CausalGNN: Causal-based Graph Neural Networks for Spatio-Temporal Epidemic Forecasting (Supplementary Material - Technical Appendix)

Lijing Wang,[1,2] **Aniruddha Adiga,** [2] **Jiangzhuo Chen,** [2] **Adam Sadilek,** [3] **Srinivasan Venkatramanan,** [2] **Madhav Marathe** [1,2]

[1] Computer Science, University of Virginia
[2] Biocomplexity Institute and Initiative, University of Virginia
[3] Google Research
lw8bn,aa5dw,chenj,srini,marathe@virginia.edu sadilekadam@google.com

Notation

Notations and their descriptions used in the paper is shown in Table 1.

Table 1: Notations and their descriptions

Notation	Description
N	number of regions
K	historical window size of training data
h	horizon of a prediction
C	number of features
$G(\mathcal{V}, \mathcal{E}, \mathcal{T})$	dynamic graph of N regions with T time points
$\mathbf{A}_t \in \mathbb{R}^{N \times N}$	attention matrix
$\mathbf{C}_t \in \mathbb{R}^{N \times C}$	matrix of node features for N regions
$\mathbf{Q}_t \in \mathbb{R}^{N \times 4}$	matrix of causal features for N regions
$\mathbf{P}_t \in \mathbb{R}^{N \times 3}$	matrix of causal parameters for N regions
$\mathbf{H}_t^c, \mathbf{H}_t^f, \mathbf{H}_t, \tilde{\mathbf{H}}_t$	matrices of hidden states of causal encoding, feature encoding, spatial encoding, and temporal encoding
F_c, F_f, F, F_s	hidden dimensions
$\hat{\mathbf{Y}}, \hat{\mathbf{Y}}^c$	matrix of predictions for N regions
$\mathbf{Y}, \mathbf{Y}^c$	matrix of true values for N regions

Pseudo Code of the Proposed Framework

The pseudocode of the model training process is described in Algorithm 1.

Algorithm 1: CausalGNN training

Input: $G(\mathcal{V}, \mathcal{E}, \mathcal{T})$; Historical window size K; forecast horizon h.
Output: Model parameters Θ

$b \leftarrow$ a batch training sample
for *each instance* $\in b$ **do**
  $\mathbf{Q}_{T-K+1} \leftarrow \mathbf{Q}^g_{T-K+1}$ ▷ Initializing causal features with real data
  for t *in* $T-K+1 \dots T$ **do**
    $\mathbf{H}_t^c \leftarrow \text{CE}(\mathbf{Q}_t)$ ▷ Causal encoding
    $\mathbf{H}_t^f \leftarrow \text{FE}(\mathbf{C}_t)$ ▷ Feature encoding
    $\tilde{\mathbf{H}}_t \leftarrow \text{TE}(\mathbf{H}_t^f, \mathbf{H}_t^c, \mathbf{H}_{t-1})$ ▷ Temporal embedding, $\mathbf{H}_{T-K} \leftarrow \mathbf{H}^f_{T-K+1}$
    $\mathbf{H}_t \leftarrow \text{AGCN}(\tilde{\mathbf{H}}_t, \mathbf{A}_t)$ ▷ Spatial embedding
    $\mathbf{P}_t \leftarrow \text{CD}(\tilde{\mathbf{H}}_t)$ ▷ 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* $T+1 \dots T+h-1$ **do**
    $\hat{\mathbf{Y}}^c_{t+1}, \mathbf{Q}_{t+1} \leftarrow \text{SIRD}(\mathbf{Q}_t, \mathbf{P}_T)$ ▷ Causal simulating for another $h-1$ steps
  $\mathbf{H}^c_{T+h} \leftarrow \text{CE}(\mathbf{Q}_{T+h})$ ▷ Causal encoding
  $\hat{Y} \leftarrow \text{Output}(\mathbf{H}_T, \mathbf{H}^c_{T+h})$ ▷ 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

Experimental Settings

In this section, we will provide additional information for reproducing experiments.

Data Sources

Disease dynamics datasets were collected via the JHU COVID-19 surveillance dashboard[1]. **Geographical adjacency datasets**: Country adjacency and US state adjacency matrices are manually collected and cleaned. US county adjacency is downloaded from the US Census Bureau[2]. **Population datasets**: The country population (2020) data is collected from the worldometers website [3]. The US state and county population (2019) datasets are downloaded from the US Census Bureau[4].

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

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

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

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

Metrics

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

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

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

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

- The **Mean absolute percentage error (MAPE)** measures the size of the error between two variables in percentage terms:

$$\text{MAPE} = (\frac{1}{n}\sum_{i=1}^{n}|\frac{z_i - \hat{z}_i}{z_i + 1}|) * 100 \tag{2}$$

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

Setting of Baselines

Unless specified in this section, in our experiment, we use the same parameter settings as those described in the original paper.

- **SIR** is a single-patched SIR compartmental model. We calibrate model parameters based on surveillance data (daily new confirmed cases) for each region. Predictions are made by persisting the current parameter values to the future time points and run simulations.
- **PatchSEIR** is a network-based SEIR compartmental model for influenza forecasting. We use a gravity model to generate a network flow of mobility. We use the same rationale with SIR method for calibrating and predicting.
- **Autoregressive (AR)** uses observations from previous time steps as the input to a regression equation to predict the value at the next time step. We adopt an AR model of order 28.
- **Autoregressive Moving Average (ARMA)** 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.
- **Recurrent Neural Network (RNN)** is a one layer RNN model with hidden state dimension as 32.
- **Gated Recurrent Unit (GRU)** is a one layer GRU model with hidden state dimension as 32.
- **Long-Short Term Memory (LSTM)** is a one layer LSTM model with hidden state dimension as 32.
- **DCRNN** combines graph convolution networks with recurrent neural networks in an encoder-decoder manner for traffic forecasting.
- **CNNRNN-Res** combines CNN, RNN, and residual links in one framework for influenza forecasting. It employs RNN to encode temporal information and CNN to fuse information from data of different regions. We set the residual window size as 3 and all the other parameters are set as the same as the original paper.
- **LSTNet** uses CNN and RNN to extract short-term local dependency patterns among variables and to discover long-term patterns for time series trends in traffic forecasting.
- **STGCN** integrates graph convolution and gated temporal convolution through spatio-temporal convolutional blocks for traffic forecasting.
- **Cola-GNN** uses location-aware attention graph neural networks to combine graph structures and time series features in a dynamic propagation process.
- **STAN** integrates disease dynamics theory into graph neural network training for COVID-19 forecasting. Partial data such as ICU visits are not available for our selected regions thus has been omitted from the model implementation.

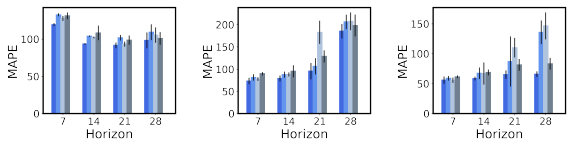
Note that SIR, PatchSEIR, CNNRNN-Res, Cola-GNN, and STAN are proposed for epidemic forecasting while DCRNN, LSTNet, and STGCN are proposed for traffic forecasting.

More Experimental Results

In this section, we will present further experimental results including ablation study in terms of MAPE, sensitivity analysis on major hyperparameters, fairness analysis of the proposed model across regions, and more examples and detailed analysis on epidemiological context.

Ablation Study (MAPE)

We present the comparison of forecasting performance in terms of MAPE for CausalGNN, CausalGNN w/o csl, CausalGNN w/o grf, and CausalGNN w/o att. The observations are similar with MAE (described in the main paper) thus are omitted here for the sake of brevity.



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

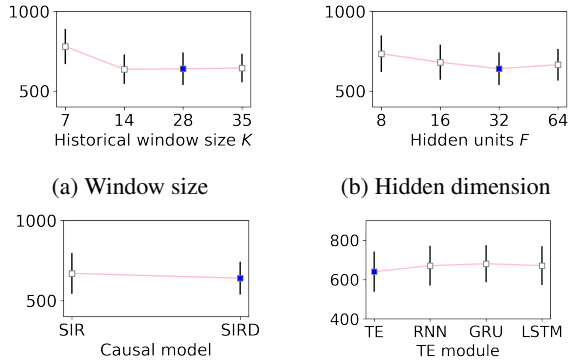
Figure 1: Ablation analysis on major components of the proposed model (MAPE).

Sensitivity Analysis

In this section, we show sensitivity analysis on some of the hyperparameters of CausalGNN: historical window size K (Figure 2a), hidden dimension F (i.e., $F_c = F_f = 2F_s = F$) (Figure 2b), causal model (Figure 2c), and TE module (Figure 2d). Except the varying hyperparameters, all the

other settings are the same with parameter settings described in the main paper. We report MAE performance on the US-State dataset with horizon=28 in Figure 2. The blue squares correspond to the same results from the main experiment. The MAPE performance shows similar observations thus are omitted for the sake of brevity.

Major observations and discussion: 1) Figure 2a shows that the model performance gets improved when K increases from 7 to 14, however, there is no obvious improvement after $K = 14$. We choose 28 in the main experiment as it equals to the historical window size K. 2) Figure 2b shows that the performance gets improved as F increases. We observe no performance improvement by increasing F value from 32 to 64. We choose 32 in the main experiment to optimize the model performance. 3) The results in Figure 2c 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, with the availability of data, we recommend to use a model that is as realistic as possible to mitigate the forecasting error imported by an assumption bias. 4) Figure 2d shows that the model performance does not vary too much in terms of TE module type. We choose the current TE module for its smaller parameter size compared with RNN, GRU, and LSTM.



(a) Window size (b) Hidden dimension

(c) Causal model (d) TE module

Figure 2: Sensitivity analysis on (a) historical window size K: 7, 14, 28, 35; (b) hidden dimension F: 8, 16, 32, 64; (c) causal model: SIR, SIRD; (d) TE module: TE, RNN, GRU, LSTM.

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. There is no standard method in the community to evaluate a system's fairness. In this section, we perform fairness analysis on our model by evaluating its performance across regions with a broad range of demographic distributions and other variability. The administrative divisions, e.g., US states or US counties, which are recognized divisions of a country, are valid regions to perform fairness analysis. We will evaluate the CausalGNN performance across US counties.

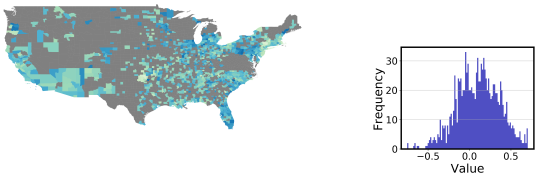
By the definition of MAE and MAPE in Equation 1 and 2, we know that MAE is scale dependant while MAPE is scale independent. MAPE can be used to compare a model on different regions. However, it divides the absolute error of the model by the actual data values. If there are data values close to 0, which is true for daily new confirmed cases of many counties during testing days (i.e., from March 21, 2021, to April 23, 2021), dividing by those very small values greatly inflates the value of MAPE. To remove the bias imported by MAPE, we computed Pearson correlation (PCORR) between predicted curves and the ground truth curves for every counties:

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

$$\text{PCORR} = \frac{\sum_{i=1}^{m}(\hat{z}_i - \bar{\hat{z}})(z_i - \bar{z})}{\sqrt{\sum_{i=1}^{m}(\hat{z}_i - \bar{\hat{z}})^2}\sqrt{\sum_{i=1}^{m}(z_i - \bar{z})^2}} \quad (3)$$

  PCORR ranges in $[-1, +1]$ and larger values are better. PCORR metric is scale independent thus can be used to compare a model performance across different regions.

Figure 3a shows the choropleth plot with counties shaded according to their PCORR performance values for 1351 counties. The colorbar ranges from -1 to 1 while the darker color represents the larger PCORR values and the grey color shows invalid counties in our experiment. In Figure 3b we show a corresponding histogram over PCORR values. We observe 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.



(a) US map (b) Histogram

Figure 3: CausalGNN performance distribution over US counties. (a) The choropleth plot with counties in PCORR performance. (b) The histogram of PCORR values.

Epidemiological Context in More Examples

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. To illustrate how the causal module can help in improving the model performance, we show three examples in the main paper. In this section, we show more examples by

comparing the 7 days ahead forecasts of confirmed cases on April 18, 2021 by CausalGNN (blue dots) and CausalGNN w/o csl (red crosses) for 50 US states in Figure 4. In each subplot, the black curve represents the ground truth curve while the orange curve represents the generated causal forecasts by the causal module in CausalGNN. Both solid lines and dots are smoothed by Savitzky–Golay filter[5] with window size 7 and polynomial order 1 to remove biases in daily reporting of cases. The shaded area is the input window.

Major observations and discussion: Similar with the observations in the main paper, we can observe that the causal module can generate meaningful curves (orange curves) compared with the ground truth curves (black curves) in most states. This indicates that CausalGNN can reveal mechanistic causal process by producing meaningful causal parameters to provide meaningful epidemiological context for GNN learning in our experiment settings. We can also observe that CausalGNN makes a better forecast (blue dots) than CausalGNN w/o csl (red crosses) for most states (i.e., the blue dots are closer to the black curves than the red crosses on the forecasting day). This means that the causal module proposed in our model can help in improving the model performance generally.

NOTE: The examples we present here do not mean that our model can learn meaningful parameters for all forecasting days in all regions, but this is a good start of building explainable deep learning models for epidemic forecasting by the proposed model. More systematic and rigorous experimental analysis is needed in the future. 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 the causal module. This would be our future work.

[5]https://en.wikipedia.org/wiki/Savitzky\%E2\%80\%93Golay_filter

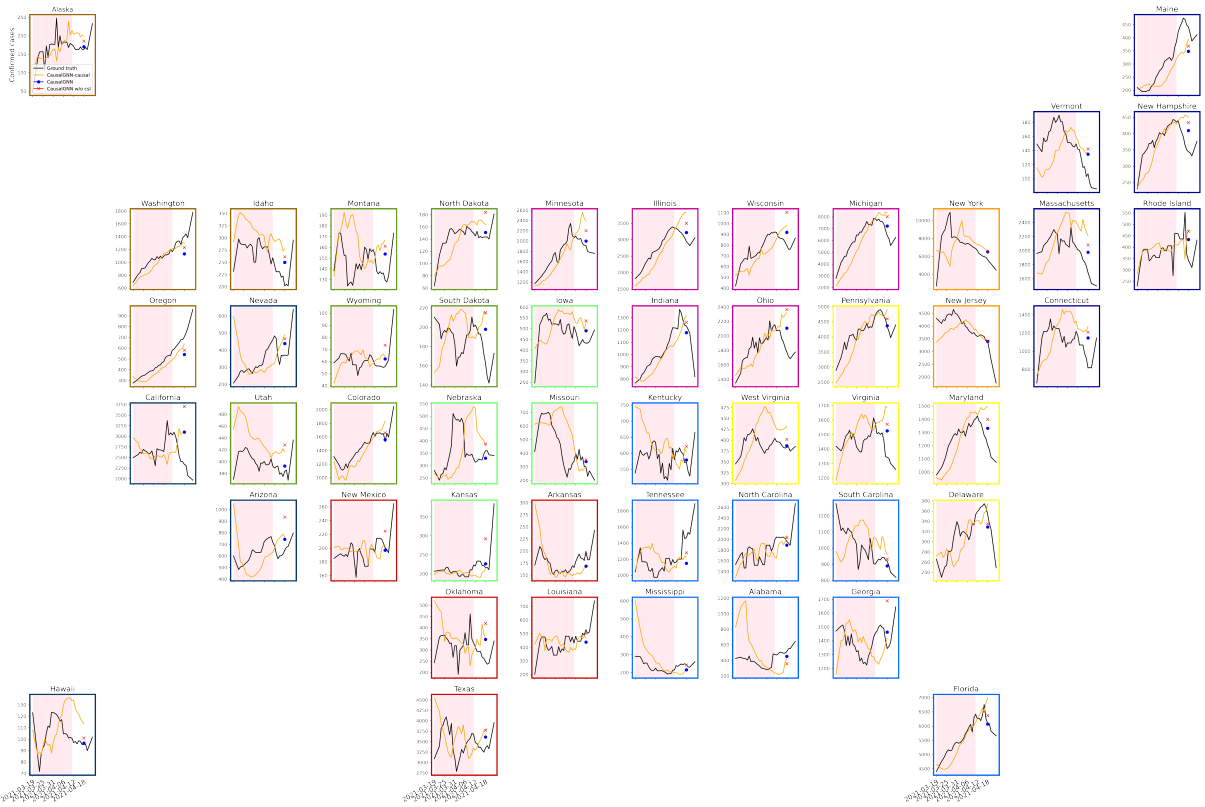


Figure 4: US state level forecasts of COVID-19 new confirmed cases with horizon 7 on April 18, 2021 by CausalGNN (blue dots) and CausalGNN w/o csl (red crosses). The black curves represent the ground truth while the orange curves represent the predicted curves of generated causal forecasts by the causal module in CausalGNN. Both solid lines and dots are smoothed values. The shaded area is the input window.