2010; Kawale et al., 2013; Khandelwal et al., 2015, 2017; Karpatne et al., 2017; Wong et al., 2009; Xu et al., 2015) – see (Karpatne et al., 2017) for a discussion on this subject. However, none of these works investigate high-resolution epidemic forecasting.

TDEFSI method: Our method combines DNNs and high-resolution epidemic simulations to enable accurate weekly high-resolution ILI forecasts from flat-resolution observations. Compared with mechanistic methods, TDEFSI avoids searching optimal disease model parameters over a high dimensional space because it does not need to identify any specific mechanistic models for the forecasting. Compared with data-driven methods (statistical and deep learning methods), TDEFSI explicitly models spatial and social heterogeneity in a region from the training data. It can capture the heterogeneous dynamics between high-resolution regions, as well as underlying causal processes and mathematical theories. In addition, the large volume of synthetic training data helps TDEFSI to overcome the risk of overfitting due to sparse observation data.

Data augmentation for time series. Data augmentation in DNNs is the process of generating artificial data in order to reduce overfitting. It has been shown to improve the DNNs' generalization capabilities in many tasks especially in computer vision tasks such as image or video recognition (Schlüter and Grill, 2015). Various augmentation techniques have been applied to specific problems, including affine transformation of the original images (Vasconcelos and Vasconcelos, 2017; Rizk et al., 2019; Wong et al., 2016) and unsupervised generation of new data using Generative Adversarial Nets (GANs) (Perez and Wang, 2017; Gurumurthy et al., 2017; Marchesi, 2017; Zhu et al., 2017) or variational autoencoder (VAE) models (Rizk et al., 2019), etc. However, the techniques for image augmentation do not generalize well to time series. The main reason is that image augmentation is not expected to change the class of an image, while for time series data, one cannot confirm the effect of such transformations on the nature of a time series. In what follows we introduce related work on time series data augmentation.

Data augmentation for time series classification: For time series classification (TSC) problems, one of the most popular methods is the slicing window technique, originally introduced for CNNs in (Cui et al., 2016). The method was inspired by the image cropping technique for computer vision tasks (Zhang et al., 2016). In (Kvamme et al., 2018), it was adopted to improve the CNNs' mortgage delinquency prediction using customer's historical transactional data. The authors in (Krell et al., 2018) used it to improve the Support Vector Machines accuracy for classifying electroencephalographic time series. The authors in (Um et al., 2017) proposed a novel data augmentation method (including window slicing, permutating, rotating,

time-warping, scaling, magnitude-wrapping, jiterring, cropping) specific to wearable sensor collected time series data. Le Guennec et al. (2016) extended the slicing window technique with a warping window that generated synthetic time series by warping the data through time. It extracted multiple small-size windows from a single window and lengthens/shortens a part of the window data, respectively. The methods were reported to reduce classification error on several types of time series data. Forestier et al. (2017) proposed to average a set of time series as a new synthetic series. It relied on an extension of Dynamic Time Warping (DTW) Barycentric Averaging (DBA).

Data augmentation for time series regression: Unlike data augmentations for TSC, data augmentation for time series regression (TSR) has not been well investigated yet to the best of our knowledge. Bergmeir et al. (2016) presented a method using Box-Coxfor transformation followed by a Seasonal and Trend decomposition using Loess (STL) decomposition to separate the time series into trend, seasonal part, and remainder. The remainder was then bootstrapped using a moving block bootstrap, and a new series was assembled using this bootstrapped remainder.

All above methods for TSC or TSR apply techniques directly on observed time sequences, which generate synthetic data at the same resolution as the original data. In our problem, we try to forecast at a higher resolution when there is no or very sparse high-resolution observations.

TDEFSI method: We generate synthetic high-resolution data using high performance computing based simulations of well accepted causal processes that capture epdemic dynamics. Different from data augmentation techniques introduced above, we synthesize high-resolution data which is not available or quite sparse in the real world.

3.2.3 Problem Formulation

Given an observed time series of weekly ILI incidence for a specific region, we focus on predicting ILI incidence for both the region and its subregions in short-term. Without loss of generality, in this work we consider making forecasts for a state of the USA and all counties in the state, using observations only from CDC state level ILI incidence data (CDC, 2019b). In this setting, state level forecasting is flat-resolution, while county level forecasting is high-resolution. The proposed framework is not limited to this setting and can be generalized for subregion forecasting in any region, e.g., state level forecasting in a country where only national level surveillance data is available. Our proposed method is different from traditional ILI incidence forecasting methods in that the model is trained on synthetic ILI incidence data but forecasts by taking ILI surveillance data as inputs.

Let $\mathbf{y} = \langle y_1, y_2, \cdots, y_T, \cdots \rangle$ denote the sequence of weekly state level ILI incidence, where $y_i \in \mathbb{R}$. Let $\mathbf{y}^C = \langle y_1^C, y_2^C, \cdots, y_T^C, \cdots \rangle$ denote the sequence of weekly ILI incidence for a particular county C within the state. Assume that there are K counties

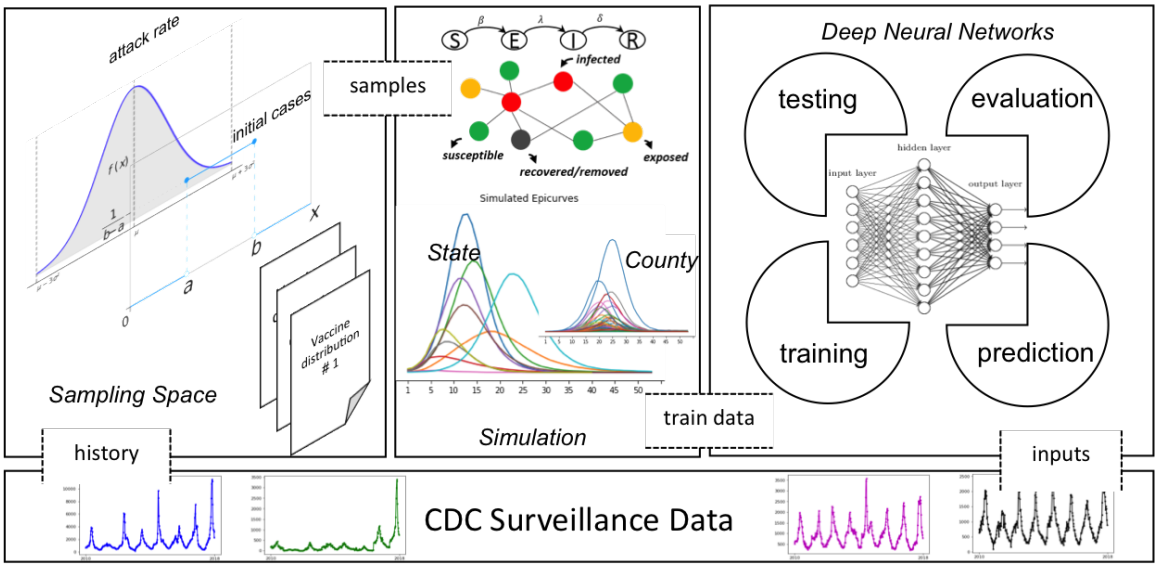


Figure 3.1: TDEFSI framework. In this framework, a region-specific disease parameter space for a disease model is constructed based on historical surveillance data. Synthetic training data consisting of both state level and county level weekly ILI incidence curves is generated by simulations parameterized by samples from the parameter space. An LSTM-based model is trained on the synthetic data. The trained model produces forecasts by taking surveillance data as the input.

$\mathcal{D} = \{C_1, C_2, \cdots, C_K\}$ in the state. Let $\mathbf{y}_t^{\mathcal{D}} = \{y_t^C | C \in \mathcal{D}\}$ denote ILI incidence of all counties in the state at week t. Suppose we are given only state level ILI incidence up to week T. The problem is defined as predicting both state level and county level incidence at week t, where $t = T + 1$, denoted as $\mathbf{z}_t = (y_t, \mathbf{y}_t^{\mathcal{D}}), \mathbf{z}_t \in \mathbb{R}^{K+1}$, given $\langle y_1, y_2, \cdots, y_T \rangle$.

In our problem, when training a DNN, we consider three types of physical consistency requirements based on epidemiologic domain knowledge. They are **temporal consistency**, **spatial consistency**, and **non-negative consistency**. (i) Temporal consistency: the ILI diseases transmit via person to person contacts. The number of infected cases at the current time point depends on the number of infected cases at the previous time points. In addition, infected persons' incubation periods and infectious periods vary due to the heterogeneity among individuals. In our work, we use an LSTM (Hochreiter and Schmidhuber, 1997) to capture the temporal dependencies among variables. (ii) Spatial consistency: the high-resolution ILI incidence should be consistent with the flat-resolution ILI incidence. In our problem, this consistency is represented as $y_t = \sum_{C \in \mathcal{D}} y_t^C$, i.e., the state incidence equals the sum of ILI incidence at the county level. (iii) Non-negative consistency: the number of infected cases at time t is either zero or a positive value, denoted as $y_t, y_t^C \geq 0$.

3.2.4 Framework

The TDEFSI framework consists of three major components (shown in Figure 3.1): (*i*) *Disease model parameter space construction*: given a state and an existing disease model, we estimate a marginal distribution for each model parameter based on the surveillance data of the state and its neighbors; (*ii*) *Synthetic training data generation*: we generate a synthetic training dataset at both flat-resolution and high-resolution scales for that state by running simulations parameterized from the parameter space; (*iii*) *DNN training and forecasting*: we design a two-branch DNN model trained on the synthetic training dataset and use surveillance data as its inputs for forecasting. It works as follows: given a theory-based mechanistic model, a disease model parameter space is identified and estimated using surveillance data. Then a synthetic training dataset is generated using the mechanistic model and the learned parameter space. Next, a two-branch DNN is built and trained with the generated synthetic dataset. Finally, the learned model is used to make forecasts. We will elaborate on the details in the following subsections.

SEIR-based epidemic simulation. We simulate the spread of the disease in a synthetic population via its social contact network. In this work we use the synthetic social contact network of each state in the USA (a brief description of the methodology used for constructing the synthetic population and the social network can be found in (Wang et al., 2020b)). The SEIR disease model is widely used for ILI diseases (Kuznetsov and Piccardi, 1994). Each person is in one of the following four health states at any time: susceptible (S), exposed (E), infectious (I), recovered or removed (R). A person v is in the susceptible state until he becomes exposed. If v becomes exposed, he remains so for $p_E(v)$ days, called the incubation period, during which he is not infectious. Then he becomes infectious and remains so for $p_I(v)$ days, called the infectious period. Both $p_E(v)$ and $p_I(v)$ are sampled from corresponding distributions, as shown in Algorithm 2, e.g., $p_E(v) \sim \{1 : 0.3, 2 : 0.5, 3 : 0.2\}$ means that an exposed person will remain so for 1 day with probability 0.3, 2 days with probability 0.5, and 3 days with probability 0.2, similar to $P_I(v)$. Finally, he becomes removed (or recovered) and remains so permanently. While the SEIR model characterizes within-host disease progression, between-host disease propagation is modeled by transmissions from person to person with a probability parameter τ, through either complete mixing or heterogeneous connections between people. With our contact network model, the disease spreads in a population in the following way. It can only be transmitted from an infectious node to a susceptible node. On any day, if node u is infectious and v is susceptible, disease transmission from u to v occurs with probability $p(\tau, w(u, v))$, where $w(u, v)$ represents the contact duration between node u and node v. The disease propagates probabilistically along the edges of the contact network.

Various simulators are developed to model human mobility, disease spread, and public health intervention. They include compartment-based patch models (Flahault

et al., 2006; Lee et al., 2012; Lunelli et al., 2009), as well as agent-based models such as EpiFast (Bisset et al., 2009), GSAM (Parker and Epstein, 2011), and FluTE (Chao et al., 2010). Any of these simulators can be used in TDEFSI to generate synthetic training data. In this work, we adopt an agent-based simulator EpiFast (Bisset et al., 2009). The outputs are individual infections with their days of being infected in a simulated season. They can be aggregated to any temporal and spatial scale, such as daily (weekly) state (county) level ILI incidence. Vaccine intervention I_V can be implemented in EpiFast simulations, by specifying the quantity of vaccines applied to the population in each week. Next we describe how to estimate a distribution on the parameter space $\mathcal{P}(p_E, p_I, \tau, N_I, I_V)$ from CDC historical data, where N_I denotes the initial number of infections. In our simulations, N_I of the population are infectious while all the rest are susceptible at the beginning of the simulation.

Disease model parameter space. Of the parameters, (p_E, p_I) can be taken from literature (Marathe et al., 2011). We assume that each of (τ, N_I, I_V) follows a distribution that can be estimated from historical data. For clarity, we define an epidemiological week in a calendar year as **ew**, and a seasonal week in a flu season as **sw**, where $ew(40)$ is $sw(1)$. The historical time series of CDC surveillance data (refers to historical training data) used to construct parameter space is split into seasons at $ew(40)$ of each year. That is, each flu season starts from $ew(40)$ of a calendar year and ends in $ew(39)$ of the next year. Note that this applies to the USA, but **sw** may be specified differently for other countries.

We want to highlight that the number of clinically attended cases and the reported or tested cases are lower than the actual number of cases in the population. Additionally, reporting rates can vary between regions. To address the gap between ILINet case count and population case count, we scale the former with a scaling factor, called surveillance ratio. The ratio is different among different states. We assume that the ratio between ILI cases captured by CDC ILINet (denoted ILITOTAL) and ILI cases in the population (ILIPOP) is the same as that between patients of all diseases captured by CDC ILINet (TOTALPATIENT) and patients of all diseases in the population (PATIENTPOP). We approximate PATIENTPOP with all doctor visit data from AHRQ (AHRQ, 2017). The doctor visit data provides county level counts for total hospital visits in a year which is aggregated to state level counts later. Note that it is an underestimate. From surveillance ratio $= \frac{\text{ILITOTAL}}{\text{ILIPOP}} = \frac{\text{TOTALPATIENT}}{\text{PATIENTPOP}}$, we can derive the only unknown ILIPOP. Table 3.1 presents the surveillance ratios for all the states.

Firstly, we collect observations of each parameter value as follows:

- **Initial Case Number (N_I)**: we collect the ILI incidence of $sw(1)$ of each season for the target state and its neighboring states (i.e., geographically contiguous states). For example, given a state New Jersey, the training data includes 6 seasons from 2010-2011 to 2015-2016, its neighbors are Delaware, New York, and Pennsylvania. Then we can collect $6 * 4 = 24$ samples of N_I for NJ.

Table 3.1: Surveillance ratios for each state in the US.

Alabama: 0.0759	Kansas: 0.1093	New York: 0.1204
Alaska: 0.1143	Kentucky: 0.1114	North Carolina: 0.0875
Arizona: 0.0723	Louisiana: 0.0931	North Dakota: 0.1960
Arkansas: 0.0894	Maine: 0.1931	Ohio: 0.1339
California: 0.0628	Maryland: 0.0755	Oklahoma: 0.1039
Colorado: 0.0764	Massachusetts: 0.1380	Oregon: 0.1050
Connecticut: 0.1047	Michigan: 0.1356	Pennsylvania: 0.1299
Delaware: 0.1030	Minnesota: 0.0898	Rhode Island: 0.0932
District of Columbia: 0.1852	Mississippi: 0.0874	South Carolina: 0.0663
Florida: 0.0582	Missouri: 0.1492	South Dakota: 0.1882
Georgia: 0.0701	Montana: 0.1739	Tennessee: 0.0811
Hawaii: 0.0705	Nebraska: 0.1329	Texas: 0.0738
Idaho: 0.1190	Nevada: 0.0643	Utah: 0.0913
Illinois: 0.1066	New Hampshire: 0.1566	Vermont: 0.2111
Indiana: 0.1215	New Jersey: 0.0692	Virginia: 0.0914
Iowa: 0.1420	New Mexico: 0.1258	Washington: 0.0885
Kansas: 0.1093	New York: 0.1204	West Virginia: 0.1684

- **Vaccine Intervention (I_V)**: we collect vaccination schedules of the past influenza seasons in the USA (CDC, 2018). Each schedule consists of timing and percentage coverage of vaccine application throughout the season. Vaccine efficacy (reduction of disease transmission probability) and compliance rate (probability that a person will take the vaccine) are set according to a survey used in (Wang et al., 2019a), which is conducted by Gfk.com, under the National Institutes of Health grant no. 1R01GM109718. This survey collects data on demographics of the respondents and their preventive health behaviors during a hypothetical influenza outbreak. We assume that each person follows a common compliance rate, and the state level vaccine schedule is the same as the nationwide schedule.

- **Transmissibility (τ)**: First we compute the overall attack rate (i.e., the fraction of population getting infected in the season) of each historical season for the target state and its neighboring states. Then for each attack rate ar, say of season s and state r, we calibrate a transmissibility value as the solution to $\min_\tau |AR(EpiFast(\tau, P_E, P_I, N_I, I_V)) - ar|$ using Nelder-Mead (Nelder and Mead, 1965) algorithm, where p_E and p_I are sampled for each person from the distributions shown in Table 3.2; N_I is the initial case number of season s and state r; I_V is the vaccination schedule for season s; $EpiFast(\cdot)$ is a simulation run on the population of state j with the parameters $(\tau, P_E, P_I, N_I, I_V)$; and $AR(\cdot)$ computes attack rate from the output of $EpiFast(\cdot)$. Details of this process are shown in Algorithm 2.

Secondly, for τ and N_I, we fit the collected samples to several distributions including normal, uniform. Then we run KS-test (the null hypothesis being that the sample

Algorithm 2: Calibrating disease model parameter τ

Input: Simulator PS, CDC historical data $histCDC$, and synthetic social contact networks $Network$.

Output: Calibrated τ^*.

$p_E \sim \{1:0.3, 2:0.5, 3:0.2\}$ (Marathe et al., 2011; Wang et al., 2019a);
$p_I \sim \{3:0.3, 4:0.4, 5:0.2, 6:0.1\}$ (Marathe et al., 2011; Wang et al., 2019a);
$I_V = \emptyset$;
$regions$ = {state and its adjacent neighbors};
$seasons$ = {available seasons of $histCDC$};
$\tau^* = \emptyset$;
for r *in* $regions$ **do**
    for s *in* $seasons$ **do**
        $totalili_{(r,s)} = TOTAL(histCDC_{(r,s)})$ `// Computing total ILI incidence for region` r `of season` s
        $ar_{(r,s)} = \frac{totalili_{(r,s)}}{population_{(r)}}$ `// Computing attack rate for region` r `of season` s
        $\tau^*_{(r,s)} = \min_\tau |AR(EpiFast(\tau, P_E, P_I, I_V, N_{I(r,s)}, Network_{(r,s)})) - ar_{(r,s)}|$ `// Calibrating transmissibility` τ `using EpiFast`
        $\tau^* = \tau^* \cup \tau^*_{(r,s)}$
    end
end

is drawn from the reference distribution) to choose a distribution with the highest significance (p-value). For I_V, we assume the six vaccination schedules follow a discrete uniform distribution. In this way, a region-specific parameter space $\mathcal{P}$ is constructed. The learned parameter space is shown in Table 3.2. Note that each parameter in $\mathcal{P}$ follows a marginal distribution.

Synthetic training dataset. For each simulation run, a specific parameter setting is sampled from $\mathcal{P}$, and the simulator is called to generate daily individual health states. These individual health states are aggregated to get state and county level weekly incidences, called *synthetic epicurves*. Week 1 in the synthetic epicurve corresponds to $sw(1)$ of a flu season. Large volumes of high-resolution synthetic data are generated by repeating the sampling and simulating process. Let us denote all simulated epicurves by $\Omega = \{(\mathbf{y}_{(i)}, \mathbf{y}^{\mathcal{D}}_{(i)}) \in \mathbb{R}^{\ell \times (K+1)} | i = 1, 2, \cdots, r\}$, where ℓ is the length of an epicurve (number of weeks), K is the number of counties in the state, and r is the total number of simulation runs. Algorithm 3 describes the generating process.

Compared with CDC surveillance data, the training dataset Ω is prominent in two aspects: (i) it includes high-resolution spatial dependencies between subregions; (ii) the large volume of synthetic training data reduces the possibility of overfitting when

Table 3.2: Marginal distributions of the parameter spaces for VA and NJ. $\mathcal{N}$ denotes normal distribution, $\mathcal{U}$ denotes uniform distribution.

Parameter	State	Name	Distribution	P-value
p_E	VA	Discrete distribution	(1:0.3, 2:0.5, 3:0.2) Marathe et al. (2011); Wang et al. (2019a)	-
	NJ	Discrete distribution	(1:0.3, 2:0.5, 3:0.2) Marathe et al. (2011); Wang et al. (2019a)	-
p_I	VA	Discrete distribution	(3:0.3, 4:0.4, 5:0.2, 6:0.1) Marathe et al. (2011); Wang et al. (2019a)	-
	NJ	Discrete distribution	(3:0.3, 4:0.4, 5:0.2, 6:0.1) Marathe et al. (2011); Wang et al. (2019a)	-
τ	VA	Normal	$\mathcal{N}(\mu = 4.88\text{e}{-5}, \delta = 9.33\text{e}{-7})$	0.74
	NJ	Normal	$\mathcal{N}(\mu = 4.63\text{e}{-5}, \delta = 1.05\text{e}{-6})$	0.85
N_I	VA	Uniform	$\mathcal{U}(7355, 16278)$	0.85
	NJ	Uniform	$\mathcal{U}(567, 7647)$	0.40
I_V	VA	Discrete uniform	6 vaccination schedules CDC (2018)	-
	NJ	Discrete uniform	6 vaccination schedules CDC (2018)	-

The null hypothesis for the two-sample KS test is that both groups were sampled from populations with identical distributions. If the p-value returned by the KS test is less than a significance level, we reject the null hypothesis. In our experiments, we do not specify a significance level but instead choose the distribution with the largest p-value among multiple assumed distributions.

Algorithm 3: Generating Training Dataset for TDEFSI

Input: Simulator PS, and Parameter space $\mathcal{P}$.
Output: Simulated epicurves $\Omega = \{(\mathbf{y}_{(i)}, \mathbf{y}^{\mathcal{D}}_{(i)}) | i = 1, 2, \cdots, r\}$.

$\Omega = \emptyset$;
for $i = 1$ *to* r **do**
  $P = \text{Sample}(\mathcal{P})$ `// Parameter setting sampling`
  $(\mathbf{y}_{(i)}, \mathbf{y}^{\mathcal{D}}_{(i)}) = \text{PS}(P)$ `// Simulating based on a sampled setting` P
  $\Omega = \Omega \cup (\mathbf{y}_{(i)}, \mathbf{y}^{\mathcal{D}}_{(i)})$
end

training a DNN model. Thus the trained model has better generalization ability.

TDEFSI - a deep neural network model. LSTM (Hochreiter and Schmidhuber, 1997) is adopted in our neural network architecture to capture the inherent temporal dependency in the weekly incidence data. Figure 3.2 shows unrolled k-stacked LSTM layers. Each LSTM layer consists of a sequence of cells. The number of cells depends on the number of input time points. In this figure, the input is a time series of $y_1, ..., y_{t-1}$, the output comprises all the cell outputs $\mathbf{h}^{(k)}$ from the last layer k ("last" depth-wise, not time-wise). Each LSTM layer consists of $t-1$ cells. In the first LSTM layer (layer 0), a cell will work as described in 3.1, e.g., cell 2 takes y_1, cell state $\mathbf{c}_1^{(0)}$ and cell output $\mathbf{h}_1^{(0)}$ from the previous cell 1 as inputs, then outputs $(\mathbf{c}_2^{(0)}, \mathbf{h}_2^{(0)})$ so you could feed them into the next cell and feed $\mathbf{h}_2^{(0)}$ into the next layer (layer 1). The first LSTM layer takes $y_1, ..., y_{t-1}$ as the input, the second layer takes $\mathbf{h}_1^{(0)}, ..., \mathbf{h}_{t-1}^{(0)}$ as the input, and the rest of the layers behave in the same manner.

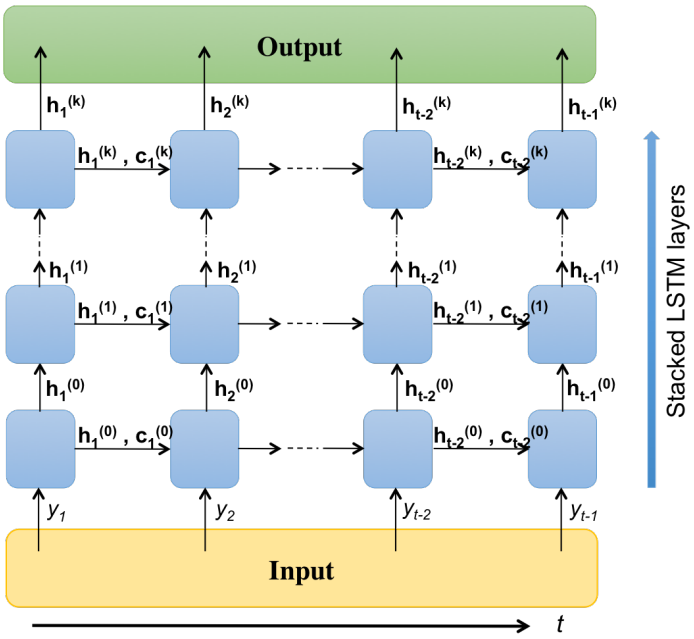


Figure 3.2: Unrolled k-stacked LSTM layers. Each LSTM layer consists of a sequence of cells. The number of cells depends on the number of input time points. In this figure, the input is a time series of $y_1, ..., y_{t-1}$, the output comprises all the cell outputs $\mathbf{h}^{(k)}$ from the last layer k ("last" depth-wise, not time-wise). Each LSTM layer consists of $t-1$ cells. In the first LSTM layer, a cell will work as described in 3.1, e.g., cell 2 takes y_1, cell state $\mathbf{c}_1^{(0)}$ and cell output $\mathbf{h}_1^{(0)}$ from the previous cell 1 as inputs, then outputs $(\mathbf{c}_2^{(0)}, \mathbf{h}_2^{(0)})$ so you could feed them into next cell and feed $\mathbf{h}_2^{(0)}$ into next layer. The first LSTM layer take $y_1, ..., y_{t-1}$ as the input, the second layer take $\mathbf{h}_1^{(0)}, ..., \mathbf{h}_{t-1}^{(0)}$ as the input, and rest of the layers behave in the same manner.

Let $H^{(i)}, 0 \leq i \leq k$ be the dimension of the hidden state in layer i. For the first layer, assume the input of the current cell is y_{t-1}. Then the computation within the cell is described mathematically as:

$$
\begin{aligned}
\mathbf{i}_{t-1}^{(0)} &= \sigma(\mathbf{W}_i^{(0)} \cdot y_{t-1} + \mathbf{U}_i^{(0)} \cdot \mathbf{h}_{t-2}^{(0)} + \mathbf{b}_i^{(0)}) \in \mathbb{R}^{H^{(0)}} \\
\mathbf{f}_{t-1}^{(0)} &= \sigma(\mathbf{W}_f^{(0)} \cdot y_{t-1} + \mathbf{U}_f^{(0)} \cdot \mathbf{h}_{t-2}^{(0)} + \mathbf{b}_f^{(0)}) \in \mathbb{R}^{H^{(0)}} \\
\mathbf{o}_{t-1}^{(0)} &= \sigma(\mathbf{W}_o^{(0)} \cdot y_{t-1} + \mathbf{U}_o^{(0)} \cdot \mathbf{h}_{t-2}^{(0)} + \mathbf{b}_o^{(0)}) \in \mathbb{R}^{H^{(0)}} \\
\widetilde{\mathbf{C}}_{t-1}^{(0)} &= tanh(\mathbf{W}_C^{(0)} \cdot y_{t-1} + \mathbf{U}_C^{(0)} \cdot \mathbf{h}_{t-2}^{(0)} + \mathbf{b}_C^{(0)}) \in \mathbb{R}^{H^{(0)}} \\
\mathbf{C}_{t-1}^{(0)} &= \mathbf{f}_{t-1}^{(0)} \circ \mathbf{C}_{t-2}^{(0)} + \mathbf{i}_{t-1}^{(0)} \circ \widetilde{\mathbf{C}}_{t-1}^{(0)} \in \mathbb{R}^{H^{(0)}} \\
\mathbf{h}_{t-1}^{(0)} &= \mathbf{o}_{t-1}^{(0)} \circ \mathbf{C}_{t-1}^{(0)} \in \mathbb{R}^{H^{(0)}}
\end{aligned}
\tag{3.1}
$$

where σ and $tanh$ are sigmoid and tanh activation functions. $\mathbf{W} \in \mathbb{R}^{H^{(0)}}, \mathbf{U} \in$

$\mathbb{R}^{H^{(0)}\times H^{(0)}}$, and $\mathbf{b} \in \mathbb{R}^{H^{(0)}}$ are learned weights and bias. $\mathbf{C}_{t-2}^{(0)}, \mathbf{h}_{t-2}^{(0)}$ are the cell state and output of the previous cell. Operator $\circ$ denotes element wise product (Hadamard product). The cell computation is similar in the layer i, but with y_{t-1} being replaced by $\mathbf{h}_{t-1}^{(i-1)} \in \mathbb{R}^{H^{(i-1)}}$, and $\mathbf{W} \in \mathbb{R}^{H^{(i)}\times H^{(i-1)}}$.

In traditional time series models, ILI incidences of the previous few weeks are used as the observations for the forecast of the current week. In TDEFSI, we use two kinds of observations: (i) ***Within-season observations***, denoted as $\mathbf{x1} = \langle y_{t-a}, \cdots, y_{t-1}\rangle$, are ILI incidence from previous a weeks which are back from time step t. (ii) ***Between-season observations***, denoted as $\mathbf{x2} = \langle y_{t-\ell*b}, \cdots, y_{t-\ell*1}\rangle$, are ILI incidences of the same sw from the past b seasons. They are used as the surrogate information to improve forecasting performance. As shown in Figure 3.3, for example, there are 4 seasons ordered by sw. The within-season observations are ILI incidence of previous $a = 3$ weeks in current season. The between-season observations are ILI incidence of the same $sw(t)$ from the past $b = 3$ seasons.

In TDEFSI model, we design a two-branch LSTM-based model to capture temporal dynamics of within-season and between-season observations. As shown in Figure 3.4, the left branch consists of stacked LSTM layers that encode within-season observations $\mathbf{x1} = \langle y_{t-a}, \cdots, y_{t-1}\rangle$. The right branch is also LSTM-based and encodes between-season observations $\mathbf{x2} = \langle y_{t-\ell*b}, \cdots, y_{t-\ell*1}\rangle$. A merge layer is added to combine the outputs of two branches. The final output is $\hat{\mathbf{z}}_t$ which consists of state level and county level forecasts (as defined in Section 3.2.3).

In the left branch, the output of the Dense layer is:

$$\mathbf{O}_l = \psi_l(\mathbf{w}_l \cdot \mathbf{h}_{t-1}^{(k_l)} + \mathbf{b}_l) \in \mathbb{R}^H \tag{3.2}$$

where k_l is the number of LSTM layers in the left branch, H is the dimension of output of the left branch, $\mathbf{w}_l \in \mathbb{R}^{H\times H^{(k_l)}}$ and $\mathbf{b}_l \in \mathbb{R}^H$, ψ_l is the activation function.

Similarly, the output of the Dense layer in the right branch is:

$$\mathbf{O}_r = \psi_r(\mathbf{w}_r \cdot \mathbf{h}_{t-1}^{(k_r)} + \mathbf{b}_r) \in \mathbb{R}^H \tag{3.3}$$

where k_r is the number of LSTM layers in the right branch, H is the dimension of output of the right branch, $\mathbf{w}_r \in \mathbb{R}^{H\times H^{(k_r)}}$ and $\mathbf{b}_r \in \mathbb{R}^H$, ψ_r is the activation function.

The merge layer combines the output from two branches by addition, denoted as:

$$\hat{\mathbf{z}}_t = \psi(\mathbf{w}[\mathbf{O}_l \oplus \mathbf{O}_r] + \mathbf{b}) \in \mathbb{R}^{K+1} \tag{3.4}$$

where $\mathbf{w} \in \mathbb{R}^{(K+1)\times H}$, $\mathbf{b} \in \mathbb{R}^{K+1}$, ψ is the activation function, and $\oplus$ denotes the element-wise addition.

This LSTM-based deep neural network model is able to connect historical ILI incidence information to the current forecast. It also allows long-term dependency learning without suffering the gradient vanishing problem. The number of LSTM

layers is a hyperparameter that we tuned by grid searching.

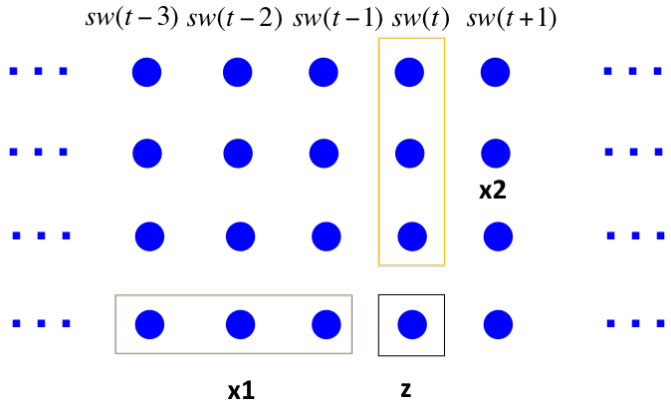


Figure 3.3: Within-season and between-season observations as the input for the TDEFSI neural network model. In this graph, there are four flu seasons (rows). Nodes in each row denote weekly ILI incidence in each season, which are ordered by sw. For a target week $sw(t)$ (black square), the model observes two kinds of information: (i) within-season observations **x1** - the ILI incidence from the previous weeks back from week $sw(t)$ (green rectangular); (ii) between-season observations **x2** - the historical ILI incidence from similar weeks of the past seasons (yellow rectangular). **z** is the target week of ILI forecasting. **x1** and **x2** are state level ILI, while **z** includes state and county level ILI.

We are interested in a predictor f, which predicts the current week's state level and county level incidence $\mathbf{z}_t$ based on the previous a weeks of within-season state level ILI incidence **x1** and the previous b seasons of between-season state level ILI incidence **x2**:

$$\hat{\mathbf{z}}_t = f([\mathbf{x1}, \mathbf{x2}]_t, \theta) \tag{3.5}$$

where θ denotes parameters of the predictor, $\hat{\mathbf{z}}_t$ denotes the forecast of $\mathbf{z}_t$. Note that **the output of f is always one week ahead forecast** in our model.

The optimization objective is:

$$\min_{\theta} \mathcal{L}(\theta) = \sum_{t} \|\mathbf{z}_t - f([\mathbf{x1}, \mathbf{x2}]_t, \theta)\|_2^2 + \mu\phi(\hat{\mathbf{z}}_t) + \lambda\delta(\hat{\mathbf{z}}_t), \tag{3.6}$$

where $\phi(\hat{\mathbf{z}}_t)$ is an activity regularizer added to the outputs for spatial consistency constraint $\hat{y}_t = \sum_{C \in \mathcal{D}} \hat{y}_t^C$:

$$\phi(\hat{\mathbf{z}}_t) = \left| \hat{y}_t - \sum_{C \in \mathcal{D}} \hat{y}_t^C \right|, \tag{3.7}$$

and $\delta(\hat{\mathbf{z}}_t)$ is an activity regularizer added to the outputs for non-negative consistency

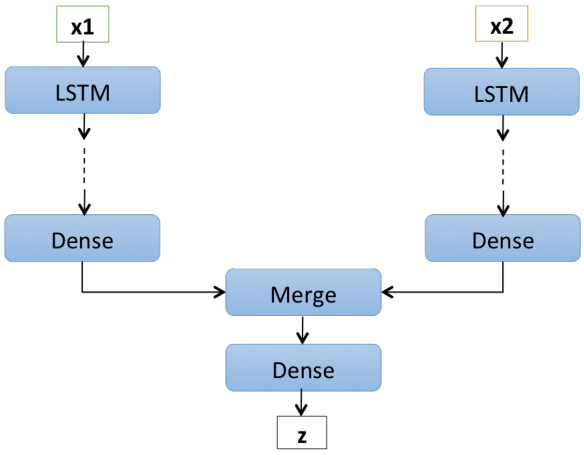


Figure 3.4: TDEFSI model architecture. This architecture consists of two branches. The left branch consists of stacked LSTM layers that encodes state level within-season observations **x1**, and the right branch consists of stacked LSTM layers that encodes state level between-season observations **x2**. A merge layer is added to combine two branches and the output **z** is the state and county level forecasts.

constraint $\hat{y}_t, \hat{y}_t^C \geq 0$:

$$\delta(\hat{\mathbf{z}}_t) = \left| \frac{1}{K+1} \sum \max(-\hat{\mathbf{z}}_t, \mathbf{0}) \right|, \tag{3.8}$$

μ, λ are two pre-specified hyperparameters, $\min(\hat{\mathbf{z}}_t, \mathbf{0})$ returns element-wise minimum value, K is the number of counties in the state, $\delta(\hat{\mathbf{z}}_t)$ returns the absolute mean of element-wise minimum values. The Adam optimization algorithm (Kingma and Ba, 2014) is used to learn θ. How the activity regularizers affect the model performance will be discussed in Section 3.2.5.

Variants of TDEFSI. The two-branch neural network architecture has multiple variants: (*i*) *TDEFSI*: A two-branch neural network as shown in Figure 3.4. (*ii*) *TDEFSI-LONLY*: Only the left branch is used to take within-season observations. (*iii*) *TDEFSI-RDENSE*: The left branch comprises of stacked LSTM layers, while the right branch uses Dense layers, which means that the model does not care about the temporal relationship between between-season data points. We will discuss the results of different variants in Section 3.2.5.

Training and forecasting. In the training process, we use synthetic training data Ω to train the TDEFSI models. The historical surveillance data is only used for constructing the disease model parameter space $\mathcal{P}$. In the predicting step, the trained model takes state level surveillance as input and makes one week ahead forecasts at both state and county levels. TDEFSI models are trained once before the target flu

season starts, then can be used for forecasting throughout the season.

In practical situations, we are interested in making forecasts for several weeks ahead using iterative method. In TDEFSI, the left branch of the model appends the most recent state level forecast to the input for predicting the target of the next week, and the right branch uses the state level ILI incidences from the past seasons with sw equal to the next week number.

3.2.5 Experiments

Datasets. CDC ILI incidence (CDC, 2019b): The CDC surveillance data used in the experiments is the weekly ILI incidence at state level from 2010 $ew(40)$ to 2018 $ew(18)$. Note that it may be revised continuously until the end of a flu season. We use the finalized data in this work. **ILI Lab tested flu positive counts of New Jersey** (DOH, 2019): To evaluate the county level forecasting performance, we collect state level and county level ILI Lab tested flu positive counts of season 2016-2017 and 2017-2018 in NJ. The data is available from $ew(40)$ to the next year's $ew(20)$. We use it as the ground truth when evaluating county level forecasting. **Google data** (Google, 2018; GHT, 2018): The Google correlate terms (keyword: influenza) of each state are queried; we choose the top 100 terms. Then the Google Health Trends of each correlated term for each state is collected and aggregated weekly from 2010 $ew(40)$ to 2018 $ew(18)$. **Weather data** (CDO, 2018): we download daily weather data (including max temperature, min temperature, precipitation) from Climate Data Online (CDO) for each state and compute weekly data as the average of daily data from 2010 $ew(40)$ to 2018 $ew(18)$. Google data and weather data are used as surrogate information in comparison methods. **Simulated data**: For each state, we generate 1000 simulated curves of weekly ILI incidence at both state level and county level. Of each curve, the first week $sw(1)$ corresponds to epi-week 40 $ew(40)$ of real seasonal curves.

We divide the real data into: *real-training:* the beginning 80% of season 2010-2011 to season 2015-2016 (251 data points per state). *real-validating:* the last 20% of season 2010-2011 to season 2015-2016 (63 data points per state). *real-testing:* season 2016-2017 to season 2017-2018 (83 data points per state). *County level real-evaluating:* county level ILI lab tested flu positive counts for NJ (64 data points per county of NJ). For TDEFSI models, we use the training dataset to learn disease parameter space, while for baselines, we use training dataset to train the model directly and use validating dataset to validate and choose the final models. Testing and county level evaluating datasets are used for all methods to evaluate their performance. And the final result of each method is the average value of 10 trials.

We divide the simulated data into: *sim-training:* 80% of 1000 simulated curves. *sim-validating:* 15% of 1000 simulated curves. *sim-testing:* 5% of 1000 simulated curves. The synthetic data is only used for training and validating of TDEFSI models.

No baselines are applied for synthetic data.

Baselines. Our method is compared with state-of-the-art deep learning methods, statistical methods, and mechanistic methods. They are:

- *LSTM* (CDC data) (Hochreiter and Schmidhuber, 1997) and *AdapLSTM* (CDC + weather data) (Venna et al., 2019) representing deep learning methods;
- *SARIMA* (CDC Data) (Benjamin et al., 2003) and *ARGO* (CDC + Google data) (Yang et al., 2015b) representing statistical methods; and
- *EpiFast* (Beckman et al., 2014) representing mechanistic methods.

AdapLSTM, LSTM, ARGO, and SARIMA can make flat-resolution forecasting directly from the model, then flat-resolution forecasts can be turned into high-resolution forecasts by multiplying by county level population proportions. EpiFast is applied for both flat-resolution and high-resolution forecasting directly.

Experiment setup. In this section, we describe the experiment settings, including simulation setting and TDEFSI model setting. Note that we conduct the experiments on two states of the USA i.e. VA and NJ. State level forecasting performance will be evaluated on both VA and NJ, while county level forecasting performance is evaluated on NJ only due to the limitation on the availability of high-resolution observations.

Disease model settings for generating simulated training data: The simulation parameter settings are listed in Table 3.2. The length of a simulated epicurve is set to $\ell = 52$, and the total runs of simulations is $r = 1000$. We adopt EpiFast as the simulator, PS='EpiFast'.

TDEFSI model settings: We set up the architectures for TDEFSI and its variants as follows:

- *TDEFSI*: The left branch consists of two stacked LSTM layers, one dense layer; the right branch consists of one LSTM layer, one dense layer. $k_l = 2$, $k_r = 1$, $H^{(k_l)} = H^{(k_r)} = 128$, $H = 256$, ψ_l, ψ_r, ψ are linear functions.
- *TDEFSI-LONLY*: The left branch consists of two stacked LSTM layers, one dense layer and no right branch. $k_l = 2$, $H^{(k_l)} = 128$, $H = 256$, ψ_l, ψ are linear functions.
- *TDEFSI-RDENSE*: The left branch consists of two stacked LSTM layers, one dense layer; the right branch consists of one dense layer. $k_l = 2$, $k_r = 0$, $H^{(k_l)} = H^{(k_r)} = 128$, $H = 256$, ψ_l, ψ_r, ψ are linear functions.

For all TDEFSI models, we set $a = 52$, $b = 5$, $(\mu, \lambda)_{VA} = (0.1, 0.1)$, $(\mu, \lambda)_{NJ} = (1, 0.01)$. We use Adam optimizer with all default values. We choose the final model using grid searching with sim-validating dataset. The grid searching space is about 500 models, including $a(10, 20, 30, 40, 50)$, $b(5)$, $\mu(0, 0.001, 0.01, 0.1, 1)$, $\lambda(0, 0.001, 0.01, 0.1, 1)$,

$k_l(1,2)$, $H(128,256)$. In the training process, the best models are selected by early stopping when the validation accuracy does not increase for 50 consecutive epochs, and the maximum epoch number is 300. Unless explicitly noted, in our experiments, these hyperparameters are set with the values described above.

Baselines settings: We elaborate the details of model setting of the baselines. Note that, in the experiments, we choose the final model with the best validation accuracy by grid searching. Unless explicitly noted, the hyperparameters are set with default values from python libraries.

- *LSTM*: It consists of one LSTM layer and one dense layer. The input is the sequence of state level ILI incidence and the output is the state level forecast of the current week. By grid searching, we set the look back window size to 52 and LSTM hidden units to 128. The Adam optimizer is used.

- *AdapLSTM* (Venna et al., 2019): This method makes forecasts using a simple LSTM model, then adjusts the forecasts by applying impacts of weather factors and spatiotemporal factors. The LSTM model has the same setting with single layer LSTM model described above. In (Venna et al., 2019), the weather features include maximum temperature, minimum temperature, humidity, and precipitation. However, humidity is not used in our experiments since it is not publicly available in the collected weather dataset. The confidences of symbol pairs (the climatic variable time series and the flu count time series) in our experiment are less than 0.3, which will lead to arbitrary adjustment for forecasts. The neighbors of each state used for spatiotemporal adjustment factor are geographical adjacent states that are the same with those used in constructing disease parameter space. For more details please refer to the original paper (Venna et al., 2019).

- *SARIMA*: We use the Seasonal ARIMA model, denoted as $SARIMA(p,d,q) \times (P,D,Q)_m$, where p is the order (number of time lags) of the autoregressive model, d is the degree of differencing (the number of times the data have had past values subtracted), q is the order of the moving-average model, m refers to the number of periods in each season, and the uppercase P, D, Q refer to the autoregressive, differencing, and moving average terms for the seasonal part of the SARIMA model. By grid searching, the selected model is $SARIMA(8,1,0) \times (5,0,0)_{52}$. No exogenous variables are used in this model.

- *ARGO* (Yang et al., 2015b): The method uses an autoregression model utilizing Google search data. We use the publicly available tool from (Yang et al., 2015b). In our experiment, we set the look back window size to 52 and the training window to 104. In the Google data we collected, all of the top 100 Google correlate terms of VA are flu related, while only one out of the top 100 Google correlated terms of NJ are flu related.

- *EpiFast* (Beckman et al., 2014): This method takes the same setting of p_E and p_I as shown in Table3.2, and searches for N_I, τ by minimizing the dissimilarity between the predicted and the actual ILI incidence using the Nelder-Mead algorithm (Nelder and Mead, 1965).

Experimental setup for testing on real seasonal ILI dataset: In these experiments, we evaluate TDEFSI models and all comparison methods. The experiments are performed on two states: Virginia (VA) and New Jersey (NJ). The county level evaluation is conducted on NJ counties. For TDEFSI and its variants, the real-training dataset is used to estimate disease parameter space, while for all baselines, real-training and real-validating are used for training directly. The county level real-evaluating dataset is only used for evaluation of the performance of county level forecasts. At each time step in the testing season, each model makes forecasts up to five weeks ahead, i.e. $horizon = \{1, 2, 3, 4, 5\}$.

Metrics. The metrics used to evaluate the forecasting performance are: *root mean squared error (RMSE)* (see Equation 2.7), *mean absolute percentage error (MAPE)* (see Equation 3.9), *Pearson correlation (PCORR)* (see Equation 2.8). Assuming we have n testing data points and $n = N \times m$ means N locations by m weeks. We denote the true value and forecast for the ith testing data point to be z_i and $\hat{z}_i$. We do not distinguish locations in calculating RMSE and MAPE. PCORR is calculated by locations and the final value is the average of all locations. Among these metrics, RMSE and MAPE evaluate ILI incidence forecasting accuracy, PCORR evaluates linear correlation between the true curve and the predicted curve.

- **Mean absolute percentage error (MAPE)**:

$$\text{MAPE} = (\frac{1}{n}\sum_{i=1}^{n}|\frac{z_i - \hat{z}_i}{z_i + 1}|) * 100 \tag{3.9}$$

  where the denominator is smoothed by 1 to avoid zero values. MAE ranges in $[0, +\infty]$ and smaller values are better.

Performance of flat-resolution forecasting. We forecast state level ILI incidence on real-testing dataset for VA and NJ. Table 3.3 shows the performance on RMSE, MAPE, PCORR for (a) VA and (b) NJ with horizon=$\{1, 2, 3, 4, 5\}$. Figure 3.5 presents the overall performance across all states, weeks, horizons. (i) *Performance on RMSE*: In VA, TDEFSI, TDEFSI-LONLY, TDEFSI-RDENSE, SARIMA, ARGO, and LSTM achieve similar performance that is better than EpiFast and AdapLSTM. Compared with other methods, AdapLSTM does not perform well with small horizons while EpiFast has poor performance with large horizons. In NJ, TDEFSI, TDEFSI-LONLY, and TDEFSI-RDENSE consistently outperform others across the horizon. Overall, TDEFSI and its variants slightly outperform comparison methods in RMSE.

Table 3.3: State level performance across season 2016-2017 and 2017-2018 for VA and NJ with horizon = 1, 2, 3, 4, 5. The best value is marked in bold, and the second-best value is marked with underline.

	VA					NJ				
RMSE	1	2	3	4	5	1	2	3	4	5
SARIMA	**824**	1463	2059	2440	2682	218	464	690	891	1050
ARGO	1073	1592	2072	2444	2580	313	512	717	760	874
LSTM	1083	1629	**2013**	2273	**2438**	240	470	699	902	1070
AdapLSTM	2012	2038	2264	2382	2449	586	729	640	871	1006
EpiFast	1300	2087	2989	3674	4284	238	382	567	725	871
TDEFSI	1000	**1447**	2014	**2358**	2544	**174**	**344**	511	665	757
TDEFSI-LONLY	900	1572	2119	2582	2742	197	373	531	696	801
TDEFSI-RDENSE	1109	1686	2136	2421	2540	193	358	**506**	**630**	**711**
MAPE	1	2	3	4	5	1	2	3	4	5
SARIMA	**15.96**	**32.57**	**50.62**	**65.60**	77.94	**13.28**	24.32	35.62	48.32	59.99
ARGO	31.06	54.00	73.69	78.97	77.85	24.96	33.14	44.52	50.05	54.60
LSTM	38.40	49.29	58.80	67.98	71.00	39.44	78.53	131.19	189.79	243.40
AdapLSTM	42.67	51.22	61.02	67.33	**70.60**	64.30	64.77	65.56	74.14	76.50
EpiFast	31.14	53.45	84.32	124.05	167.44	30.32	32.40	50.75	64.61	76.27
TDEFSI	25.75	40.69	58.61	74.06	88.95	18.16	29.74	43.49	55.12	66.09
TDEFSI-LONLY	22.40	35.18	59.27	89.95	123.70	15.56	32.21	45.74	60.46	72.13
TDEFSI-RDENSE	31.89	51.69	76.94	101.38	125.23	15.17	**21.74**	**29.19**	**37.95**	**44.14**
PCORR	1	2	3	4	5	1	2	3	4	5
SARIMA	0.9461	0.8271	0.6468	0.4925	0.3788	0.9541	0.8173	0.6421	0.4611	0.3195
ARGO	**0.9590**	0.8728	**0.7219**	0.4518	0.3218	0.9444	0.8005	0.6043	0.4530	0.2921
LSTM	0.9223	0.7890	0.6350	0.5050	0.4101	0.9603	0.8542	0.6995	0.5340	0.3939
AdapLSTM	0.7048	0.6397	0.5174	0.4307	0.3818	0.8113	0.5912	**0.7686**	0.4477	0.2753
EpiFast	0.8876	0.7665	0.5616	0.3906	0.2340	0.9573	0.8535	0.7044	0.3835	0.2841
TDEFSI	0.9358	0.8487	0.6892	**0.5555**	**0.4647**	**0.9683**	**0.8773**	0.7348	**0.5639**	0.4247
TDEFSI-LONLY	0.9460	**0.8776**	0.7037	0.5074	0.3266	0.9659	0.8697	0.7288	0.4946	0.3245
TDEFSI-RDENSE	0.9043	0.7824	0.6182	0.4409	0.2826	0.9654	0.8692	0.7280	0.5630	**0.4248**

(*ii*) *Performance on MAPE*: In VA, SARIMA performs the best overall among all methods. In NJ, TDEFSI-RDENSE achieves the best performance closely followed by SARIMA. Overall, SARIMA outperforms others, and TDEFSI and its variants achieve similar performance with ARGO which are better than LSTM, AdapLSTM, EpiFast. (*iii*) *Performance on PCORR*: In VA, ARGO performs the best with horizon 1,2,3 and TDEFSI achieves better performance with horizon 4,5. In NJ, TDEFSI performs the best and TDEFSI-LONLY, TDEFSI-RDENSE achieve similar performance. Overall, TDEFSI and its variants slightly outperform SARIMA, ARGO, LSTM, while they are much better than AdapLSTM and EpiFast.

Figure 3.6 shows the *weekly* state level model performance measured on season 2017-2018 using RMSE: The x-axis denotes *ew* number, the value is averaged over 5 horizons. A log y-scale is used. The black vertical line marks the peak week of the season. We observe that these models perform with great variance around the beginning and the end of a season than in weeks near the peak.

In general, the proposed model and its variants achieve comparable or better performance than the comparison methods on the state level ILI forecasting.

Performance of high-resolution forecasting. The performance of county level forecasts is evaluated on NJ counties. Note that EpiFast, TDEFSI, TDEFSI-LONLY,

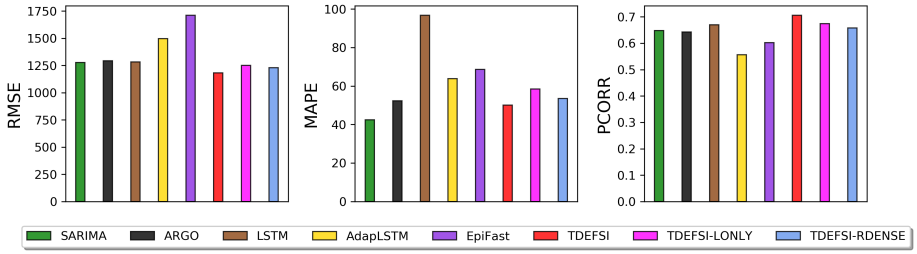
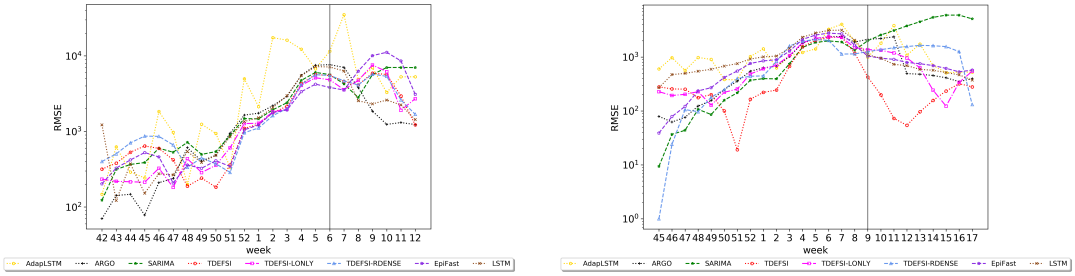


Figure 3.5: State level performance (RMSE, MAPE, PCORR). The value is averaged across two states, two seasons, and 5 horizons.



(a) VA, 2017-2018 (b) NJ, 2017-2018

Figure 3.6: State level performance by weeks (RMSE). (a) VA, 2017-2018; (b) NJ, 2017-2018. TDEFSI and its variants, and all comparison methods are evaluated and compared. The x-axis denotes *ew* number, the value is averaged on 5 horizons. A log y-scale is used. The black vertical line marks the peak week of the season in the state.

TDEFSI-RDENSE make county level forecasts directly from models, while the other baselines obtain county level forecasts by multiplying state level forecasts with county population proportions. Table 3.4 shows the forecasting performance on RMSE, MAPE, PCORR with horizon=$\{1, 2, 3, 4, 5\}$. The value is the average across weeks and counties. Figure 3.7 presents the overall performance across all counties, weeks, horizons. From the table we observe that SARIMA performs well with horizon $=$ 1. TDEFSI consistently outperforms others across horizons, followed by TDEFSI-RDENSE. Among TDEFSI variants, TDEFSI and TDEFSI-RDENSE perform better than TDEFSI-LONLY, which indicates that the between-season observations are helpful for improving forecasting accuracy. The figure shows consistent results with the table. Overall, our method outperforms the comparison methods on the county level forecasting.

Discussion. In general, for state level, AdapLSTM and EpiFast do not perform very well in our experiments compared with other methods. For AdapLSTM, weather features are considered for post adjustment of LSTM outputs. As stated in (Venna et al., 2019), the weather factors are estimated using time delays computed by apriori

Table 3.4: County level performance for counties of NJ with horizon = 1, 2, 3, 4, 5. The value is the average of 21 counties of NJ across season 2016-2017 and 2017-2018. The best value is marked in bold, and the second-best value is marked with underline.

	NJ-Counties				
RMSE	1	2	3	4	5
SARIMA	**30.58**	38.02	48.60	58.92	67.68
ARGO	33.69	39.89	49.61	51.46	57.35
LSTM	33.80	41.95	52.25	61.56	68.30
AdapLSTM	36.67	45.30	39.46	51.70	59.60
EpiFast	34.34	36.74	40.51	47.40	54.09
TDEFSI	35.17	**31.40**	**34.70**	**40.44**	**45.95**
TDEFSI-LONLY	33.13	36.45	42.41	50.63	56.22
TDEFSI-RDENSE	34.79	31.59	35.22	40.98	46.35
MAPE	1	2	3	4	5
SARIMA	575.19	550.74	540.04	525.20	525.57
ARGO	649.32	552.18	498.42	430.74	366.89
LSTM	745.52	876.56	1066.80	1264.64	1417.91
AdapLSTM	584.18	489.51	417.72	599.53	717.61
EpiFast	712.97	632.96	577.74	519.37	487.54
TDEFSI	**260.95**	**247.70**	**209.69**	**270.58**	**308.95**
TDEFSI-LONLY	603.33	528.62	478.08	454.52	435.50
TDEFSI-RDENSE	614.95	499.13	412.68	360.99	315.78
PCORR	1	2	3	4	5
SARIMA	**0.8645**	0.7474	0.5678	0.3806	0.2211
ARGO	0.8606	0.7388	0.5455	0.3922	0.2211
LSTM	0.8611	0.7699	0.6132	0.4234	0.2597
AdapLSTM	0.7260	0.5150	0.6717	0.3710	0.2205
EpiFast	0.8555	0.7762	0.6450	0.3530	0.2133
TDEFSI	0.7877	**0.8500**	**0.7835**	**0.6425**	**0.4710**
TDEFSI-LONLY	0.8499	0.7669	0.6184	0.4146	0.2176
TDEFSI-RDENSE	0.7860	0.8063	0.7056	0.5467	0.3774

associations and selected by the largest confidence. However, in our experiment, they all show very low confidences (less than 0.3). This may cause arbitrary adjustment for forecasts and consequently poor performance. For EpiFast, one possible reason is that we did not find a good estimate of the underlying disease model for a specific region and season due to the noisy CDC observations. If we rank the performance of all methods, ARGO performs slightly better on VA than on NJ. The possible reason is that about 80% of the top 100 Google correlated terms for NJ are irrelevant to flu and most of them have zero frequencies, while the top 100 correlated terms for VA are of good quality. This will give ARGO a better performance on VA than on NJ. Similarly, LSTM performs relatively better on VA than on NJ. One possible reason is that LSTM cannot learn a pattern that has never occurred in the historical observations. So its performance depends on whether a similar epicurve occurred in previous seasons. As

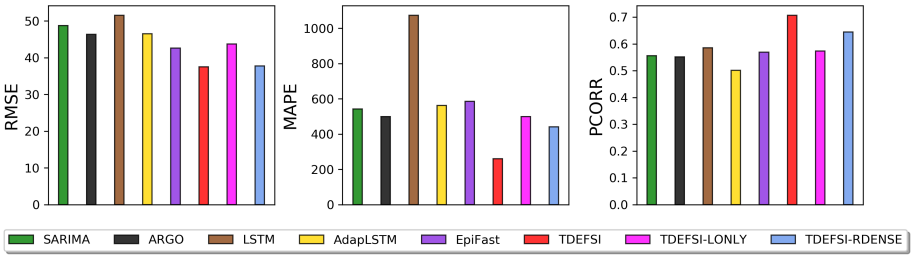


Figure 3.7: County level performance (RMSE, MAPE, PCORR). The value is averaged on two seasons, 5 horizons and 21 counties of NJ.

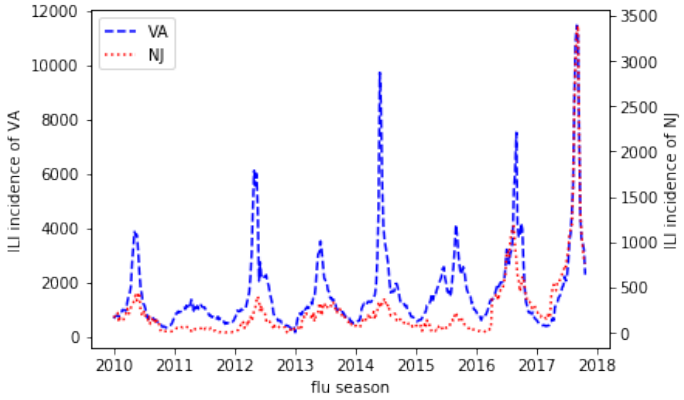


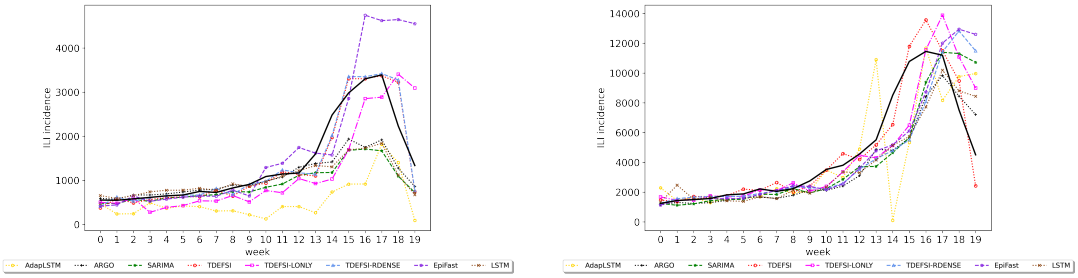
Figure 3.8: CDC surveillance ILI incidence of VA (blue dash line) and NJ (red dot line). It is observable that, for testing season 2017-2018, a similar epi-curve (i.e., similar curve shape and the peak size) occurs at season 2014-2015 in VA, while no similar seasons could be found in NJ.

shown in Figure 3.8, the epicurve of VA 2017-2018 is similar to that of VA 2014-2015, and 2016-2017 is similar to 2012-2013. However, the epicurve of NJ 2017-2018 seems to be much higher than all previous ones, as well as 2016-2017. In actuality, this is the limitation of all data driven models. On the contrary, TDEFSI models have stable performance on both VA and NJ. They manage to avoid overfitting through training on a large volume of synthetic training data. In addition, the simulated training dataset includes many realistic simulated patterns that are unseen in the real world, thus provides a better generalizability to our models.

As seen through the results, TDEFSI enables high-resolution forecasting that outperforms baselines. Meanwhile, it achieves comparable/better performance than the comparison methods at state level forecasting. In the proposed framework, the large volume of realistic simulated data allows us to train a more complex DNN model and reduces the risk of overfitting. My experiments demonstrate that TDEFSI integrates the strengths of ANN methods and causal methods to improve epidemic

forecasting.

Interpretability of TDEFSI. As we discussed above, we found that compared with baselines, TDEFSI performs better on NJ than on VA. A possible reason is that for testing season 2017–2018, a similar epi-curve occurs at season 2014–2015 in VA, while no similar seasons could be found in historical data for NJ. Let's take a closer look at the predicted curves. In Figure 3.9a and 3.9b we show the predicted curves together with the ground truth curve of 2017-2018 flu season for VA and NJ. We can observe that for NJ, TDEFSI and its variants can capture the peak intensity even this was not seen in the history. However, the data-driven baselines (AdapLSTM, ARGO, SARIMA, LSTM) did not capture the new pattern as they never saw the pattern in the historical data. Our methods can forecast unseen patterns because many unseen patterns are included in the synthetic training data. On the contrary, most of data-driven baselines can capture the pattern of VA 2017-2018 season because they have seen similar patterns in the historical training data.

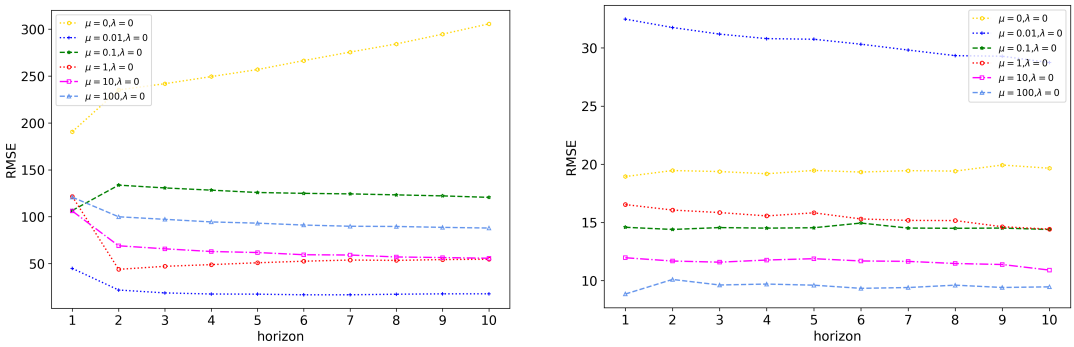


(a) NJ, 2017-2018 (b) VA, 2017-2018

Figure 3.9: Predicted curves for season 2017-2018. (a) NJ, 2017-2018; (b) VA, 2017-2018. This is state level forecasts with horizon = 3. The black curves are ground truth.

Sensitivity analysis. In this section, we conduct sensitivity analysis on two regularizer coefficients μ and λ in equation equation 3.6, which control the weights of the spatial constraint ϕ and non-negative constraint δ in the loss function. $\mu = 0$ means no spatial constraint and $\lambda = 0$ means no non-negative constraint. We train TDEFSI by setting $a = 52, b = 5$ with various μ, λ values. We then use the trained models to make forecasts for Season 2017-2018 of VA and NJ. The performance is evaluated using RMSE.

Spatial consistency: The experiments are conducted using $\lambda = 0$ and $\mu = \{0, 0.001, 0.01, 0.1, 1, 10, 100\}$. We evaluate the spatial consistency by computing RMSE of the predicted state level ILI incidence and the summation of the predicted county level ILI incidence, i.e., $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - \sum_{C\in\mathcal{D}} \hat{y}_i^C)^2}$. Figure 3.10 shows the spatial consistency error



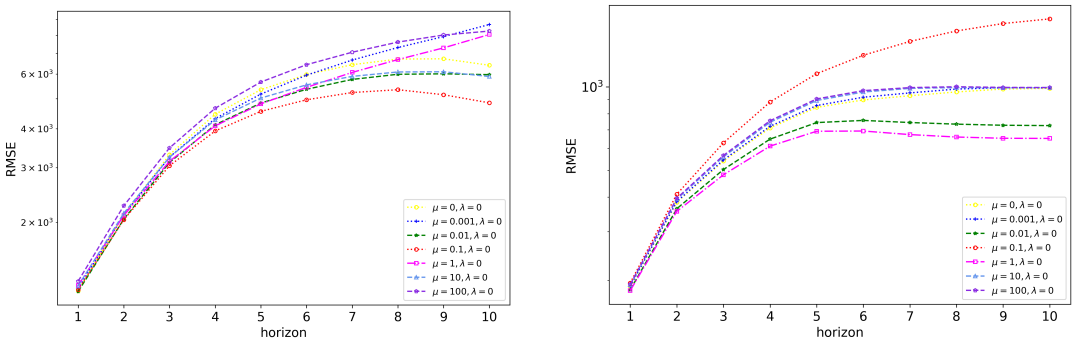
(a) VA, 2017-2018 (b) NJ, 2017-2018

Figure 3.10: Spatial consistency error with different μ values. Spatial consistency error (computed as $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - \sum_{C\in\mathcal{D}} \hat{y}_i^C)^2}$) on (a) VA, 2017-2018; (b) NJ, 2017-2018. The coefficient of the spatial consistency regularizer is set to $\mu = \{0, 0.001, 0.01, 0.1, 1, 10, 100\}$. The results show that the spatial consistency error does not vary much with horizon, but significantly depends on μ. The optimal μ differs between states.

measured by RMSE on (a) VA, 2017-2018 and (b) NJ, 2017-2018. The results show that the spatial consistency error does not vary much with horizon, but significantly depends on μ. The possible reason is that ,in TDEFSI model, the input is only state level data, so the LSTM layers learn the temporal pattern on state level time sequence which closely relates to model performance with horizons. However, spatial information is not propagated along the cells during training, but only compounds in the last step of outputs, thus is not impacted by horizons. The optimal μ differs between states. The results indicate that TDEFSI enables the spatial consistency with a proper μ value. However, a better spatial consistency does not mean a better model forecasting performance. In practice, we need to keep balance between keeping good spatial consistency and maintaining good model performance.

To evaluate the significance of the spatial consistency constraint for model forecasting power, we compare the forecasting performance of models on real seasonal data with various μ using RMSE (shown in Figure 3.11). For VA, the best performance is the model with $\mu = 0.1$. For NJ, the best performance is the model with $\mu = 1$. Overall, the spatial consistency constraint with a proper coefficient, which may vary between different regions, helps improve the forecasting performance.

Non-negative consistency: The experiments are conducted using $\mu = 0$ and $\lambda = \{0, 0.001, 0.01, 0.1, 1, 10, 100\}$. Similar to the spatial consistency evaluation, we compare the performance of models with various λ using RMSE (shown in Figure 3.12). For VA, the best performance is the model with $\lambda = 1$, and the models with the non-negative consistency constraint ($\lambda \leq 1$) outperform the model without the constraint. For



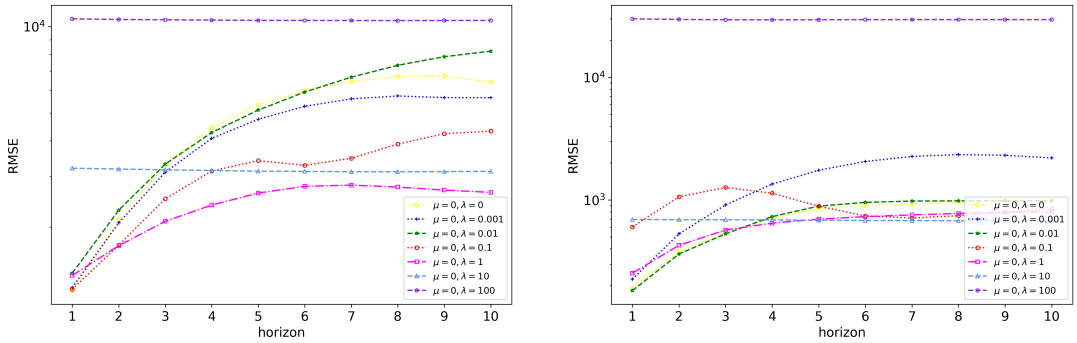
(a) VA, 2017-2018 (b) NJ, 2017-2018

Figure 3.11: TDEFSI performance with different μ values. spatial consistency constraints of different coefficients $\mu = \{0, 0.001, 0.01, 0.1, 1, 10, 100\}$. The performance is evaluated on (a) VA 2017-2018 season and (b) NJ 2017-2018 season. The results show that the coefficient μ has significant influence on the model forecasting performance especially with large horizons. The optimal value of μ should be chosen independently in different regions. A log y-scale is used in RMSE and MAPE.

NJ, the best performance is the model with $\lambda = 1$. For both VA and NJ, from the figures we observe that the models with λ equal or larger than 10 will have no predicting power (i.e., they are almost horizontal lines with high RMSE). The possible reason is that a strong penalty (large λ) may cause the weights of the hidden units to shrink towards zero. When $\mathbf{W}, \mathbf{U}$ in Equation 3.1 become zero the LSTM layer gives a constant output. This will make the network stop learning and output constant forecasts. Overall, the non-negative consistency constraint with a proper coefficient, which may vary between different regions, helps improve the forecasting performance.

Implications: The computational experiments show that these constraints can lead to a better domain consistency as well as improve the forecasting performance. By incorporating physical consistency, TDEFSI enables theory guided deep learning for epidemic forecasting. Spatial and non-negative consistency constraints also positively influence the overall performance. However, we note that no single parameter setting works across all scenarios thus context specific tuning is needed.

Uncertainty estimation. In the epidemic forecasting domain, probabilistic forecasting is important for capturing the uncertainty of the disease dynamics and to better support public health decision making. Probabilistic forecasting with deep learning models is challenging due to the lack of interpretability of such models. Most works on this are based on Bayesian Neural Networks. Gal and Ghahramani (2016) proved that using dropout technique was equivalent to Bayesian NN's and proposed MCDropout to estimate uncertainty in deep learning. The proposed method was computationally efficient. We implement MCDropout in TDEFSI and demonstrate



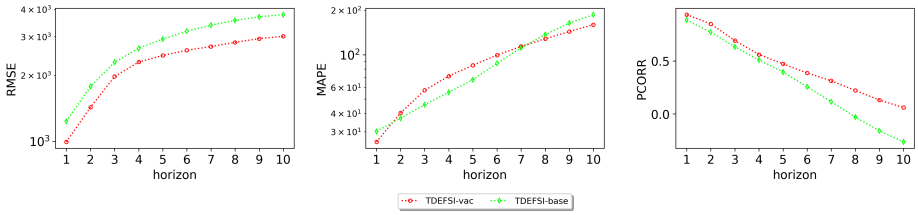
(a) VA, 2017-2018 (b) NJ, 2017-2018

Figure 3.12: TDEFSI performance with non-negative consistency constraints of different coefficients $\lambda = \{0, 0.001, 0.01, 0.1, 1, 10, 100\}$. The performance is evaluated on (a) VA 2017-2018 season; (b) NJ 2017-2018 season. The results show that the coefficient λ has significant influence on the model forecasting performance. The optimal value of λ should be chosen independently in different regions. A log y-scale is used in RMSE.

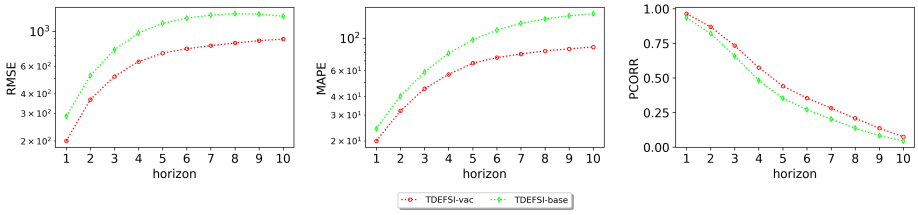
estimation of uncertainty with a case study of state level forecasting for NJ season 2016-2017. The model setting is the same as that described in the experiment setting section, and the MC number is 20. Figure 3.14 shows the curve of mean estimations with predictive intervals of $(mean \pm k * std)$ where $k = \{0.5, 1, 1.5, 2\}$. We can observe that all ground truths are within 2 standard deviations.

What-if forecasting. In this section, we are going to discuss a potential ability of TDEFSI framework – *what-if forecasting*. What-if forecasting means forecasts that can capture various what-if scenarios as epidemic is evolving. It is crucial when an unprecedented epidemic happens and consequently some public health interventions get involved during a flu season. The existing data-driven methods are hard to capture such patterns since they can only see historical observations. In our framework, high computing simulations of epidemic process allow us to incorporate multiple scenarios that are not really happening in current time but are of high possibility to happen in the future into the deep neural network model training. What-if scenarios can be any assumptions, such as emerging of the second virus strain during the season, temporary interventions like school closure or multiple vaccine schedules due to an intensive flu outbreak, etc. These scenarios can be involved in the epidemic simulations which will then generate new training data. The models trained on this data can capture the what-if scenarios leading to what-if forecasts, so helping the public health decision making and planning. What-if forecasting works as following using our framework:

- Make an assumption ζ



(a) VA, 2017-2018



(b) NJ, 2017-2018

Figure 3.13: State level forecasting performance comparison between TDEFSI models trained on the base-case simulated training dataset (TDEFSI-base) and the vaccine-case simulated training dataset (TDEFSI-vac). They test on VA, 2017-2018 with a horizon up to ten weeks ahead. TDEFSI-vac outperforms TDEFSI-base across three metrics. A log y-scale is used in RMSE and MAPE.

- Parameterize ζ, $\Theta(\zeta)$, so that it can be applied to the epidemic simulation using casual models;
- Generate synthetic epi-curves using the simulations with $\Theta(\zeta)$;
- Expand training data with new generated curves;
- Train TDEFSI model on new training data;
- Make forecasting.

Note that multiple assumptions can stack in one simulation. In the following, we will show a what-if scenario analysis with respect to vaccination-based intervention I_V.

In my experiments, we setup I_V using real world vaccination schedules so that the generated synthetic training dataset considers vaccination intervention context. Let's assume another scenario where there is no vaccination intervention applied, i.e., $I_V = \emptyset$. We generated two synthetic training datasets: (i) ***vaccine-case***: generated by simulations with I_V; and (ii) ***base-case***: generated by simulations that share the common settings of p_E, p_I, τ, N_I with vaccine-case except $I_V = \emptyset$. We train TDEFSI on the vaccine-case and base-case with the same settings described in Section 3.2.5, and denote the trained models as ***TDEFSI-vac*** and ***TDEFSI-base***, respectively. Note that here TDEFSI-vac is the same as TDEFSI in the previous experiments.

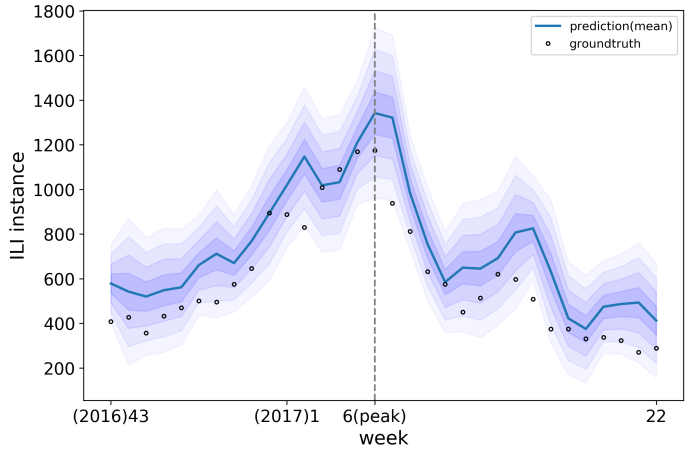


Figure 3.14: NJ state level mean predicted curve with predictive intervals of $(mean \pm k * std)$ where $k = \{0.5, 1, 1.5, 2\}$. The black circles are ground truths. We can observe that all ground truths are within 2 standard deviations.

Figure 3.13b and Figure 3.13a show the state level forecasting performance of NJ and VA on RMSE, MAPE, and PCORR using real-testing dataset. We observe that TDEFSI-vac significantly outperforms TDEFSI-base for all metrics on both states except that for the MAPE result of VA, TDEFSI-vac is compatible with TDEFSI-base. The results indicate that by using TDEFSI, vaccination-based interventions applied in the simulations can significantly improve the forecasting performance. The models learned from the vaccine-case dataset benefit from realistic settings thus are more generalizable to unseen surveillance data. The proposed framework is extensible for other realistic interventions, such as school closure or antivirals, to further improve the forecasting performance. These are future works to be explored.

3.3 CausalGNN

3.3.1 Motivation

As we have discussed, it is an area of active research for modeling and forecasting the spatial and temporal evolution of infectious disease. In the context of new emerging epidemics (e.g., COVID-19 pandemic), the forecasting problem has been particularly complicated (more discussions in Section 1.1.2). Applying existing methods to solve such forecasting problems presents several major challenges:

- The network-based compartmental models, compared to single-patched models, explicitly account for the connectivity among patches. Thus, it is more promising in capturing the relation between the model parameters and spatiotemporal data. However, calibrating such models, especially at the high geographical resolution, is challenging given the need to capture time-varying inter- and intra-regional effects which may not be accounted for in the static/dynamic travel matrix. For example, for the United States with 3000+ counties and W weeks of data, there are technically $3000 \times W+$ entries in the spatiotemporal transmissibility matrix to be calibrated, making traditional Bayesian techniques computationally intensive and susceptible to overfitting due to the limited training data size.

- Deep learning models especially GNN-based models usually require a sufficiently large quality dataset to train the large number of model parameters in order to avoid overfitting. Existing works (Wu et al., 2018; Deng et al., 2020) proposed models whose parameter sizes increased with graph node size making them fail when forecasting over a large number of locations. Reducing the complexity of such models is crucial for accurate forecasting.

- Prior works of physics, biology, and epidemiology (Karpatne et al., 2017; Wang et al., 2020b; Gao et al., 2021) have shown the evidence that incorporating domain knowledge into data-driven models can help improve spatiotemporal forecasting algorithms. However, existing deep learning models barely explicitly consider epidemiological context (Wu et al., 2018; Deng et al., 2020) for epidemic forecasting. Such models are prone to be overfitting leading to failures in long-term forecasting, especially when the data is noisy and sparse such as COVID-19 surveillance data at the US county level.

- Our model TDEFSI cannot applied directly to COVID-19 forecasting since there is no seasonal historical training data. Furthermore, generating realistic synthetic training data is particularly challenging due to the rapidly co-evolving dynamics of individual behavioral adaptations, government policies, and disease spread.

To address the above challenges, we propose **Causal**-based **G**raph **N**eural **N**etwork (CausalGNN). The CausalGNN attempts to capture the spatial and temporal dynamics via a well designed GNN module and uses a causal module to mutually provide and embed causal features to get epidemiological context. Causal constraints are also added to further improve model forecasting performance. The major contributions are summarized below:

- We propose a novel spatiotemporal learning framework that learns a latent space to combine the spatiotemporal and causal embeddings using graph-based non-linear transformations. Previous works (Wang et al., 2020b; Gao et al., 2021) have not considered the theory generated features in graph embedding. We present a jointly learning process for incorporating epidemiological context in GNN learning.

- We design an attention-based dynamic GNN module to embed spatial and temporal signals from disease dynamics. The parameter size in our design is agnostic to the number of locations thus leading to a stable forecasting performance on datasets of varying location numbers.

- We incorporate single-patched compartmental models into the framework to provide epidemiological context. Different from traditional compartmental models, in our framework, the patches are connected via a learned GNN. Calibration is done through GNN training, which is computationally efficient. The causal outputs are embedded as graph node features and used to regularize neural network forecasts for causal-based forecasting, leading to better forecasting performance.

- In order to allow for interaction between the causal and GNN modules, we design a causal encoder to encode causal features as node embedding to propagate over the graph and a causal decoder to infer mechanistic model parameters from latent space at each time step. We are the first to propose this iterative feedback mechanism that benefits from the learning in both modules.

- We evaluate the proposed framework for forecasting daily new confirmed case counts of COVID-19 at global, US state, and US county levels. Comparing with a broad range of baselines, our model performs the best in most cases. Through an ablation study, we demonstrate the effectiveness of GNN module, attention mechanism, and causal module in improving model performance.

3.3.2 Related Work

Epidemic forecasting methods are introduced in Section 1.1.3, COVID-19 forecasting methods and spatiotemporal forecasting methods are presented in Section 2.2, and

hybrid methods that combining mechanistic models and deep learning models are discussed in Section 3.1 and Section 3.2.2. We omit the details here for the sake of brevity. In summary, it is challenging to calibrate network-based mechanistic models, particularly at high geographical resolution, given the need to capture time-varying inter- and intra-regional effects. Using GNN-based models to capture spatiotemporal disease spread dynamics is promising as more sufficient good-quality surveillance data become available. However, existing methods barely explicitly consider epidemiological context leading to difficulties in explaining the learned model. Furthermore, their model complexity increases with graph node size making them fail when forecasting over a large number of locations.

CausalGNN method: Our method combines mechanistic models and deep learning models in a novel spatiotemporal learning framework. Different from TDEFSI (described in Section 3.2) which adopts a sequential learning process that involves the disease model calibration, the parameter space construction, and the deep learning model training in separate and sequential steps. Our method learns disease models and GNNs in an interactive way so that they can mutually impact each other. The jointly learning process allows deep learning models to explicitly incorporate epidemiological context while generate meaningful disease model parameters leading to better forecasting performance and better explainability of the learned model. To the best of our knowledge, the proposed CausalGNN is among the first significant hybrid methods that can achieve decent forecasting accuracy and can gain explainability of a learned GNN-based model while keeping a graph-size-agnostic parameter size (i.e., the parameter size of the model does not increase as the number of graph nodes increases).

3.3.3 Problem Formulation

We assume N regions in total, and each region is associated with a time series of reference data (for instance, confirmed cases of COVID-19). We define a dynamic graph on the N regions as $G(\mathcal{V}, \mathcal{E}, \mathcal{T})$, where $\mathcal{V}$ is the set of N nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of weighted edges, and $\mathcal{T}$ is the set of T time points at weekly or daily granularity depending on the available data. Let $\mathbf{C}_t = (\mathbf{c}_{i,t}) \in \mathbb{R}^{N \times C}$ represent the matrix of node features for N nodes where C is the feature number. An edge $e_{ij,t} \in \mathcal{E}$ connecting nodes v_i and v_j is weighted by an adjacency matrix $\mathbf{A}_t$ (such as geographical adjacency) where $a_{ij,t}$ denotes the impact of node v_j on node v_i at time t. The edge weights can differ at each time depending on the type of adjacency matrix. We denote the historical window size as K where $K \leq T$. The objective is to predict an epidemiological target at future time $T + h$ for N regions by looking back K time points where h denotes the horizon time.

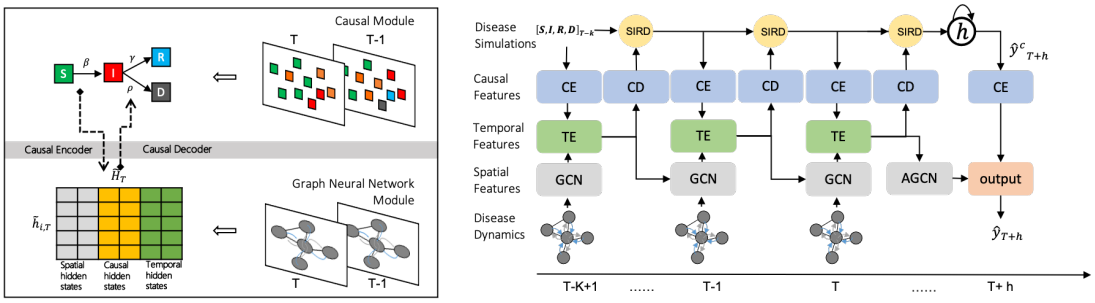


Figure 3.15: Framework of CausalGNN. The framework consists of a causal module and an attention-based dynamic GNN module. In the causal module, we run a single patch SIRD model for each region. In GNN module, the input data is a sequence of disease dynamic graphs that are fed into GCN layers in time order to learn dynamic spatial features. At each time step, we use a temporal encoder (TE) layer to learn temporal features from hidden states of a GCN layer. The two modules interact through a causal encoder (CE) and a causal decoder (CD) module. CE is to encode causal features from SIRD model for graph node embedding and CD is to decode the hidden states of a TE layer as SIRD model parameters at each time step.

3.3.4 Framework

The proposed framework (shown in Figure 3.15) consists of two major modules: 1) a causal module to provide epidemiological context for GNN learning via ordinary differential equations; 2) an attention-based dynamic GNN module (we called ADGNN) to capture the spatial and temporal disease dynamics via GCN layers and temporal encoder (TE) layers. The two modules interact with each other through causal encoder (CE) and decoder (CD) layers. The number of each type of the layer equals to the input length (K). They share common parameters along the time steps. It works as follows: we run a single patch SIRD model for each location. For each time of SIRD computation except the first one (initialized with ground truth data), the current disease model parameters are inferred from the current CD layer. After each SIRD computation, we save the current causal features for the next computation and feed them into the current TE layer. In ADGNN, the input data is embedded as a sequence of input embedding via an input layer. For each GCN layer except the first one, the input of the current GCN layer is the output of the previous TE layer which combines the hidden outputs from the GCN layer, the input layer, and the causal encoder layer at previous time. This iteration is repeated by K times. Then an attention-based GCN (AGCN) layer is employed to learn hidden impact between two hidden states from the last TE layer. The SIRD computation is run for h steps further with the most recent disease model parameters to get causal features at time $T+h$. The hidden features from the AGCN layer and the last CE layer are combined and fed into an output layer to get the final forecast. The pseudocode of the model training process is

Algorithm 4: CausalGNN training

Input: $G(\mathcal{V}, \mathcal{E}, \mathcal{T})$; Geographical adjacency matrix $\mathbf{A}$; Historical window size K; forecast horizon h.

Output: Model parameters Θ

$b \leftarrow$ a batch training sample
for *each instance* $\in b$ **do**
  $\mathbf{Q}_t \leftarrow \mathbf{Q}^g_{T-K+1}$ `// Initializing causal features`
  $\tilde{\mathbf{H}}_t \leftarrow \mathbf{H}^f_{T-K+1}$ `// Initializing temporal embedding`
  for t *in* $T-K+2 \dots T$ **do**
    $\mathbf{H}^c_t \leftarrow \text{CausalEncoder}(\mathbf{Q}_t)$ `// Causal encoding`
    $\mathbf{H}^f_t \leftarrow \text{Input}(\mathbf{C}_t)$ `// Input embedding`
    $\mathbf{H}_t \leftarrow \text{GCNLayer}(\tilde{\mathbf{H}}_{t-1}, \mathbf{A}_t)$ `// Spatial embedding`
    $\tilde{\mathbf{H}}_{t+1} \leftarrow \text{TempEncoder}(\mathbf{H}_t, \mathbf{H}^f_t, \mathbf{H}^c_t)$ `// Temporal embedding`
    $\mathbf{P}_t \leftarrow \text{CausalDecoder}(\tilde{\mathbf{H}}_{t-1})$ `// Causal decoding`
    $\hat{\mathbf{Y}}^c_{t+1}, \mathbf{Q}_{t+1} \leftarrow \text{SIRD}(\mathbf{Q}_t, \mathbf{P}_t)$ `// Causal simulating`
  for t *in* $T+2 \dots T+h$ **do**
    $\hat{\mathbf{Y}}^c_t, \mathbf{Q}_t \leftarrow \text{SIRD}(\mathbf{Q}_{t-1}, \mathbf{P}_T)$ `// Causal simulating for another` $h-1$ `steps`
  $\mathbf{H}^c \leftarrow \text{CausalEncoder}(\mathbf{Q}_{T+h})$ `// Causal encoding at time` $T+h$
  $\mathbf{H}^o \leftarrow \text{AGCN}(\tilde{\mathbf{H}}_T)$ `// Attention-based spatial embedding`
  $\hat{Y} \leftarrow \text{Output}(\mathbf{H}^o, \mathbf{H}^c)$ `// Predicting`
  $\hat{Y}^c \leftarrow [\hat{Y}^c_{T-K+2}, \dots, \hat{Y}^c_{T+h}]$
  $\Theta \leftarrow \text{BackProp}\big(LossFunc(\mathbf{Y}, \hat{\mathbf{Y}}, \mathbf{Y}^c, \hat{\mathbf{Y}}^c, \Theta)\big)$ ▷ `Adam opt`

described in Algorithm 4.

SIRD causal modeling. Inspired by the previous works on theory-guided data-driven algorithms developed across multiple domains (Karpatne et al., 2017; Wang et al., 2020b), we import epidemiological context by incorporating causal-based differential equations into deep learning framework. In this work, we focus on COVID-19 confirmed case forecasting. Based on the availability of daily confirmed, death, and recovered counts reported by the surveillance system, we choose a single patch compartmental susceptible(S)-infected(I)-recovered(R)-deceased(D) (SIRD) model (Loli Piccolomini and Zama, 2020) to simulate the COVID-19 spread in each location. Other models such as SIR can also work. We discuss this in the experiment section via an ablation study. Consider a population of N_i individuals in patch i, each of whom can be in one of the following states: S, I, R, D. Compartmental models operate under a homogeneous mixing assumption, i.e., every individual can directly infect any other individual. Let $\mathbf{q}_{i,t} = [S_i(t), I_i(t), R_i(t), D_i(t)]$ denote the *causal feature* vector where the element represents the cumulative number of individuals in each of the states, at

time t, and $\sum \mathbf{q}_{i,t} = N_i, \forall t$. Similarly, $\Delta\mathbf{q}_{i,t} = [\Delta S_i(t), \Delta I_i(t), \Delta R_i(t), \Delta D_i(t)]$ denotes the newly added number of individuals in each state. The dynamics of epidemic spread in such a patch model are described by the following equations:

$$\begin{aligned}
\Delta S_i(t+1) &= -\beta_i(t) S_i(t) \frac{I_i(t)}{N_i} \\
\Delta I_i(t+1) &= \beta_i(t) S_i(t) \frac{I_i(t)}{N_i} - \gamma_i(t) I_i(t) - \rho_i(t) I_i(t) \\
\Delta R_i(t+1) &= \gamma_i(t) I_i(t) \\
\Delta D_i(t+1) &= \rho_i(t) I_i(t)
\end{aligned} \tag{3.10}$$

where $\beta(t)$ denotes the transmissibility, $\gamma(t)$ and $\rho(t)$ denote the recovery rate and mortality rate, respectively, at time t. We also assume that individuals who become recovered do not get infected again. The *causal parameter* vector is denoted as $\mathbf{p}_{i,t} = [\beta_i(t), \gamma_i(t), \rho_i(t)]$. We denote $\mathbf{P}_t = (\mathbf{p}_{i,t}) \in \mathbb{R}^{N\times 3}$ as a matrix of causal parameters and $\mathbf{Q}_t = (\mathbf{q}_{i,t}) \in \mathbb{R}^{N\times 4}$ as a matrix of causal features for N regions at time t.

In our framework, $\mathbf{P}_t$ is inferred by a neural network and $\mathbf{Q}_{t+1}$ is updated as $\mathbf{Q}_t + \Delta\mathbf{Q}_{t+1}$ where the initial values $\mathbf{Q}_{T-K+1}$ are given as the input. The SIRD equations are run iteratively for K steps using inferred parameters $\mathbf{P}_{T-K+1}, \ldots, \mathbf{P}_T$ and are run for another $h-1$ steps using $\mathbf{P}_T$. This generates a series of $\Delta\mathbf{Q}_{T-K+2}, \ldots, \Delta\mathbf{Q}_{T+h}$ which will be fed into the GNN module in time order and then be used to regularize model forecasts in loss function.

GCN-based dynamic graph encoding (GCN). We leverage GCN (Kipf and Welling, 2017) to generate node embedding based on local network neighborhoods through message passing. The neighborhoods are defined using an adjacency matrix. A traditional GCN model consists of multiple layers for a single graph convolution. In our problem, the node features and adjacency matrix vary across time, hence we implemented a dynamic GCN (Deng et al., 2019a) that a GCN layer corresponds to a time step to learn spatial and temporal features. The number of GCN layers is the number of time points in the input sequence and they share a common parameter set. This dynamic GCN architecture allows our model to recurrently propagate forward the spatial and temporal features with a small parameter size.

The dynamic graph G is represented as a sequence of static graphs $[G_{T-K+1}, \cdots, G_T]$ with adjacency matrices $[\mathbf{A}_{T-K+1}, \cdots, \mathbf{A}_T]$ where K is the historical window size. We use a geographical adjacency matrix for all time points. At time t, let $\mathbf{H}_t^f \in \mathbb{R}^{N\times F_f^{(t)}}$ represent the matrix of hidden states of disease dynamics for N nodes by the input layer:

$$\mathbf{H}_t^f = \sigma\big(\mathbf{C}_t \mathbf{W}_f^{(t)} + \mathbf{b}_f^{(t)}\big) \in \mathbb{R}^{N\times F_f^{(t)}}, \tag{3.11}$$

where $\mathbf{W}_f^{(t)} \in \mathbb{R}^{C\times F_f^{(t)}}$, $\mathbf{b}_f^{(t)} \in \mathbb{R}^{F_f^{(t)}}$ are model parameters and σ is sigmoid activation

function. $F_f^{(t)}$ is the hidden dimension. And let $\mathbf{H}_t \in \mathbb{R}^{N \times F^{(t)}}$ denotes the matrix of hidden states from the GCN layer. A GCN layer at time $t+1$ maps from $\mathbf{H}_t$ to $\mathbf{H}_{t+1}$ as:

$$\mathbf{H}_{t+1} = g\big(\hat{\mathbf{A}}_t \tilde{\mathbf{H}}_t \mathbf{W}^{(t)} + \mathbf{b}^{(t)}\big) \in \mathbb{R}^{N \times F^{(t+1)}}, \tag{3.12}$$

where $\mathbf{W}^{(t)} \in \mathbb{R}^{F^{(t)} \times F^{(t+1)}}$, $\mathbf{b}^{(t)} \in \mathbb{R}^{F^{(t+1)}}$ are model parameters. $\hat{\mathbf{A}}_t = \tilde{\mathbf{D}}_t^{-\frac{1}{2}} \tilde{\mathbf{A}}_t \tilde{\mathbf{D}}_t^{-\frac{1}{2}}$ is the normalized symmetric adjacency matrix, where $\tilde{\mathbf{A}}_t = \mathbf{A}_t + \mathbf{I}_N$ and $\mathbf{I}_N$ is an identity matrix. $\tilde{\mathbf{D}}_t^{-\frac{1}{2}}$ is the degree matrix computed as $\mathbf{D}_{ii,t} = \sum_j \tilde{\mathbf{A}}_{ij,t}$. $\tilde{\mathbf{H}}_t \in \mathbb{R}^{N \times F^{(t)}}$ is the output from the previous temporal encoder (TE) layer. g is the activation function adopted as rectified linear units (ReLU) (Nair and Hinton, 2010).

Temporal encoder (TE). To consider temporal features in the graph, at each time step, we employ a temporal encoder (TE) layer to re-encode the node hidden representatives including hidden states from the current disease dynamics, the current causal features, and the current GCN layer. In Equation 3.12 the output $\tilde{\mathbf{H}}_t$ from the previous TE layer at time t is computed as:

$$\begin{aligned}
\dot{\mathbf{H}}_t^f &= \mathbf{H}_t^f \mathbf{W}_f^{(t)} + \mathbf{b}_f^{(t)}, \\
\dot{\mathbf{H}}_t^c &= \mathbf{H}_t^c \mathbf{W}_c^{(t)} + \mathbf{b}_c^{(t)}, \\
\dot{\mathbf{H}}_t &= \mathbf{H}_t \mathbf{W}_g^{(t)} + \mathbf{b}_g^{(t)}, \\
\tilde{\mathbf{H}}_t &= \tanh\left(\left[\dot{\mathbf{H}}_t^f \| \dot{\mathbf{H}}_t^c \| \dot{\mathbf{H}}_t\right]\right) \in \mathbb{R}^{N \times F^{(t)}},
\end{aligned} \tag{3.13}$$

where $\mathbf{H}_t^f, \mathbf{H}_t^c, \mathbf{H}_t$ are the hidden embedding from the input layer, the causal encoder layer (described below), and the GCN layer, respectively. $\mathbf{W}_f^{(t)} \in \mathbb{R}^{F_f^{(t)} \times a}$, $\mathbf{W}_c^{(t)} \in \mathbb{R}^{F_c^{(t)} \times b}$, $\mathbf{W}_g^{(t)} \in \mathbb{R}^{F_g^{(t)} \times c}$, $\mathbf{b}_f^{(t)} \in \mathbb{R}^a$, $\mathbf{b}_c^{(t)} \in \mathbb{R}^b$, and $\mathbf{b}_g^{(t)} \in \mathbb{R}^c$ are model parameters, and $a + b + c = F^{(t)}$. $\|$ represents concatenate operation. The TE module can be replaced by existing RNN modules such as RNN, GRU, or LSTM. We use the simple one to keep the parameter size small.

Causal-based encoder (CE) and decoder (CD). A causal encoder (CE) is designed to encode causal features as node embedding at time t. It works as:

$$\mathbf{H}_t^c = \tanh\left(\mathbf{Q}_t \mathbf{W}_e^{(t)} + \mathbf{b}_e^{(t)}\right) \in \mathbb{R}^{N \times F_c^{(t)}}, \tag{3.14}$$

where $\mathbf{W}_e^{(t)} \in \mathbb{R}^{4 \times F_c^{(t)}}$ and $\mathbf{b}_e^{(t)} \in \mathbb{R}^{F_c^{(t)}}$ are model parameters and σ is tanh function.

In causal module, the disease model parameters $\mathbf{P}_t$ for N nodes are inferred dynamically from GNN module via a causal decoder (CD).

$$\mathbf{P}_t = \sigma\big(\tilde{\mathbf{H}}_t \mathbf{W}_d^{(t)} + \mathbf{b}_d^{(t)}\big) \in \mathbb{R}^{N \times 3}, \tag{3.15}$$

where $\tilde{\mathbf{H}}_t$ is the hidden representatives from TE module. $\mathbf{W}_d^{(t)} \in \mathbb{R}^{F^{(t)} \times 3}$ and $\mathbf{b}_d^{(t)} \in \mathbb{R}^3$

are model parameters and σ is the sigmoid activation function.

Attention-based GCN (AGCN). In GNN module, we apply a GCN layer at each time step to learn spatial features of the dynamic graph using an adjacency matrix. However, the disease dynamics change and co-evolve at each time step thus the static or historical dynamic adjacency matrix cannot reveal the true connectivity. We want the model to learn an adaptive relationship between two nodes for forecasting at future time points. We define an attention matrix $\dot{\mathbf{A}} = (a_{ij}) \in \mathbb{R}^{N \times N}$ from the hidden states $\tilde{\mathbf{H}}_T \in \mathbb{R}^{N \times F^{(T)}}$ of the last TE layer to weight the graph edges at the last GCN layer and we call it the attention-based GCN (AGCN) layer. a_{ij} denotes the impact of node j on node i, computed as:

$$a_{ij} = \mathbf{v}^T g(\mathbf{W}_s \mathbf{h}_{i,T} + \mathbf{W}_t \mathbf{h}_{j,T} + \mathbf{b}_a) + b_a, \tag{3.16}$$

where g is ReLU that is applied element-wise; $\mathbf{W}_s, \mathbf{W}_t \in \mathbb{R}^{F_a \times F^{(T)}}$, $\mathbf{v}, \mathbf{b}_a \in \mathbb{R}^{F_a}$, and $b_a \in \mathbb{R}$ are model parameters. We use softmax function to normalize each row in $\dot{\mathbf{A}}$. Note that $\dot{\mathbf{A}}$ is an asymmetric matrix meaning that the impact of region i on region j is different than vice versa.

By using $\dot{\mathbf{A}}$, a GCN layer maps from $\tilde{\mathbf{H}}_T$ to $\mathbf{H}^o$ as:

$$\mathbf{H}^o = g\big(\dot{\mathbf{A}} \tilde{\mathbf{H}}_T \mathbf{W}_o + \mathbf{b}_o\big) \in \mathbb{R}^{N \times F_o}, \tag{3.17}$$

$\mathbf{W}_o \in \mathbb{R}^{F^{(T)} \times F_o}$, $\mathbf{b}_o \in \mathbb{R}^{F_o}$ are model parameters. g is ReLU function. The output of the AGCN layer will be fed into the output layer.

Output layer. As described above, the causal parameters $\mathbf{P}_T$ are used to run the SIRD model $h-1$ steps further to generate causal forecasts and $\mathbf{Q}_{T+h}$ will then be fed into a CE layer to generate $\mathbf{H}^c \in \mathbb{R}^{N \times F_c}$. We concatenate $\mathbf{H}^c \in \mathbb{R}^{N \times F_c}$ and the output of AGCN layer $\mathbf{H}^o$ and feed them to an output layer for final forecast:

$$\hat{\mathbf{Y}} = \phi\Big(\Big[\mathbf{H}^o \parallel \mathbf{H}^c\Big] \mathbf{W}_o + b_o\Big) \in \mathbb{R}^{N \times 3}, \tag{3.18}$$

where $\mathbf{W}_o \in \mathbb{R}^{(F_c + F_o) \times 3}$, $b_o \in \mathbb{R}^3$ are model parameters, ϕ is an identity function, and $\hat{\mathbf{Y}} = [\Delta\hat{\mathbf{I}}(T+h), \Delta\hat{\mathbf{R}}(T+h), \Delta\hat{\mathbf{D}}(T+h)]$ denotes the predicted causal vectors at time $T+h$ for N regions.

Optimization. We consider forecasting loss of causal module and GNN module in the loss function and then optimize a ℓ_1-norm loss via gradient descent:

$$\mathcal{L}(\Theta) = \|\mathbf{Y} - \hat{\mathbf{Y}}\| + \sum_{t=T-K+2}^{T+h} \|\mathbf{Y}_t^c - \hat{\mathbf{Y}}_t^c\|, \tag{3.19}$$

Table 3.5: Dataset statistics: min, max, mean, and standard deviation (std) of patient counts; dataset size means number of locations multiplied by # of days.

Data set	Size	Min	Max	Mean	std
Globe	93×355	0	823225	3988	15381
US-State	52×355	0	62168	1670	3192
US-County	1351×355	0	34497	59	238

where $\hat{\mathbf{Y}}_t^c = [\Delta\mathbf{I}(t), \Delta\mathbf{R}(t), \Delta\mathbf{D}(t)]$ denotes the causal vector from SIRD simulations at time t for N regions and $\mathbf{Y}, \mathbf{Y}^c$ represents the corresponding ground truth values.

Predicting. In the framework, given an epidemiological target, we have two predictive vectors from causal module and GNN module respectively. We use the forecasts from the GNN module as our final forecasts as it embeds hidden information from both modules via the output layer.

Model complexity. The number of parameters of the proposed model is $O(C \times F^{(t)} + F^{(t)} \times F^{(t)})$. It is agnostic to the number of locations in the dataset. In my setting, C and $F^{(t)}$ are limited to small numbers. Thus, the proposed model can capture spatiotemporal patterns and causal features of disease transmissions in an elegant and efficient way. We will provide more detailed analysis in the experiment section.

3.3.5 Experiments

In this section, we briefly describe the data preparation, the evaluation metrics, and baselines. We evaluate the proposed model on COVID-19 daily new confirmed case count forecasting at global, US state, and US county levels and compare the model with the state of the art. We also analyze the results and the methodology.

Datasets. We use three kinds of datasets for our experiments: disease dynamics datasets, geographical adjacency datasets, and population datasets. **Disease dynamics datasets**: We collected COVID-19 surveillance data for experiments via the JHU COVID-19 surveillance dashboard[2]. It contains daily confirmed, death, and recovered counts at global, US state and county levels, as well as locations' latitude and longitude, from May 3, 2020, to April 23, 2021. We compute daily new added counts based on the collected data. The US state dataset also includes hospitalization count. We select countries with population size of more than 8.7 million and US counties with more than 3000 confirmed cases by March 20, 2021 to ensure the data source accuracy. Finally, we include 93 countries, 52 states, and 1351 counties. Their statistics are shown in Table 3.5. **Geographical adjacency datasets**: Country

[2]Source:`https://github.com/CSSEGISandData/COVID-19`

Table 3.6: MAE and MAPE performance of different methods on the three datasets with leadtime= 7, 14, 21, 28. Mean and 95% confidence interval of 5 runs are shown. Bold face indicates the best result of each column and underlined the second-best.

	Globe				US-State				US-County			
MAE(↓)	7	14	21	28	7	14	21	28	7	14	21	28
SIR	4777±819	4880±615	5090±1125	5182±282	677±31	738±58	831±51	854±60	38.8±3.3	44.2±2.7	51.3±2.8	58.5±2.0
PatchSEIR	4419±500	4562±601	4737±349	5167±298	633±78	687±78	757±37	876±77	73.5±5.4	84.4±6.7	100.8±14.6	110.6±6.4
AR	2298±10	3024±7	3619±17	4258±246	377±1	580±3	683±11	758±28	24.8±0.1	33.6±0.3	34.2±0.3	35.6±0.4
ARMA	2254±13	2987±14	3596±23	4239±60	379±3	583±3	686±9	750±17	23.8±0.1	27.0±0.1	33.0±3.8	35.9±0.7
RNN	2395±44	2871±28	3328±27	3596±178	369±13	525±38	660±191	745±181	21.5±1.0	25.0±2.4	35.9±5.0	36.1±3.7
GRU	2189±48	2916±30	3379±37	3620±130	385±27	504±80	660±100	833±174	31.6±4.1	30.1±11.5	35.1±8.8	38.1±11.1
LSTM	1911±16	2585±11	3050±21	3598±140	344±9	421±24	552±161	748±134	23.4±1.0	**23.7±0.3**	33.0±4.7	37.6±1.3
DCRNN	2287±189	2892±137	3369±71	3804±177	393±20	470±26	657±172	702±165	22.4±1.3	25.3±1.0	31.2±6.7	36.3±6.4
CNNRNN-Res	4143±649	4526±572	4467±437	4479±390	642±31	658±41	732±94	856±148	29.8±1.3	31.0±1.1	33.4±1.1	36.0±2.8
LSTNet	2693±91	3535±125	3909±209	4285±155	443±19	597±34	744±73	815±53	24.5±0.7	28.0±1.7	31.5±1.0	33.2±1.6
STGCN	4750±796	4325±357	4669±202	4494±162	580±19	630±19	699±95	793±60	23.7±1.1	26.5±2.7	30.9±2.7	32.7±5.4
Cola-GNN	2314±231	3012±682	3225±263	3755±175	384±30	497±19	613±124	810±343	22.5±1.4	37.7±19.1	34.5±7.5	37.5±9.3
STAN	**1851±172**	2628±144	3163±138	3574±142	350±16	428±27	512±80	**622±122**	22.2±0.7	25.3±1.9	28.5±1.7	31.2±4.6
CausalGNN	1905±192	**2509±144**	**3045±211**	**3313±58**	**340±13**	**419±16**	**500±68**	645±51	**21.4±0.4**	24.3±0.1	**27.5±0.7**	**29.4±0.7**
MAPE(↓)	7	14	21	28	7	14	21	28	7	14	21	28
SIR	577±46	335±61	285±10	298±19	141±27	147±34	139±26	146±31	233.7±9.3	260.9±11.5	217.8±9.3	234.0±2.3
PatchSEIR	342±26	268±17	225±13	228±21	152±26	143±24	153±31	180±23	472.9±15.0	479.4±18.2	546.3±10.0	642.1±31.5
AR	**108±0.3**	109±0.7	110±0.7	130±13.1	93±1.0	114±1.7	150±8.4	178±18.6	79.7±0.1	79.4±0.9	81.8±0.9	84.7±0.9
ARMA	109±0.3	110±2.8	110±1.3	127±9.7	91±2.4	111±2.0	146±12.0	175±25.5	75.6±0.1	89.4±0.6	92.2±13.4	86.8±0.7
RNN	131±13	98±10	108±13	112±3	86±9	130±34	158±61	192±85	62.9±13.4	87.0±6.3	98.2±9.8	137.2±31.9
GRU	124±10	115±13	99±13	113±11	95±14	98±40	118±28	210±97	64.3±4.6	79.9±12.0	118.6±20.0	134.3±42.0
LSTM	126±4	104±1	95±4	119±16	84±3	89±7	122±48	182±54	62.5±6.0	**62.1±3.7**	108.2±27.2	134.1±33.3
DCRNN	135±14	120±10	122±7	132±10	95±3	107±3	132±37	137±38	63.5±6.1	71.0±4.6	85.4±6.8	96.1±25.8
CNNRNN-Res	230±41	218±20	206±48	204±20	108±11	136±26	150±34	167±11	90.8±12.2	91.9±8.4	97.3±13.8	103.7±19.2
LSTNet	131±4	114±17	129±14	147±19	86±3	110±9	137±11	171±11	72.7±3.0	81.3±8.3	86.5±4.8	107.9±9.7
STGCN	210±13	173±9	168±24	163±16	129±11	146±21	155±37	180±31	62.9±4.4	71.4±6.3	83.8±10.1	85.3±12.1
Cola-GNN	125±31	119±53	96±10	100±11	95±16	119±27	122±14	218±144	**56.5±5.7**	101.1±10.4	110.2±12.7	123.4±22.5
STAN	126±5	**96±7**	92±10	109±11	86±1	96±1	108±1	**109±3**	75.4±1.9	82.8±0.6	95.4±1.0	104.9±2.4
CausalGNN	123±4	99±6	**91±9**	**98±17**	**81±4**	**88±4**	**106±11**	146±10	62.1±3.7	65.6±1.0	**72.3±4.4**	**73.1±2.8**

adjacency and US state adjacency matrices are manually collected and cleaned. US county adjacency is downloaded from the US Census Bureau[3]. **Population datasets**: The country population (2020) data is collected from the worldometers website [4]. The US state and county population (2019) datasets are downloaded from the US Census Bureau[5].

Metrics. The metrics used to evaluate the forecasting performance are: *mean absolute error (MAE)* (see Equation 3.20) and *mean absolute percentage error (MAPE)* (see Equation 3.9). Assuming we have n testing data points and $n = N \times m$ means N locations by m days. We denote the true value and forecast for the ith testing data point to be z_i and $\hat{z}_i$. We do not distinguish locations in calculating MAE and MAPE.

- The **Mean absolute error (MAE)** is a measure of absolute difference between two variables:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|z_i - \hat{z}_i| \tag{3.20}$$

  MAE ranges in $[0, +\infty]$ and smaller values are better.

[3]Source:`https://www2.census.gov/geo/docs/reference/county_adjacency.txt`

[4]Source:`https://www.worldometers.info/world-population/population-by-country/`

[5]Source:`https://www.census.gov/data/datasets/time-series/demo/popest/2010s-counties-total.html`

Baselines. To serve as baselines, we implemented a broad range of classic and the state-of-the-art forecasting methods.

- **SIR** is a single patch SIR compartmental model. **PatchSEIR** (Venkatramanan et al., 2017) is a network-based SEIR compartmental model for influenza forecasting. We use a gravity model (Barbosa et al., 2018) to generate a network flow of mobility.

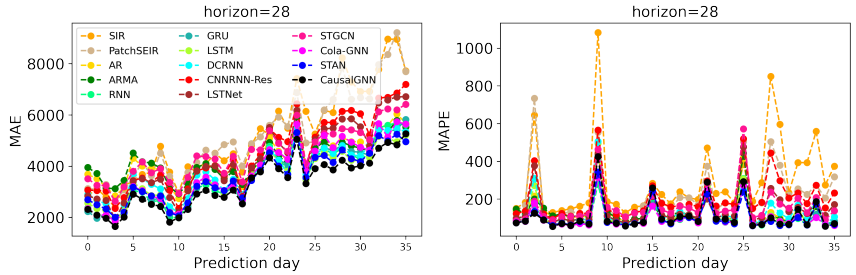
- **AR** uses observations from previous time steps as input to a regression equation to predict the value at the next time step. We adopt an AR model of order 28. **ARMA** (Contreras et al., 2003) is used to describe weakly stationary stochastic time series in terms of two polynomials for the autoregression (AR) and the moving average (MA). We set AR order to 28 and MA order to 2.

- **RNN** (Werbos, 1990) is a one layer RNN model with hidden state dimension as 32. **GRU** (Cho et al., 2014) is a one layer GRU model with hidden state dimension as 32. **LSTM** (Hochreiter and Schmidhuber, 1997) is a one layer LSTM model with hidden state dimension as 32.

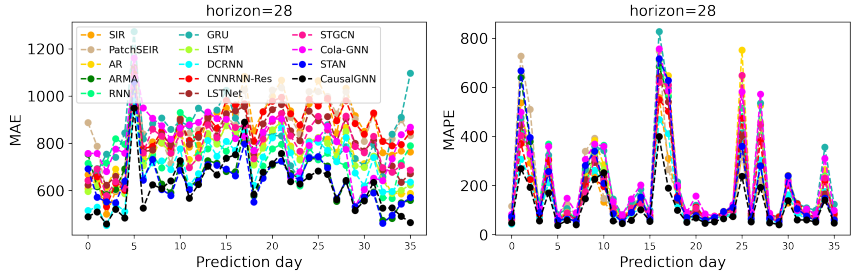
- **DCRNN** (Li et al., 2017) combines GCNs with RNNs in an encoder-decoder manner. **CNNRNN-Res** (Wu et al., 2018) combines CNNs, RNNs, and residual links in one framework. It employs RNNs to encode temporal information and CNNs to fuse information from data of different locations. **LSTNet** (Lai et al., 2018) uses CNNs and RNNs to extract short-term local dependency patterns among variables and to discover long-term patterns for time series trends.

- **STGCN** (Yu et al., 2017) integrates graph convolution and gated temporal convolution through spatiotemporal convolutional blocks for traffic forecasting. **Cola-GNN** (Deng et al., 2020) uses location-aware attention graph neural networks to combine graph structures and time series features in a dynamic propagation process. **STAN** (Gao et al., 2021) integrates disease dynamics theory into GNN training for COVID-19 forecasting. Partial data such as ICU visits are not available for our selected locations thus has been omitted from the model implementation.

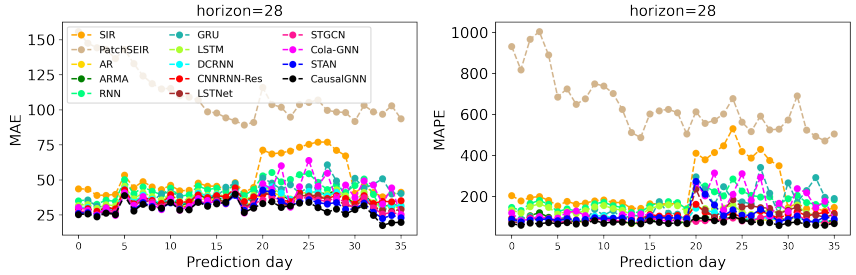
Settings and implementation details. For all models, the historical window $H = 28$. Unless otherwise specified, all baselines have parameters set in accordance with the original paper. In our model, the hidden dimensions of the input layer ($F_f^{(t)}$), GCN layers ($F^{(t)}$), and causal encoder ($F_c^{(t)} = F_c$) and decoder ($F^{(t)}$) layers are 32. AGCN layer hidden dimension (F_a) is set as 16 ($\frac{F^{(t)}}{2}$) and the output layer hidden dimension (F_o) is equal to K. We set the hidden dimension of linear transformation in equation 4 as $a = 12, b = 10, c = 10$. All the parameters are initialized with Glorot initialization (Glorot and Bengio, 2010). We set batch size as 32, epoch number as 1000. We use mean absolute error (MAE) loss and Adam (Kingma and Ba, 2014) optimizer with default settings, and early stopping with patience of 100 epochs for all model training. The collected disease dynamics datasets are split into training



(a) Globe



(b) US-State



(c) US-County

Figure 3.16: Performance of MAE and MAPE computed across all locations at various forecast days. It is observable that the model performance varies across the days, but our model performs the best in most of the days.

datasets (from May 3, 2020, to March 20, 2021) and testing datasets (from March 21, 2021, to April 23, 2021). For each data point in a testing dataset, we make 7, 14, 21, and 28 days ahead forecasting of the data point. All results are an average of 5 randomized trials. We show experiment results with their means and 95% confidence intervals. All programs are implemented using Python 3.7.4 and PyTorch 1.4.0 with CUDA 10.1 in a Simple Linux Utility for Resource Management (SLURM) system with K80, P100, V100, and RTX2080 NVIDIA GPU devices that serve in random.

Results and Analysis

In this section, we first show model performance on daily new confirmed case count forecasting. Then we present an ablation study to show the importance of each component in our framework. We also conduct sensitivity analysis on several hyperparameters and analyze model complexity of all baselines. Then we discuss the interpretability of the proposed model. At last, we present the comparison results with methods from COVID-19 Forecast Hub.

Forecasting performance. We evaluate our method and all baselines on three datasets in four different forecasting horizons (horizon = 7, 14, 21, 28). Table 3.6 shows the model performance in terms of MAE and MAPE. It is to be noted that the MAE performance of different datasets is not comparable while the MAPE performance is comparable.

We observe that the CausalGNN performs consistently better than the baselines across multiple scales and with increasingly horizon. A possible reason is that it considers both graph structure information and disease transmission dynamics. STAN with similar components also performs well in long-term forecasting. However, CausalGNN performs better than STAN in most cases because it not only adds a causal-based regularizer in the loss function but also jointly encodes causal features into GNN learning, which provides epidemiological context recurrently. Another possible reason is that CausalGNN model complexity is smaller than STAN which may avoid overfitting particularly when the training data is small in size and noisy in quality. Cola-GNN performs well on the Globe and US-State datasets but not on the US-County dataset. The possible reason is that its model size increases linearly with the squared number of locations (N^2) leading to overfitting to the US-County dataset of 1351 locations.

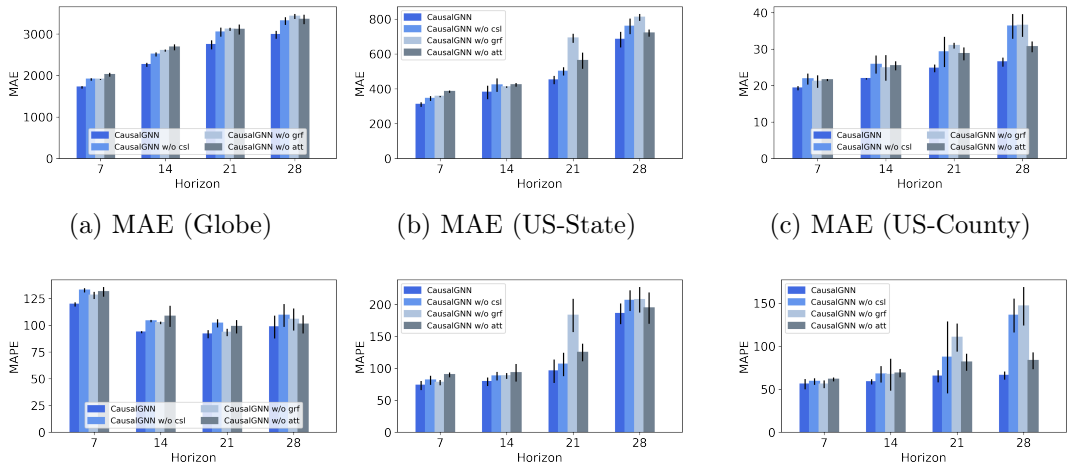
SIR and PatchSEIR perform worse than data-driven methods, especially for long-term forecasting. PatchSEIR performs worse than SIR at county level. As we mentioned in the beginning of this work, single patched models do not consider the spatial connectivity thus fail to capture spatial disease transmission dynamics. PatchSEIR leverages a gravity model-generated network but may not represent real world mobility activities. Further, calibrating is prone to overfitting on the US-County dataset due to the large number of counties (discussed in Section 3.3.1). In our framework, the patches are connected via a learned GNN and allows the spatial and temporal disease dynamics to exchange information in a latent space. The results demonstrate the practical value of our design.

Compared with GNN-based models like Cola-GNN, STAN, and CausalGNN, the vanilla RNN, GRU, LSTM models perform well in horizon=7,14. However, as the horizon increases their advantages have diminished. This indicates the importance of capturing spatial and temporal disease transmission patterns in the input data for long term forecasting. In most cases, the classic statistical methods (AR, ARMA) show a poorer performance than the classic RNNs (RNN, GRU, LSTM). This implies the

importance of modeling non-linear patterns for achieving good forecasting performance.

The proposed model can capture spatial and temporal patterns using a dynamic GNN architecture with a relatively small parameter size. This provides a consistent forecasting performance across datasets of varying number of locations when compared with existing spatiotemporal models.

Figure 3.16 shows the model performance of MAE and MAPE computed across all locations at various forecast day. We observe that the model performance varies across the days but our model performs the best in most of the days. We also observe that the MAE values at the Global level increase by days. The trend in MAE values coincides with the trend in the number of global daily new confirmed cases, which increases day by day from March 21, 2021, to April 23, 2021. The same happens to the MAE values at US-State and US-County levels. However, the MAPE results show a flat trend with interval spikes across days (variability in reporting across day of a week). The spikes of MAPE are caused by the noise in the testing datasets. These observations indicate that all models are implemented in a fair manner and perform stably across days.



(a) MAE (Globe) (b) MAE (US-State) (c) MAE (US-County)

(d) MAPE (Globe) (e) MAPE (US-State) (f) MAPE (US-County)

Figure 3.17: Ablation analysis on major components of the proposed model: CausalGNN w/o csl, CausalGNN w/o grf, and CausalGNN w/o att. The analysis is conducted at three datasets and evaluated by MAE and MAPE.

Ablation study. To explore the effect of the causal module and graph structure in our model, we conduct an ablation analysis on three datasets.

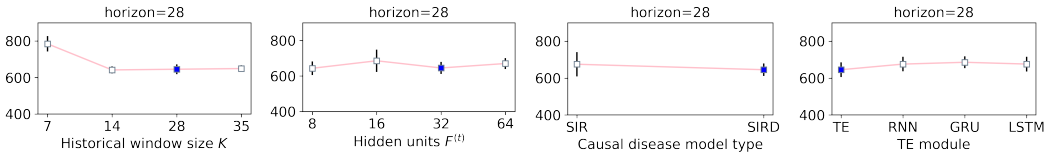
- **CausalGNN w/o csl**: Remove the SIRD causal encoder and decoder layers from the proposed model and remove the second term from the loss function in Equation 3.19. We call the removed components as the causal module (CSL).

- **CausalGNN w/o grf**: Remove the GCN layers and the AGCN layer from the model architecture. This means remove the geographical adjacency information and the attention mechanism. We call the removed parts as graph structure module (GRF).

- **CausalGNN w/o att**: Remove the AGCN layer from the proposed model architecture, which means remove the attention mechanism. We call the excluded component as the graph attention module (ATT).

We present the comparisons of forecasting performance in terms of MAE and MAPE for the above-described model configurations in Figure 3.17. Each comparison group (of the same metric, dataset and horizon) involves four models: CausalGNN, CausalGNN w/o csl, CausalGNN w/o grf, and CausalGNN w/o att. Within a group, CausalGNN serves as the baseline, a model with a larger MAE or MAPE value than the baseline indicates a more important role of the missing component in that model.

Major observations and discussion: CausalGNN always performs the best among the four models on different datasets and horizons. This implies that all three components play important roles in improving our model performance. Specifically, in long-term forecasting (horizon=21,28), CausalGNN w/o grf performs the worst on three datasets, followed by CausalGNN w/o csl with the second worst and by CausalGNN w/o att. This indicates that GRF plays the most important role in improving long-term forecasting performance. It complies with the fact that incorporating cross-spatial signals is crucial for a good epidemic forecasting model. Also, GRF's importance increases with increasing spatial resolution which is intuitive as the spatial interdependence is higher at state and county level. The results also show that adding the CSL to the framework can lead to a performance improvement. This demonstrates the effectiveness of the CSL in improving epidemic forecasting performance.



(a) Window size (b) Hidden dimension (c) Mechanistic model (d) TE module

Figure 3.18: Sensitivity analysis on (a) historical window size $K = 7, 14, 28, 35$, (b) Hidden dimension $F^{(t)} = 8, 16, 32, 64$, (c) Mechanistic model (SIR, SIRD), and (d) TE module (TE, RNN, GRU, LSTM). The blue markers represents the MAE performance of CausalGNN at the US state level from the main experiment.

Sensitivity analysis. In this section, we show sensitivity analysis on some of the hyperparameters of CausalGNN: historical window size K (Figure 3.18a), hidden

dimension $F^{(t)}$ (Figure 3.18b), causal disease model (Figure 3.18c), and TE module (Figure 3.18d). Except the varying hyperparameters, all the other settings are the same with parameter setting described in the setting paragraph. We only report MAE performance on US-State dataset with horizon=28. The other results show similar observations thus are omitted for the sake of brevity.

Major observations and discussion: (1) Figure 3.18a shows that the model performance gets improved when K increases from 7 to 14, however, there is no obvious improvement after $K = 14$. (2) Figure 3.18b shows that the performance changes as $F^{(t)}$ increases. We observe that there is no significant performance improvement by increasing $F^{(t)}$ value. We keep a value less than 64 in the experiment to reduce the model parameter size. (3) The results in Figure 3.18c show that there is no significant difference between the performance of using SIRD model and SIR model. We choose SIRD model since it complies with the fact that COVID-19 virus can cause deaths. For future use of our framework, we recommend using a model that is as realistic as possible to mitigate the forecasting error imported by an assumption bias. (4) Figure 3.18d shows that the model performance does not vary too much in terms of TE module type. We prefer a module of smaller parameter size.

Model complexity. The number of parameters of our model is agnostic to the number of locations N, as well as RNN, GRU, and LSTM models. The parameter sizes of AR, ARMA, and LSTNet increase linearly with the N while those of CNNRNN-Res and Cola-GNN increase linearly with N^2. The parameter sizes of SIR and PatchSEIR are linearly increasing with $N \times T_0$ where T_0 is the length of historical time series. We compare the model parameter size of all methods in Table 3.7. The results show that compared with the other graph-based neural network models, CausalGNN keeps a relatively small parameter size even when the number of locations increases. This demonstrates that our method can perform stably on different datasets.

Fairness analysis. Besides predicting the spread of an infectious disease, AI systems can be used for other important tasks, such as predicting the presence and severity of a medical condition or matching people to jobs. Any unfairness in such systems can have a far-reaching impact. Therefore, it is critical to work towards systems that are fair and inclusive for all. In this section, we perform fairness analysis on our model and evaluate its performance across regions with a broad range of demographic distributions and other variability. We show the MAPE performance over a map of the US in Figure 3.22a and a corresponding distribution over values in Figure 3.22b. Compared with Figure 3.22e which shows the urban-rural classification scheme for counties over the US map, we observe that the urban counties achieve better MAPE performance than the rural counties. One possible explanation is that the rural counties have low level confirmed cases (many have 0 counts) compared to the urban counties and MAPE is biased by those data points of very small ground truth values due to the nature of MAPE metric (please refer to the MAPE definition shown in the

Table 3.7: Model parameter size comparison on the US-State, Globe, and US-county datasets. κ denotes the real parameter size on US-State level. We show real parameter size for US-State level and relative values for US-County and Global level.

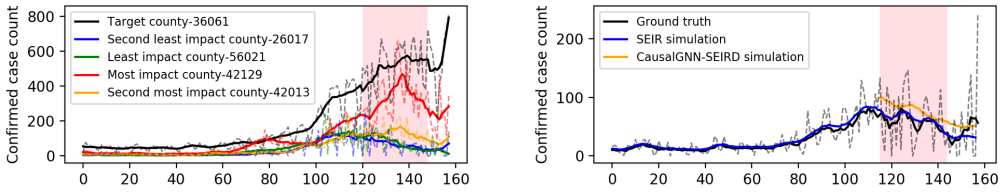
Methods	**US-State**	**Globe (κ)**	**US-County (κ)**
SIR	16.6K	1.79	2.60
PatchSEIR	16.6K	1.79	2.60
AR	1.5K	1.79	25.98
ARMA	2.9K	1.79	25.98
RNN	0.5K	1.00	1.00
GRU	1.4K	1.00	1.00
LSTM	1.9K	1.00	1.00
DCRNN	21	1.00	1.00
CNNRNNRes	9.7K	2.04	201.98
LSTNet	13.3K	1.61	20.48
STGCN	14.6K	1.01	1.35
ColaGNN	5.7K	2.05	323.51
STAN	8K	0.96	0.96
CausalGNN	1.4K	0.97	0.97

metrics section). To remove the bias imported by the metric, we computed Pearson correlation (PCORR) between predicted curves and the ground truth curves for all counties and show their performance distribution in Figure 3.22c and 3.22d. The magnitude of PCORR metric is independent of the ground truth values. We see that the model performs similarly across counties and there is no discernible pattern in the forecast distribution. This indicates that our model can perform fairly well in all counties.

Interpretability. The aim of the proposed framework is to provide not only correct inferences but also the mechanistic understanding of the learned deep learning model as well as the model forecasts. We show an example of the learned attentions of New York City, NY in figure 3.19a. We smoothed the curves of daily new confirmed cases by Savitzky–Golay filter (Savitzky and Golay, 1964) with window size 7 and polynomial order 1 to remove biases in daily reporting of cases. The counties with the highest and the second highest attention values are not geo-adjacent to the target county but show similar trend with the target curve within the input time duration, while the counties with the lowest and the second lowest values show an opposite trend (uptrend vs. downtrend). The results indicate that spatial attentions provide indicators for future event forecasts.

Figure 3.19b shows an example of the learned SIRD model of Hardin, KY. We present the simulated curves by a single patch SEIR model using post-calibrated parameters (shown by the blue line), and a single patch SIRD model using parameters inferred by CausalGNN (shown by the yellow line). The curves are smoothed for ease of viewing. The results show that CausalGNN can reveal mechanistic causal

process by producing meaningful causal parameters which can provide meaningful epidemiological context for GNN learning. By using the inferred causal parameters, we can run SIRD model independently to produce multiple forecasts such as death count. Furthermore, our model enables counterfactual forecasting by introducing different circumstance such as vaccine schedule to the simulations in causal space. The example we present here does not mean that our model can learn meaningful parameters in all places, but it is a good start of building explainable deep learning models by our method. More systematic and rigorous experimental analysis is needed in the future.



(a) Learned attentions (b) Learned SIRD

Figure 3.19: An example of (a) the learned attentions of New York City and (b) the learned SIRD model of Hardin, KY. Shaded area is the input. Solid curves are smoothed.

To illustrate how the causal module can help in improving the model performance, I compare the state level forecasts by CausalGNN and CausalGNN w/o csl in Figure 3.20. It shows the forecasts of confirmed cases of 2021-04-18 at the US state level. The black curve represents the ground truth while the orange curve represents the generated causal forecasts by the causal module in CausalGNN. Both solid lines and dots are smoothed values. The shaded area is the input window. The blue dots represent the forecasts by CausalGNN and the red crosses represent the forecasts by CausalGNN w/o csl. They are both 7 days ahead forecasting. We can observe that the causal module can generate meaningful curves and CausalGNN makes a better forecast than CausalGNN w/o csl for most states.

Comparison with state-of-the-art forecasts. To demonstrate the practical value of the proposed model, we add a dropout layer to the model and generate probabilistic forecasts for the US county level. The daily forecasts are aggregated to weekly values and then compared with state-of-the-art forecasts submitted to CDC COVID-19 Forecast Hub (marked by *). Among the 70 modeling teams present in the Hub only a handful of them provide county-level forecasts. In order to make a fair comparison, we only consider teams that have been providing consistent forecasts across most locations and targets since August 2020.

State-of-the-art forecasts: (1) ***COVIDhub-ensemble*** linearly combines the forecasts from teams with uniform weights produce probabilistic forecasts. (2) ***CU-select*** is a metapopulation county-level SEIR model. (3) ***JHU_IDD-CovidSP***

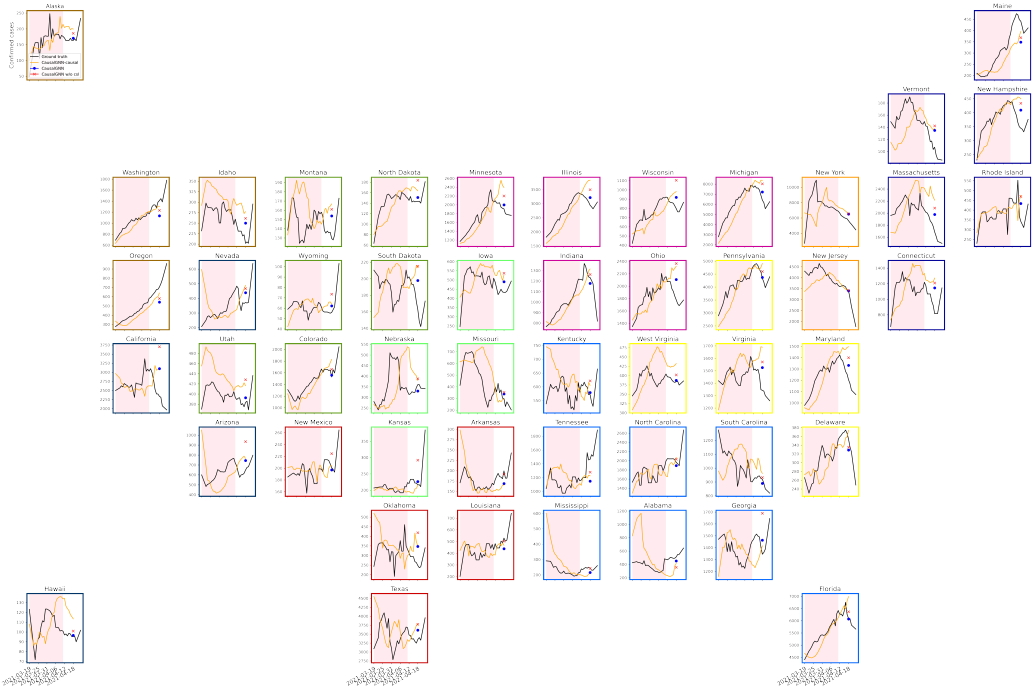


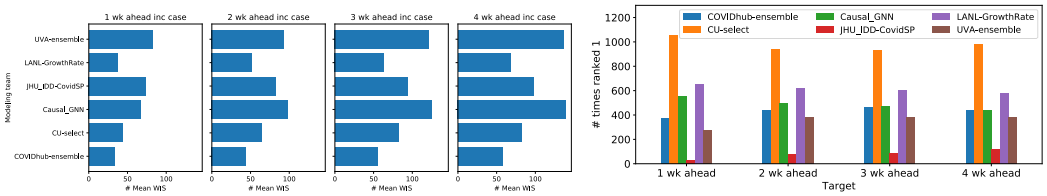
Figure 3.20: US state level forecasts of COVID-19 new confirmed cases at date 2021-04-18 by CausalGNN and CausalGNN w/o csl. The black curve represents the ground truth while the orange curve represents the generated causal forecasts by the causal module in CausalGNN. Both solid lines and dots are smoothed values. The shaded area is the input window. The blue dots represent the forecasts by CausalGNN and the red crosses represent the forecasts by CausalGNN w/o csl. They are both 7 days ahead forecasting.

is a county-level metapopulation model with commuting and stochastic SEIR disease dynamics with social-distancing indicators. (4) ***LANL-GrowthRate*** is SI model with a dynamic growth rate parameter assumed to follow a statistical model. (5) ***UVA-ensemble*** uses Bayesian Model Averaging (BMA) to combine forecasts from AR, ARIMA, LSTM, Kalman Filter, and metapopulation SEIR models.

We employ the **Interval Score (IS)** (Bracher et al., 2021) for evaluating the performance of the probabilistic forecast F.

$$IS_\alpha(F, y) = (u - l) + \frac{2}{\alpha}(l - y)\mathbb{1}(y < l) + \frac{2}{\alpha}(y - u)\mathbb{1}(y > u) \tag{3.21}$$

where $(1 - \alpha) \times 100\%$ is the prediction interval of F characterized by the upper bound u and the lower bound l that is likely to contain the forecast value y. $\mathbb{1}(\cdot)$ is the indicator function that outputs binary value. The IS is computed for a various prediction intervals and their weighed combination yields the **Weighted Interval**



(a) WIS (b) WIS rank

Figure 3.21: Probabilistic forecasting performance compared with state-of-the-art forecasts. (a) The mean WIS computed across all locations and all forecast weeks (in the testing dataset) for each of the modeling team. (b) The number of locations across the weeks that a model is ranked 1 by WIS.

Score (WIS):

$$WIS_{\alpha_{0:l}} = \frac{1}{L+1} \sum_{k=0}^{L} \frac{\alpha_l}{2} IS_{\alpha_l}(F, y) \tag{3.22}$$

L is the number of prediction intervals and smaller values are better (see Section 1.1.2 for the introduction of IS and WIS).

Figure 3.21 shows the probabilistic forecasting performance compared with state-of-the-art forecasts in CDC COVID-19 Forecast Hub. From Figure 3.21a, we see that the proposed CausalGNN outperforms UVA-ensemble and JHU_IDD-CovidSP for 1 week ahead forecasting and is comparable with UVA-ensemble for 2, 3, and 4 weeks ahead forecasting. COVIDhub-ensemble ensemble the forecasts of all thus is prone to perform the best. In Figure 3.21b we plot the number of locations across the weeks that a model is ranked top 1 by WIS. We observe that across targets CausalGNN is one of the top three performing models (there are six comparison methods). We'd like to point out that all COVIDhub teams conduct delicate inspection on their *real-time* submitted forecasts every week. They retrained their models when new data becomes available and involved expert feedback as part of the forecasting loop. However, CausalGNN made retrospective forecasting i.e., no model retraining when forecasting the testing data points without expert feedback. Thus, our model does not surpass these state-of-the-art results in WIS performance. However, the mediocre performance still indicates that our model matches the average performance of these state-of-the-art results. This demonstrates the practical value of our model very well.

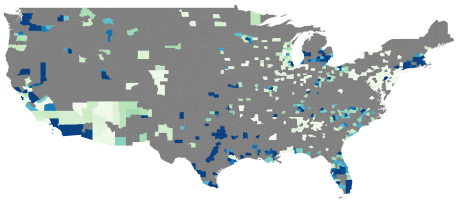
3.4 Conclusions and Open Questions

In this chapter, we discussed two frameworks that combine theory-based mechanistic models with deep learning models for spatial and temporal epidemic forecasting. They are one of the first efforts that have been made towards using mechanistic causal

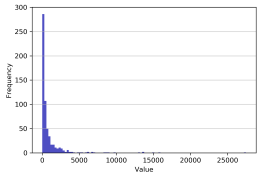
theory to enhance data-driven models.

First, TDEFSI is proposed to train an LSTM-based model with theory generated synthetic training data. The learned model can provide accurate high-resolution forecasts using low-resolution time series data. The proposed framework transfers theory-based causal mechanism from a mechanistic model to a deep learning model. Unlike data augmentation that are directly applied on observed data for time series classification or regression, the proposed framework generates synthetic high-resolution data using high-performance-computing-oriented simulations of epidemic processes over realistic social contact networks, which is not available or quite sparse in the real world. In addition, incorporating mechanistic models into the framework enables what-if forecasting by deep learning models. A direction for future work is to investigate the use of synthetic data generated by social, epidemiological, and behavioral models in conjunction with observed data to improve epidemic forecasts. In this work, we try to reduce the gap between simulated and real world data distributions by simulating with parameter settings learned from observations so that the generated epi-curves are realistic. In future work, we plan to further reduce the gap by using synthetic data based on real-time observations to train the neural networks.

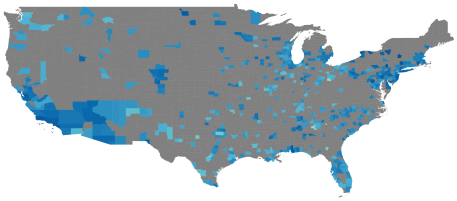
Second, CausalGNN is proposed to create a jointly learning process between deep learning models and theory-based mechanistic models. The attention-based dynamic GNN module embeds spatiotemporal features in an efficient way, leading to better spatiotemporal forecasting performance. We incorporate a causal module into the framework via a mutually learning mechanism to provide epidemiological context to the learned GNN model, leading to better long-term forecasting performance. Future directions may include: (1) testing the proposed framework on datasets of other epidemics, such as Ebola or influenza; (2) exploring what-if forecasting via the mechanistic models; (3) conducting a deeper analysis on the learned model for explainability; (4) considering fairness in AI-based epidemic forecasting.



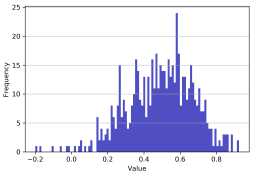
(a) US map - MAPE



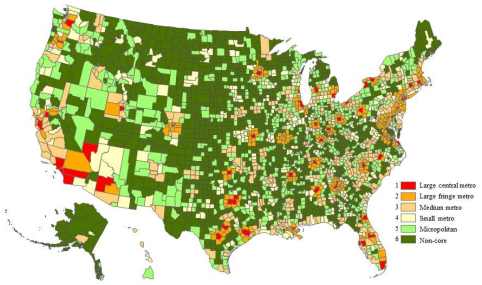
(b) Histogram - MAPE



(c) US map - PCORR



(d) Histogram - PCORR



(e) 2013 urban-rural classification scheme for counties

Figure 3.22: CausalGNN performance distribution over US counties. (a) MAPE performance across counties. The darker the smaller MAPE values. (b) MAPE distributions over counties. (c) PCORR performance across counties. The darker the smaller MAPE values. (d) PCORR distributions over counties. (e) 2013 urban-rural classification scheme for counties[6].

Chapter 4

General Conclusions and Perspectives

In this thesis, we investigated deep learning methods for reliable epidemic forecasting with the aim of producing accurate and explainable forecasts. To promote the accuracy of deep learning-based forecasting models, we investigated GNN-based frameworks that consider temporal and spatial signals using a novel large scale mobility dataset. We discussed how the mobility information helps to understand the disease transmission dynamics. To explore the explainability of deep learning-based forecasting models, we investigated a new emerging direction that combines deep learning models with theory-based mechanistic models to incorporate epidemiological context. The results are as follows.

First, we introduced a large-scale aggregated spatiotemporal mobility data and incorporated it into GNNs. The proposed model leverages priors from domain knowledge and mobility data and uses those to instruct the model learning. We showed that the proposed model can capture cross-location co-evolving disease dynamics and generate more accurate forecasts compared with baselines who do not leverage mobility information. We also showed that the proposed model provides a natural representation of disease and human mobility dynamics to develop spatially explicit forecasts thus leading to better forecasting accuracy.

As future work, the proposed method is flexible to account for any static and dynamic spatiotemporal signals and can be extended to forecast other diseases dynamics, such as seasonal flu. Also, text data, such as online posts or news, provide timely information about disease dynamics. For example, social media users may report their symptoms through online posts, which are known to be the best signals for early disease detection, even before diagnoses. However, traditional surveillance data of disease dynamics usually do not include text knowledge leading to the lack of semantic features in model learning process. Leveraging text data to generate semantic embeddings for GNN-based models' learning can be explored.

Second, we proposed two frameworks TDEFSI and CausalGNN that work towards

enhancing deep learning models with theory-based mechanistic models with the aim of providing accurate forecasts as well as gaining a mechanistic understanding from a learned model. We first proposed TDEFSI who shows a sequential learning process where mechanistic models are used to generate context-specific synthetic training data and then an LSTM-based model is trained with the synthetic data. We showed that TDEFSI produces accurate high-resolution forecasts from flat-resolution observations. We also showed that the high-performance-computing-based simulations allow us to make forecasts that are context specific and capture the underlying causal processes. Moreover, we showed that physical constraints based on epidemiological prior help improve forecasting accuracy.

We further proposed CausalGNN who adopts a jointly learning process that learns a latent space to combine the spatiotemporal and causal embeddings using graph-based non-linear transformations. We showed that the proposed model achieves better forecasting performance than the data-driven baselines who do not employ mechanistic models. Compared with the other GNN-based models in the baselines, our model has a relatively small parameter size which does not increase as the number of locations increases. We also showed that the epidemiological context provided by mechanistic simulations can be used to regularize model forecasts leading to better forecasting accuracy. To gain the understanding of the learned model, we further presented that the learned model can generate meaningful disease model parameters.

Explainability of deep learning models is scientifically challenging but also an important requirement to help policy makers trust the models when applying AI-based technologies for epidemic forecasting tasks. Future work would be to further explore the explainability of the learned model using epidemiological theory. The presented ideas that combine theory and deep learning models can be generalized to forecast other diseases such as mental health and new emerging diseases which have sparse surveillance data to train a deep learning model. Immediate uses of the proposed framework are 1) augmenting existing training datasets with realistic synthetic data to generalize the learned model for unseen data patterns, 2) generating synthetic training datasets for different temporal and spatial resolutions where real-world observations are absent. In addition, incorporating causal models into the frameworks enables what-if forecasting by deep learning models. Another possible future work is to make what-if forecasts with the proposed frameworks.

Besides accuracy and explainability of deep learning model, fairness in AI systems is also an active area of research. AI systems can be used for other critical tasks, such as predicting the presence and severity of a medical condition or matching people to jobs and partners. Any unfairness in such systems can have a wide-scale impact. Thus, as the impact of AI increases across sectors and societies, it is critical to work towards systems that are fair and inclusive for all. Fairness in an AI-based system is critical yet not well-defined problem. In epidemic forecasting domain, unfairness in the system with deep learning models exists because the models learn from existing epidemic data collected from the real world, and so an accurate model may learn

or even amplify problematic pre-existing biases in the data based on race, gender, religion or other characteristics. Furthermore, there is no standard definition of fairness in epidemic forecasting system. Given these challenges, a future work would be to build deep learning-based epidemic forecasting system that can advance fairness and inclusiveness in an epidemic at different temporal and spatial scales. I plan to define fairness in epidemic forecasting problems, assess biases in the collected datasets, and select representative datasets to train and test the model. I also plan to incorporate the defined fairness into model design, deployment, and evaluation.

References

Hani M Aburas, B Gultekin Cetiner, and Murat Sari. 2010. Dengue confirmed-cases prediction: A neural network model. *Expert Systems with Applications* 37, 6 (2010), 4256–4260.

Bijaya Adhikari, Xinfeng Xu, Naren Ramakrishnan, and B Aditya Prakash. 2019. Epideep: Exploiting embeddings for epidemic forecasting. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 577–586.

Aniruddha Adiga, Devdatt Dubhashi, Bryan Lewis, Madhav Marathe, Srinivasan Venkatramanan, and Anil Vullikanti. 2020a. Mathematical models for covid-19 pandemic: a comparative analysis. *Journal of the Indian Institute of Science* (2020), 1–15.

Aniruddha Adiga, Srinivasan Venkatramanan, James Schlitt, Akhil Peddireddy, Allan Dickerman, Andrei Bura, Andrew Warren, Brian D Klahn, Chunhong Mao, Dawen Xie, et al. 2020b. Evaluating the impact of international airline suspensions on the early global spread of COVID-19. *medRxiv* (2020).

Aniruddha Adiga, Lijing Wang, Benjamin Hurt, Akhil Sai Peddireddy, Przemyslaw Porebski, Srinivasan Venkatramanan, Bryan Lewis, and Madhav Marathe. 2021. All models are useful: Bayesian ensembling for robust high resolution covid-19 forecasting. (2021).

Aniruddha Adiga, Lijing Wang, Adam Sadilek, Ashish Tendulkar, Srinivasan Venkatramanan, Anil Vullikanti, Gaurav Aggarwal, Alok Talekar, Xue Ben, Jiangzhuo Chen, et al. 2020c. Interplay of global multi-scale human mobility, social distancing, government interventions, and COVID-19 dynamics. *medRxiv* (2020).

AHRQ. 2017. Hospital visits for a population. `https://www.ahrq.gov/data/resources/index.html`. Accessed June 01, 2017.

Marco Ajelli, Bruno Gon**c**calves, Duygu Balcan, Vittoria Colizza, Hao Hu, José J Ramasco, Stefano Merler, and Alessandro Vespignani. 2010. Comparing large-scale

computational approaches to epidemic modeling: agent-based versus structured metapopulation models. *BMC infectious diseases* 10, 1 (2010), 1–13.

Cleo Anastassopoulou, Lucia Russo, Athanasios Tsakris, and Constantinos Siettos. 2020. Data-based analysis, modelling and forecasting of the COVID-19 outbreak. *PloS one* 15, 3 (2020), e0230405.

Apple. 2020 (accessed August 29, 2020). *Mobility Trends Reports.* `https://www.apple.com/covid19/mobility`

Parul Arora, Himanshu Kumar, and Bijaya Ketan Panigrahi. 2020. Prediction and analysis of COVID-19 positive cases using deep learning models: A descriptive case study of India. *Chaos, Solitons & Fractals* (2020), 110017.

Norman TJ Bailey et al. 1975. *The mathematical theory of infectious diseases and its applications.* Number 2nd edition. Charles Griffin & Company Ltd 5a Crendon Street, High Wycombe, Bucks HP13 6LE.

Frank Ball, Tom Britton, Thomas House, Valerie Isham, Denis Mollison, Lorenzo Pellis, and Gianpaolo Scalia Tomba. 2015. Seven challenges for metapopulation models of epidemics, including households models. *Epidemics* 10 (2015), 63–67.

Hugo Barbosa, Marc Barthelemy, Gourab Ghoshal, Charlotte R James, Maxime Lenormand, Thomas Louail, Ronaldo Menezes, José J Ramasco, Filippo Simini, and Marcello Tomasini. 2018. Human mobility: Models and applications. *Physics Reports* 734 (2018), 1–74.

Batuhan Bardak and Mehmet Tan. 2015. Prediction of influenza outbreaks by integrating Wikipedia article access logs and Google flu trend data. In *2015 IEEE 15th International Conference on Bioinformatics and Bioengineering (BIBE).* IEEE, 1–6.

Aleix Bassolas, Hugo Barbosa-Filho, Brian Dickinson, Xerxes Dotiwalla, Paul Eastham, Riccardo Gallotti, Gourab Ghoshal, Bryant Gipson, Surendra A Hazarie, Henry Kautz, et al. 2019. Hierarchical organization of urban mobility and its connection with city livability. *Nature communications* 10, 1 (2019), 1–10.

Richard Beckman, Keith R Bisset, Jiangzhuo Chen, Bryan Lewis, Madhav Marathe, and Paula Stretz. 2014. Isis: A networked-epidemiology based pervasive web app for infectious disease pandemic planning and response. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 1847–1856.

Michael A Benjamin, Robert A Rigby, and D Mikis Stasinopoulos. 2003. Generalized autoregressive moving average models. *Journal of the American Statistical association* 98, 461 (2003), 214–223.

Christoph Bergmeir, Rob J Hyndman, and José M Benítez. 2016. Bagging exponential smoothing methods using STL decomposition and Box-Cox transformation. *International journal of forecasting* 32, 2 (2016), 303–312.

Keith R. Bisset, Jiangzhuo Chen, Xizhou Feng, V.S. Anil Kumar, and Madhav V. Marathe. 2009. EpiFast: A Fast Algorithm for Large Scale Realistic Epidemic Simulations on Distributed Memory Systems. In *Proceedings of the 23rd international conference on Supercomputing*. ACM, 430–439.

Isaac I Bogoch, Alexander Watts, Andrea Thomas-Bachli, Carmen Huber, Moritz UG Kraemer, and Kamran Khan. 2020. Potential for global spread of a novel coronavirus from China. *Journal of Travel Medicine* 27, 2 (2020), taaa011.

Rebecca K Borchering, Cécile Viboud, Emily Howerton, Claire P Smith, Shaun Truelove, Michael C Runge, Nicholas G Reich, Lucie Contamin, John Levander, Jessica Salerno, et al. 2021. Modeling of future COVID-19 cases, hospitalizations, and deaths, by vaccination rates and nonpharmaceutical intervention scenarios—United States, April–September 2021. *Morbidity and Mortality Weekly Report* 70, 19 (2021), 719.

Johannes Bracher, Evan L Ray, Tilmann Gneiting, and Nicholas G Reich. 2021. Evaluating epidemic forecasts in an interval format. *PLoS computational biology* 17, 2 (2021), e1008618.

Leo Breiman. 1996. Bagging predictors. *Machine learning* 24, 2 (1996), 123–140.

Logan C Brooks, David C Farrow, Sangwon Hyun, Ryan J Tibshirani, and Roni Rosenfeld. 2015. Flexible modeling of epidemics with an empirical Bayes framework. *PLoS Comput Biol* 11, 8 (2015), e1004382.

Gerrit Burgers, Peter Jan van Leeuwen, and Geir Evensen. 1998. Analysis scheme in the ensemble Kalman filter. *Monthly weather review* 126, 6 (1998), 1719–1724.

CDC. 2018. Historical Seasonal Influenza Vaccine Schedule. `https://www.cdc.gov/flu/professionals/vaccination/vaccinesupply.htm`. Accessed June 01, 2018.

CDC. 2019a. Disease Burden of Influenza. `https://www.cdc.gov/flu/about/disease/burden.htm`. Accessed April 01, 2019.

CDC. 2019b. Fluview Interactive. `https://www.cdc.gov/flu/weekly/fluviewinteractive.htm`. Accessed April 20, 2019.

CDO. 2018. Climate Data Online. `https://www.ncdc.noaa.gov/cdo-web/datasets`. Accessed August 28, 2018.

Centers for Disease Control and Prevention. 2018. *Forecast the 2018–2019 Influenza Season Collaborative Challenge; 2018.* https://predict.cdc.gov/api/v1/attachments Accessible online at https://predict.cdc.gov/api/v1/attachments/flusight%202018%E2%80%932019/flu_challenge_2018-19_tentativefinal_9.18.18.docx.

Prithwish Chakraborty, Pejman Khadivi, Bryan Lewis, Aravindan Mahendiran, Jiangzhuo Chen, Patrick Butler, Elaine O Nsoesie, Sumiko R Mekaru, John S Brownstein, Madhav V Marathe, et al. 2014. Forecasting a moving target: Ensemble models for ILI case count predictions. In *Proceedings of the 2014 SIAM international conference on data mining*. SIAM, 262–270.

Prithwish Chakraborty, Bryan Lewis, Stephen Eubank, John S Brownstein, Madhav Marathe, and Naren Ramakrishnan. 2018. What to know before forecasting the flu. *PLOS Computational Biology* 14, 10 (2018), e1005964.

Dennis L Chao, M Elizabeth Halloran, Valerie J Obenchain, and Ira M Longini Jr. 2010. FluTE, a publicly available stochastic influenza epidemic simulation model. *PLoS computational biology* 6, 1 (2010), e1000656.

Vinay Kumar Reddy Chimmula and Lei Zhang. 2020. Time series forecasting of COVID-19 transmission in Canada using LSTM networks. *Chaos, Solitons & Fractals* (2020), 109864.

Matteo Chinazzi, Jessica T Davis, Marco Ajelli, Corrado Gioannini, Maria Litvinova, Stefano Merler, Ana Pastore y Piontti, Kunpeng Mu, Luca Rossi, Kaiyuan Sun, et al. 2020. The effect of travel restrictions on the spread of the 2019 novel coronavirus (COVID-19) outbreak. *Science* 368, 6489 (2020), 395–400.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

Gerardo Chowell and Ruiyan Luo. 2021. Ensemble bootstrap methodology for forecasting dynamic growth processes using differential equations: application to epidemic outbreaks. *BMC medical research methodology* 21, 1 (2021), 1–18.

Gerardo Chowell, Lisa Sattenspiel, Shweta Bansal, and Cécile Viboud. 2016. Mathematical models to characterize early epidemic growth: A review. *Physics of life reviews* 18 (2016), 66–97.

Gerardo Chowell, Amna Tariq, and James M Hyman. 2019. A novel sub-epidemic modeling framework for short-term forecasting epidemic waves. *BMC medicine* 17, 1 (2019), 1–18.

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

Javier Contreras, Rosario Espinola, Francisco J Nogales, and Antonio J Conejo. 2003. ARIMA models to predict next-day electricity prices. *IEEE transactions on power systems* 18, 3 (2003), 1014–1020.

Zhicheng Cui, Wenlin Chen, and Yixin Chen. 2016. Multi-scale convolutional neural networks for time series classification. *arXiv preprint arXiv:1603.06995* (2016).

Raj Dandekar and George Barbastathis. 2020. Neural Network aided quarantine control model estimation of global Covid-19 spread. *arXiv preprint arXiv:2004.02752* (2020).

Raj Dandekar, Chris Rackauckas, and George Barbastathis. 2020. A Machine Learning-Aided Global Diagnostic and Comparative Tool to Assess Effect of Quarantine Control in COVID-19 Spread. *Patterns* 1, 9 (2020), 100145.

Songgaojun Deng, Huzefa Rangwala, and Yue Ning. 2019a. Learning Dynamic Context Graphs for Predicting Social Events. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1007–1016.

Songgaojun Deng, Shusen Wang, Huzefa Rangwala, Lijing Wang, and Yue Ning. 2019b. Graph Message Passing with Cross-location Attentions for Long-term ILI Prediction. *arXiv preprint arXiv:1912.10202* (2019).

Songgaojun Deng, Shusen Wang, Huzefa Rangwala, Lijing Wang, and Yue Ning. 2020. 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*. 245–254.

DOH. 2019. ILI Weekly Reports. `http://www.nj.gov/health/cd/statistics/flu-stats/`. Accessed April 20, 2019.

Colin Doms, Sarah C Kramer, and Jeffrey Shaman. 2018. Assessing the Use of Influenza Forecasts and Epidemiological Modeling in Public Health Decision Making in the United States. *Scientific reports* 8, 1 (2018), 12406.

Andrea Freyer Dugas, Mehdi Jalalpour, Yulia Gel, Scott Levin, Fred Torcaso, Takeru Igusa, and Richard E Rothman. 2013. Influenza forecasting with Google flu trends. *PloS one* 8, 2 (2013), e56176.

Ceyhun Eksin, Keith Paarporn, and Joshua S Weitz. 2019. Systematic biases in disease forecasting-the role of behavior change. *Epidemics* (2019).

Stephen Eubank, Hasan Guclu, VS Anil Kumar, Madhav V Marathe, Aravind Srinivasan, Zoltan Toroczkai, and Nan Wang. 2004. Modelling disease outbreaks in realistic urban social networks. *Nature* 429, 6988 (2004), 180–184.

James Faghmous, Hung Nguyen, Matthew Le, and Vipin Kumar. 2014. Spatio-Temporal Consistency as a Means to Identify Unlabeled Objects in a Continuous Data Field. In *AAAI Conference on Artificial Intelligence*.

Christopher C Fischer, Kevin J Tibbetts, Dane Morgan, and Gerbrand Ceder. 2006. Predicting crystal structure by merging data mining with quantum mechanics. *Nature Materials* 5 (07 2006), 641.

Antoine Flahault, Elisabeta Vergu, Laurent Coudeville, and Rebecca F Grais. 2006. Strategies for containing a global influenza pandemic. *Vaccine* 24, 44 (2006), 6751–6755.

Germain Forestier, Franccois Petitjean, Hoang Anh Dau, Geoffrey I Webb, and Eamonn Keogh. 2017. Generating synthetic time series to augment sparse datasets. In *2017 IEEE international conference on data mining (ICDM)*. IEEE, 865–870.

Yoav Freund, Robert E Schapire, et al. 1996. Experiments with a new boosting algorithm. In *icml*, Vol. 96. Citeseer, 148–156.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*. 1050–1059.

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

Song Gao, Jinmeng Rao, Yuhao Kang, Yunlei Liang, Jake Kruse, Doerte Doepfer, Ajay K Sethi, Juan Francisco Mandujano Reyes, Jonathan Patz, and Brian S Yandell. 2020. Mobile phone location data reveal the effect and geographic variation of social distancing on the spread of the COVID-19 epidemic. *arXiv preprint arXiv:2004.11430* (2020).

Yuyang Gao, Liang Zhao, Lingfei Wu, Yanfang Ye, Hui Xiong, and Chaowei Yang. 2019. Incomplete Label Multi-Task Deep Learning for Spatio-Temporal Event Subtype Forecasting. In *AAAI*.

GHT. 2018. Google Health Trends. `https://trends.google.com/trends`. Accessed August 28, 2018.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. *arXiv preprint arXiv:1704.01212* (2017).

Giulia Giordano, Franco Blanchini, Raffaele Bruno, Patrizio Colaneri, Alessandro Di Filippo, Angela Di Matteo, and Marta Colaneri. 2020. Modelling the COVID-19 epidemic and implementation of population-wide interventions in Italy. *Nature Medicine* (2020), 1–6.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*.

Tilmann Gneiting, Fadoua Balabdaoui, and Adrian E Raftery. 2007. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 69, 2 (2007), 243–268.

Tilmann Gneiting and Adrian E Raftery. 2007. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association* 102, 477 (2007), 359–378.

Google. 2018. Google Correlate Data. `https://www.google.com/trends/correlate`. Accessed August 28, 2018.

Swaminathan Gurumurthy, Ravi Kiran Sarvadevabhatla, and R Venkatesh Babu. 2017. Deligan: Generative adversarial networks for diverse and limited data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 166–174.

Andrew Harvey and Paul Kattuman. 2020. Time series models based on growth curves with applications to forecasting coronavirus. *Covid Economics, Vetted and Real-Time Papers* 24 (2020).

Geoffroy Hautier, Christopher C Fischer, Anubhav Jain, Tim Mueller, and Gerbrand Ceder. 2010. Finding Nature's Missing Ternary Oxide Compounds Using Machine Learning and Density Functional Theory. *Chemistry of Materials* 22, 12 (2010), 3762–3767.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580* (2012).

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

Nicolas Hoertel, Martin Blachier, Carlos Blanco, Mark Olfson, Marc Massetti, Marina Sánchez Rico, Frédéric Limosin, and Henri Leleu. 2020. A stochastic agent-based model of the SARS-CoV-2 epidemic in France. *Nature medicine* 26, 9 (2020), 1417–1421.

Jennifer A Hoeting, David Madigan, Adrian E Raftery, and Chris T Volinsky. 1999. Bayesian model averaging: a tutorial (with comments by M. Clyde, David Draper and EI George, and a rejoinder by the authors. *Statistical science* 14, 4 (1999), 382–417.

Zixin Hu, Qiyang Ge, Li Jin, and Momiao Xiong. 2020. Artificial intelligence forecasting of covid-19 in china. *arXiv preprint arXiv:2002.07112* (2020).

Ting Hua, Chandan K Reddy, Lei Zhang, Lijing Wang, Liang Zhao, Chang-Tien Lu, and Naren Ramakrishnan. 2018. Social Media based Simulation Models for Understanding Disease Dynamics. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, 3797–3804.

Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E Hopcroft, and Kilian Q Weinberger. 2017. Snapshot ensembles: Train 1, get m for free. *arXiv preprint arXiv:1704.00109* (2017).

De Kai, Guy-Philippe Goldstein, Alexey Morgunov, Vishal Nangalia, and Anna Rotkirch. 2020. Universal masking is urgent in the covid-19 pandemic: Seir and agent based models, empirical validation, policy recommendations. *arXiv preprint arXiv:2004.13553* (2020).

Sasikiran Kandula, Daniel Hsu, and Jeffrey Shaman. 2017. Subregional nowcasts of seasonal influenza using search trends. *Journal of medical Internet research* 19, 11 (2017), e370.

Sasikiran Kandula, Teresa Yamana, Sen Pei, Wan Yang, Haruka Morita, and Jeffrey Shaman. 2018. Evaluation of mechanistic and statistical methods in forecasting influenza-like illness. *Journal of The Royal Society Interface* 15, 144 (2018), 20180174.

Amol Kapoor, Xue Ben, Luyang Liu, Bryan Perozzi, Matt Barnes, Martin Blais, and Shawn O'Banion. 2020. Examining COVID-19 Forecasting using Spatio-Temporal Graph Neural Networks. *arXiv preprint arXiv:2007.03113* (2020).

Anuj Karpatne, Gowtham Atluri, James H Faghmous, Michael Steinbach, Arindam Banerjee, Auroop Ganguly, Shashi Shekhar, Nagiza Samatova, and Vipin Kumar. 2017. Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Transactions on Knowledge and Data Engineering* 29, 10 (2017), 2318–2331.

Jaya Kawale, Stefan Liess, Arjun Kumar, Michael Steinbach, Peter Snyder, Vipin Kumar, Auroop R Ganguly, Nagiza F Samatova, and Fredrick Semazzi. 2013. A graph-based approach to find teleconnections in climate data. *Statistical Analysis and Data Mining: The ASA Data Science Journal* 6, 3 (2013), 158–179.

Matt J Keeling and Pejman Rohani. 2011. Introduction to simple epidemic models. In *Modeling infectious diseases in humans and animals*. Princeton University Press, 15–53.

Ankush Khandelwal, Anuj Karpatne, Miriam E Marlier, Jongyoun Kim, Dennis P Lettenmaier, and Vipin Kumar. 2017. An approach for global monitoring of surface water extent variations in reservoirs using MODIS data. *Remote sensing of Environment* 202 (2017), 113–128.

Ankush Khandelwal, Varun Mithal, and Vipin Kumar. 2015. Post classification label refinement using implicit ordering constraint among data instances. In *2015 IEEE International Conference on Data Mining*. IEEE, 799–804.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

Brennan Klein, Timothy LaRock, Stefan McCabe, Leo Torres, Lisa Friedland, Filippo Privitera, Brennan Lake, Moritz UG Kraemer, John S Brownstein, David Lazer, et al. 2020. Reshaping a nation: Mobility, commuting, and contact patterns during the COVID-19 outbreak. *Northeastern University-Network Science Institute Report* (2020).

Moritz UG Kraemer, Adam Sadilek, Qian Zhang, Nahema A Marchal, Gaurav Tuli, Emily L Cohn, Yulin Hswen, T Alex Perkins, David L Smith, Robert C Reiner, et al. 2020a. Mapping global variation in human mobility. *Nature Human Behaviour* (2020), 1–11.

Moritz UG Kraemer, Chia-Hung Yang, Bernardo Gutierrez, Chieh-Hsi Wu, Brennan Klein, David M Pigott, Louis Du Plessis, Nuno R Faria, Ruoran Li, William P Hanage, et al. 2020b. The effect of human mobility and control measures on the COVID-19 epidemic in China. *Science* 368, 6490 (2020), 493–497.

Mario Michael Krell, Anett Seeland, and Su Kyoung Kim. 2018. Data Augmentation for Brain-Computer Interfaces: Analysis on Event-Related Potentials Data. *arXiv preprint arXiv:1801.02730* (2018).

Yu A Kuznetsov and Carlo Piccardi. 1994. Bifurcation analysis of periodic SEIR and SIR epidemic models. *Journal of mathematical biology* 32, 2 (1994), 109–121.

Håvard Kvamme, Nikolai Sellereite, Kjersti Aas, and Steffen Sjursen. 2018. Predicting mortgage default using convolutional neural networks. *Expert Systems with Applications* 102 (2018), 207–217.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. 2018. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 95–104.

Shengjie Lai, Nick W Ruktanonchai, Liangcai Zhou, Olivia Prosper, Wei Luo, Jessica R Floyd, Amy Wesolowski, Chi Zhang, Xiangjun Du, Hongjie Yu, et al. 2020. Effect of non-pharmaceutical interventions for containing the COVID-19 outbreak: An observational and modelling study. *medRxiv* (2020).

Arielle Lasry, Daniel Kidder, Marisa Hast, and et al. 2020. Timing of Community Mitigation and Changes in Reported COVID-19 and Community Mobility — Four U.S. Metropolitan Areas. *MMWR Morb Mortal Wkly Rep 2020* 69 (2020), 451–457. `https://doi.org/10.15585/mmwr.mm6915e2`

Arthur Le Guennec, Simon Malinowski, and Romain Tavenard. 2016. Data augmentation for time series classification using convolutional neural networks. In *ECML/PKDD workshop on advanced analytics and learning on temporal data*.

Jung Min Lee, Donghoon Choi, Giphil Cho, and Yongkuk Kim. 2012. The effect of public health interventions on the spread of influenza among cities. *Journal of theoretical biology* 293 (2012), 131–142.

Xiang Li, Ling Peng, Yuan Hu, Jing Shao, and Tianhe Chi. 2016. Deep learning architecture for air quality predictions. *Environmental Science and Pollution Research* 23, 22 (2016).

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

Dianbo Liu, Leonardo Clemente, Canelle Poirier, Xiyu Ding, Matteo Chinazzi, Jessica T Davis, Alessandro Vespignani, and Mauricio Santillana. 2020. A machine learning methodology for real-time forecasting of the 2019-2020 COVID-19 outbreak using Internet searches, news alerts, and estimates from mechanistic models. *arXiv preprint arXiv:2004.04019* (2020).

Eric T Lofgren, M Elizabeth Halloran, Caitlin M Rivers, John M Drake, Travis C Porco, Bryan Lewis, Wan Yang, Alessandro Vespignani, Jeffrey Shaman, Joseph NS Eisenberg, et al. 2014. Opinion: Mathematical models: A key tool for outbreak response. *Proceedings of the National Academy of Sciences* 111, 51 (2014), 18095–18096.

Elena Loli Piccolomini and Fabiana Zama. 2020. Monitoring Italian COVID-19 spread by a forced SEIRD model. *PloS one* 15, 8 (2020), e0237417.

Antonella Lunelli, Andrea Pugliese, and Caterina Rizzo. 2009. Epidemic patch models applied to pandemic influenza: Contact matrix, stochasticity, robustness of predictions. *Mathematical Biosciences* 220, 1 (2009), 24–33.

Luca Magri and Nguyen Anh Khoa Doan. 2020. First-principles machine learning modelling of COVID-19. *arXiv preprint arXiv:2004.09478* (2020).

Achla Marathe, Bryan Lewis, Jiangzhuo Chen, and Stephen Eubank. 2011. Sensitivity of Household Transmission to Household Contact Structure and Size. *PLoS ONE* 6 (08 2011).

Marco Marchesi. 2017. Megapixel size image creation using generative adversarial networks. *arXiv preprint arXiv:1706.00082* (2017).

Noelle-Angelique M Molinari, Ismael R Ortega-Sanchez, Mark L Messonnier, William W Thompson, Pascale M Wortley, Eric Weintraub, and Carolyn B Bridges. 2007. The annual impact of seasonal influenza in the US: measuring disease burden and costs. *Vaccine* 25, 27 (2007), 5086–5096.

Haruka Morita, Sarah Kramer, Alexandra Heaney, Harold Gil, and Jeffrey Shaman. 2018. Influenza forecast optimization when using different surveillance data types and geographic scale. *Influenza and other respiratory viruses* 12, 6 (2018), 755–764.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *ICML*.

John A Nelder and Roger Mead. 1965. A simplex method for function minimization. *The computer journal* 7, 4 (1965), 308–313.

Yue Ning, Rongrong Tao, Chandan K Reddy, Huzefa Rangwala, James C Starz, and Naren Ramakrishnan. 2018. STAPLE: Spatio-Temporal Precursor Learning for Event Forecasting. In *Proceedings of the 18th SIAM International Conference on Data Mining*. SIAM, 99–107.

Elaine Nsoesie, Madhav Mararthe, and John Brownstein. 2013b. Forecasting peaks of seasonal influenza epidemics. *PLoS currents* 5 (2013).

Elaine O Nsoesie, Richard J Beckman, Sara Shashaani, Kalyani S Nagaraj, and Madhav V Marathe. 2013a. A simulation optimization approach to epidemic forecasting. *PloS one* 8, 6 (2013), e67164.

Elaine O Nsoesie, John S Brownstein, Naren Ramakrishnan, and Madhav V Marathe. 2014a. A systematic review of studies on forecasting the dynamics of influenza outbreaks. *Influenza and other respiratory viruses* 8, 3 (2014), 309–316.

Elaine O Nsoesie, Scotland C Leman, and Madhav V Marathe. 2014b. A Dirichlet process model for classifying and forecasting epidemic curves. *BMC infectious diseases* 14, 1 (2014), 1–12.

Dave Osthus, James Gattiker, Reid Priedhorsky, Sara Y Del Valle, et al. 2019. Dynamic Bayesian Influenza Forecasting in the United States with Hierarchical Discrepancy (with Discussion). *Bayesian Analysis* 14, 1 (2019), 261–312.

Jon Parker and Joshua M Epstein. 2011. A Distributed Platform for Global-Scale Agent-Based Models of Disease Transmission. *ACM Trans Model Comput Simul* 22, 1, Article 2 (12 2011), 25 pages.

Michael J Paul, Mark Dredze, and David Broniatowski. 2014. Twitter improves influenza forecasting. *PLoS currents* 6 (2014).

Luis Perez and Jason Wang. 2017. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621* (2017).

Fotios Petropoulos and Spyros Makridakis. 2020. Forecasting the novel coronavirus COVID-19. *PloS one* 15, 3 (2020), e0231236.

Manliura Datilo Philemon, Zuhaimy Ismail, and Jayeola Dare. 2019. A review of epidemic forecasting using artificial neural networks. *International Journal of Epidemiologic Research* 6, 3 (2019), 132–143.

Jennifer M Radin, Nathan E Wineinger, Eric J Topol, and Steven R Steinhubl. 2020. Harnessing wearable device data to improve state-level real-time surveillance of influenza-like illness in the USA: a population-based study. *The Lancet Digital Health* 2, 2 (2020), e85–e93.

Ankit Ramchandani, Chao Fan, and Ali Mostafavi. 2020. DeepCOVIDNet: An Interpretable Deep Learning Model for Predictive Surveillance of COVID-19 Using Heterogeneous Features and Their Interactions. *arXiv preprint arXiv:2008.00115* (2020).

Prashant Rangarajan, Sandeep K Mody, and Madhav Marathe. 2019. Forecasting dengue and influenza incidences using a sparse representation of Google trends,

electronic health records, and time series data. *PLoS computational biology* 15, 11 (2019), e1007518.

Nicholas G Reich, Logan C Brooks, Spencer J Fox, Sasikiran Kandula, Craig J McGowan, Evan Moore, Dave Osthus, Evan L Ray, Abhinav Tushar, Teresa K Yamana, et al. 2019. A collaborative multiyear, multimodel assessment of seasonal influenza forecasting in the United States. *Proceedings of the National Academy of Sciences* 116, 8 (2019), 3146–3154.

Ye Ren, Le Zhang, and Ponnuthurai N Suganthan. 2016. Ensemble classification and regression-recent developments, applications and future directions. *IEEE Computational Intelligence Magazine* 11, 1 (2016), 41–53.

Matheus Henrique Dal Molin Ribeiro, Ramon Gomes da Silva, Viviana Cocco Mariani, and Leandro dos Santos Coelho. 2020. Short-term forecasting COVID-19 cumulative confirmed cases: Perspectives for Brazil. *Chaos, Solitons & Fractals* (2020), 109853.

Hamada Rizk, Ahmed Shokry, and Moustafa Youssef. 2019. Effectiveness of Data Augmentation in Cellular-based Localization Using Deep Learning. *arXiv preprint arXiv:1906.08171* (2019).

Benjamin N Rome and Jerry Avorn. 2020. Drug evaluation during the Covid-19 pandemic. *New England Journal of Medicine* 382, 24 (2020), 2282–2284.

Abraham Savitzky and Marcel JE Golay. 1964. Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Analytical Chemistry* 36, 8 (1964), 1627–1639.

Jan Schlüter and Thomas Grill. 2015. Exploring Data Augmentation for Improved Singing Voice Detection with Neural Networks.. In *ISMIR*. 121–126.

Jürgen Schmidhuber. 2015. Deep learning in neural networks: An overview. *Neural networks* 61 (2015), 85–117.

Ransalu Senanayake, Simon O'Callaghan, and Fabio Ramos. 2016. Predicting spatio-temporal propagation of seasonal influenza using variational Gaussian process regression. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Jeffrey Shaman and Alicia Karspeck. 2012. Forecasting seasonal outbreaks of influenza. *Proceedings of the National Academy of Sciences* (2012).

Radina P Soebiyanto, Farida Adimi, and Richard K Kiang. 2010. Modeling and predicting seasonal influenza transmission in warm regions using climatological parameters. *PloS one* 5, 3 (2010), e9450.

Farzaneh Sadat Tabataba, Prithwish Chakraborty, Naren Ramakrishnan, Srinivasan Venkatramanan, Jiangzhuo Chen, Bryan Lewis, and Madhav Marathe. 2017a. A framework for evaluating epidemic forecasts. *BMC infectious diseases* 17, 1 (2017), 1–27.

Farzaneh S Tabataba, Bryan Lewis, Milad Hosseinipour, Foroogh S Tabataba, Srinivasan Venkatramanan, Jiangzhuo Chen, Dave Higdon, and Madhav Marathe. 2017b. Epidemic forecasting framework combining agent-based models and smart beam particle filtering. In *2017 IEEE international conference on data mining (ICDM)*. IEEE, 1099–1104.

Alok Talekar, Sharad Shriram, Nidhin Vaidhiyan, Gaurav Aggarwal, Jiangzhuo Chen, Srini Venkatramanan, Lijing Wang, Aniruddha Adiga, Adam Sadilek, Ashish Tendulkar, et al. 2020. Cohorting to isolate asymptomatic spreaders: An agent-based simulation study on the Mumbai Suburban Railway. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*.

Madeleine C Thomson, FJ Doblas-Reyes, Simon J Mason, Renate Hagedorn, Stephen J Connor, Thandie Phindela, AP Morse, and TN Palmer. 2006. Malaria early warnings based on seasonal climate forecasts from multi-model ensembles. *Nature* 439, 7076 (2006), 576–579.

Robert Tibshirani. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)* 58, 1 (1996), 267–288.

Paul Tseng. 2001. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications* 109, 3 (2001), 475–494.

Ashleigh R Tuite, Amy L Greer, Michael Whelan, Anne-Luise Winter, Brenda Lee, Ping Yan, Jianhong Wu, Seyed Moghadas, David Buckeridge, Babak Pourbohloul, et al. 2010. Estimated epidemiologic parameters and morbidity associated with pandemic H1N1 influenza. *Cmaj* 182, 2 (2010), 131–136.

Terry Taewoong Um, Franz Michael Josef Pfister, Daniel Pichler, Satoshi Endo, Muriel Lang, Sandra Hirche, Urban Fietzek, and Dana Kulić. 2017. Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks. *arXiv preprint arXiv:1706.00527* (2017).

UMD. 2020. *COVID-19 impact research findings*. `https://data.covid.umd.edu/findings/index.html`

UVA. 2020 (accessed May 14, 2020). *UVA COVID-19 Surveillance Dashboard*. `https://nssac.bii.virginia.edu/covid-19/dashboard/`

Willem G Van Panhuis, Sangwon Hyun, Kayleigh Blaney, Ernesto TA Marques Jr, Giovanini E Coelho, João Bosco Siqueira Jr, Ryan Tibshirani, Jarbas B da Silva Jr, and Roni Rosenfeld. 2014. Risk of dengue for tourists and teams during the World Cup 2014 in Brazil. *PLoS Negl Trop Dis* 8, 7 (2014), e3063.

Cristina Nader Vasconcelos and Bárbara Nader Vasconcelos. 2017. Increasing deep learning melanoma classification by classical and expert knowledge based image transforms. *CoRR, abs/1702.07025* 1 (2017).

Srinivasan Venkatramanan, Jiangzhuo Chen, Sandeep Gupta, Bryan Lewis, Madhav Marathe, Henning Mortveit, and Anil Vullikanti. 2017. Spatio-temporal optimization of seasonal vaccination using a metapopulation model of influenza. In *2017 IEEE International Conference on Healthcare Informatics (ICHI)*. IEEE, 134–143.

Srinivasan Venkatramanan, Bryan Lewis, Jiangzhuo Chen, Dave Higdon, Anil Vullikanti, and Madhav Marathe. 2018. Using data-driven agent-based models for forecasting emerging infectious diseases. *Epidemics* 22 (2018), 43–49.

Srinivasan Venkatramanan, Adam Sadilek, Arindam Fadikar, Christopher L Barrett, Matthew Biggerstaff, Jiangzhuo Chen, Xerxes Dotiwalla, Paul Eastham, Bryant Gipson, Dave Higdon, et al. 2021. Forecasting influenza activity using machine-learned mobility map. *Nature communications* 12, 1 (2021), 1–12.

Siva R Venna, Amirhossein Tavanaei, Raju N Gottumukkala, Vijay V Raghavan, Anthony S Maida, and Stephen Nichols. 2019. A novel data-driven model for real-time influenza forecasting. *IEEE Access* 7 (2019), 7691–7701.

Cécile Viboud, Pierre-Yves Boëlle, Fabrice Carrat, Alain-Jacques Valleron, and Antoine Flahault. 2003. Prediction of the spread of influenza epidemics by the method of analogues. *American Journal of Epidemiology* 158, 10 (2003), 996–1006.

Svitlana Volkova, Ellyn Ayton, Katherine Porterfield, and Courtney D Corley. 2017. Forecasting influenza-like illness dynamics for military populations using neural networks and social media. *PloS one* 12, 12 (2017), e0188941.

Oyas Wahyunggoro, Adhistya Erna Permanasari, and Ahmad Chamsudin. 2013. Utilization of neural network for disease forecasting. In *59th ISI world statistics congress*. Citeseer, 549–554.

Lijing Wang, Aniruddha Adiga, Jiangzhuo Chen, Bryan Lewis, Adam Sadilek, Srinivasan Venkatramanan, and Madhav Marathe. 2021a. Combining theory and data driven approaches for epidemic forecasts. *Data Mining and Knowledge Discovery Series of CRC Press* (2021).

Lijing Wang, Aniruddha Adiga, Srinivasan Venkatramanan, Jiangzhuo Chen, Bryan Lewis, and Madhav Marathe. 2020a. Examining Deep Learning Models with Multiple Data Sources for COVID-19 Forecasting. In *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 3846–3855.

Lijing Wang, Xue Ben, Aniruddha Adiga, Adam Sadilek, Ashish Tendulkar, Srinivasan Venkatramanan, Anil Vullikanti, Gaurav Aggarwal, Alok Talekar, Jiangzhuo Chen, et al. 2021b. Using Mobility Data to Understand and Forecast COVID19 Dynamics. *The 29th International Joint Conference on Artificial Intelligence Workshop on AI for Social Good* (2021).

Lijing Wang, Jiangzhuo Chen, and Achla Marathe. 2019a. A framework for discovering health disparities among cohorts in an influenza epidemic. *World Wide Web* 22, 6 (2019), 2997–3020. https://doi.org/10.1007/s11280-018-0608-8

Lijing Wang, Jiangzhuo Chen, and Madhav Marathe. 2019b. DEFSI: Deep learning based epidemic forecasting with synthetic information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 9607–9612.

Lijing Wang, Jiangzhuo Chen, and Madhav Marathe. 2020b. TDEFSI: Theory-guided Deep Learning-based Epidemic Forecasting with Synthetic Information. *ACM Transactions on Spatial Algorithms and Systems (TSAS)* 6, 3 (2020), 1–39.

Lijing Wang, Dipanjan Ghosh, Maria Gonzalez Diaz, Ahmed Farahat, Mahbubul Alam, Chetan Gupta, Jiangzhuo Chen, and Madhav Marathe. 2020c. Wisdom of the Ensemble: Improving Consistency of Deep Learning Models. *Advances in Neural Information Processing Systems* 33 (2020).

Zheng Wang, Prithwish Chakraborty, Sumiko R Mekaru, John S Brownstein, Jieping Ye, and Naren Ramakrishnan. 2015. Dynamic poisson autoregression for influenza-like-illness case count prediction. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1285–1294.

Gregory A Wellenius, Swapnil Vispute, Valeria Espinosa, Alex Fabrikant, Thomas C Tsai, Jonathan Hennessy, Brian Williams, Krishna Gadepalli, Adam Boulange, Adam Pearce, et al. 2020. Impacts of State-Level Policies on Social Distancing in the United States Using Aggregated Mobility Data during the COVID-19 Pandemic. *arXiv preprint arXiv:2004.10172* (2020).

Paul J Werbos. 1990. Backpropagation through time: what it does and how to do it. *Proc. IEEE* 78, 10 (1990), 1550–1560.

WHO. 2019. Seasonal Influenza. http://www.who.int/news-room/fact-sheets/detail/influenza-(seasonal). Accessed April 01, 2019.

Royce J Wilson, Celia Yuxin Zhang, William Lam, Damien Desfontaines, Daniel Simmons-Marengo, and Bryant Gipson. 2019. Differentially Private SQL with Bounded User Contribution. *arXiv preprint arXiv:1909.01917* (2019).

Ken CL Wong, Linwei Wang, and Pengcheng Shi. 2009. Active model with orthotropic hyperelastic material for cardiac image analysis. In *International Conference on Functional Imaging and Modeling of the Heart*. Springer, 229–238.

Sebastien C Wong, Adam Gatt, Victor Stamatescu, and Mark D McDonnell. 2016. Understanding data augmentation for classification: when to warp?. In *2016 international conference on digital image computing: techniques and applications (DICTA)*. IEEE, 1–6.

Yuexin Wu, Yiming Yang, Hiroshi Nishiura, and Masaya Saitoh. 2018. Deep learning for epidemiological predictions. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 1085–1088.

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).

Jingjia Xu, John L Sapp, Azar Rahimi Dehaghani, Fei Gao, Milan Horacek, and Linwei Wang. 2015. Robust Transmural Electrophysiological Imaging: Integrating Sparse and Dynamic Physiological Models into ECG-Based Inference. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Nassir Navab, Joachim Hornegger, William M Wells, and Alejandro Frangi (Eds.). Springer International Publishing, Cham, 519–527.

Qinneng Xu, Yulia R Gel, L Leticia Ramirez Ramirez, Kusha Nezafati, Qingpeng Zhang, and Kwok-Leung Tsui. 2017. Forecasting influenza in Hong Kong with Google search queries and statistical model fusion. *PloS one* 12, 5 (2017), e0176690.

Teresa Yamana, Sen Pei, and Jeffrey Shaman. 2020. Projection of COVID-19 Cases and Deaths in the US as Individual States Re-open May 4, 2020. *medRxiv* (2020).

Shihao Yang, Mauricio Santillana, John S Brownstein, Josh Gray, Stewart Richardson, and SC Kou. 2017. Using electronic health records and Internet search information for accurate influenza forecasting. *BMC infectious diseases* 17, 1 (2017), 332.

Shihao Yang, Mauricio Santillana, and Samuel C Kou. 2015b. Accurate estimation of influenza epidemics using Google search data via ARGO. *Proceedings of the National Academy of Sciences* 112, 47 (2015), 14473–14478.

Wan Yang, Alicia Karspeck, and Jeffrey Shaman. 2014. Comparison of filtering methods for the modeling and retrospective forecasting of influenza epidemics. *PLoS computational biology* 10, 4 (2014), e1003583.

Wan Yang, Marc Lipsitch, and Jeffrey Shaman. 2015a. Inference of seasonal and pandemic influenza transmission dynamics. *Proceedings of the National Academy of Sciences* 112, 9 (2015), 2723–2728.

Wan Yang, Donald R Olson, and Jeffrey Shaman. 2016. Forecasting influenza outbreaks in boroughs and neighborhoods of New York City. *PLoS computational biology* 12, 11 (2016), e1005201.

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.

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

Abdelhafid Zeroual, Fouzi Harrou, Abdelkader Dairi, and Ying Sun. 2020. Deep learning methods for forecasting COVID-19 time-Series data: A Comparative study. *Chaos, Solitons & Fractals* 140 (2020), 110121.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2016. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530* (2016).

Xingyu Zhang, Yuanyuan Liu, Min Yang, Tao Zhang, Alistair A Young, and Xiaosong Li. 2013. Comparative study of four time series methods in forecasting typhoid fever incidence in China. *PloS one* 8, 5 (2013), e63116.

Liang Zhao, Jiangzhuo Chen, Feng Chen, Wei Wang, Chang-Tien Lu, and Naren Ramakrishnan. 2015a. Simnest: Social media nested epidemic simulation via online semi-supervised deep learning. In *2015 IEEE International Conference on Data Mining*. IEEE, 639–648.

Liang Zhao, Qian Sun, Jieping Ye, Feng Chen, Chang-Tien Lu, and Naren Ramakrishnan. 2015b. Multi-task learning for spatio-temporal event forecasting. In *KDD*. 1503–1512.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*. 2223–2232.

Xianglei Zhu, Bofeng Fu, Yaodong Yang, Yu Ma, Jianye Hao, Siqi Chen, Shuang Liu, Tiegang Li, Sen Liu, Weiming Guo, et al. 2019. Attention-based recurrent neural network for influenza epidemic prediction. *BMC bioinformatics* 20, 18 (2019), 1–10.