STAGCN: SPATIAL-TEMPORAL ATTENTION BASED GRAPH CONVOLUTIONAL NETWORKS FOR COVID-19 FORECASTING

Nevasini Sasikumar*
Department of Computer Science
PES University
Bengaluru, India
nevasini24@gmail.com

Krishna Sri Ipsit Mantri
ipsit.iitb@gmail.com

ABSTRACT

The recent outbreak of the novel coronavirus known as the COVID-19 pandemic has harmed the lives of millions of people across the globe and has imposed a significant threat to global healthcare due to its severe transmission capacity. It's of utmost importance to be able to accurately forecast the COVID-19 pandemic and to provide the necessary precautionary measures to protect the health of individuals and prevent the spread of this deadly widespread virus. In this paper, we propose to forecast the upcoming newly infected patients that are likely to be affected by COVID-19 in prior using a novel deep learning framework, Spatio-Temporal Attention Based Graph Convolution Networks (STAGCN) to effectively make use of spatial and temporal relationships. Instead of using traditional time-series forecasting techniques at a single city using raw data, we model the problem using graphs and aim at taking into account the dependency that an infection in one city has on its neighbors. Our experiments show that STAGCN effectively captures this dependency and consistently outperforms the other conventional methods.

1 INTRODUCTION

The recent COVID-19 pandemic more elaborately also known as coronavirus disease is a severe acute respiratory syndrome caused by the SARS-CoV-2 virus, the disease was first identified in a hospital in Wuhan, China in December 2019, since then it has spread widely transforming from a local epidemic to a major global pandemic across the whole world[1]. This virus can stay without causing any symptoms for a period of 14 days this is called a pre-symptomatic and asymptomatic transmission. COVID-19 primarily spreads through respiratory droplets when an infected person talks, coughs, or sneezes and is within 6 feet of distance, the virus can survive on surfaces for different intervals of time depending on the type of surface and the conditions prevalent. Early forecasting of the number of COVID-19 cases will help to control the incubation and prevent the spread of cases in the respective city accordingly.

Lately, artificial intelligence and machine learning techniques are widely used in numerous sectors including healthcare to predict the patient's health status beforehand and to replace the lack of experienced doctors. Using time series with AI mutually involves analyses of health records over time. It majorly aims in using time series data and build models that can provide early warnings of possible potential health issues, the most appropriate treatment planning, and population health management. Time-series forecasting has always been useful in epidemic and pandemic management (Xu et al., 2020). Further, the multi-head attention mechanism (Cordonnier et al., 2020), provides a technique for tokens at different positions in the sequence to interact with each other and compute weights that quantify the relative importance of the tokens and focus on specific parts of the time series data when making predictions eventually aiding in improving accuracy.

*Correspondence to: Nevasini Sasikumar <nevasini24@gmail.com>

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

The graph is a universal language for describing complex systems and the relations between them (Riaz & Ali, 2011). Graphs are fundamental data structures, the various factors and their interactions that are responsible for COVID-19 spread can be modeled as graph entities. However, the pandemic monitoring bodies do not record these interactions and only record the count of infectious cases, recoveries, and deaths[2]. We propose a novel method to model the data inspired by STGCN (Yu et al., 2018).

Our main contributions can be summarized as follows[3]

1. **Modeling graph entities for COVID-19:** A novel way to model COVID-19 stats pertaining to the number of cases across various connected cities into graphs by computing the physical distance between cities.
2. **Spatio-Temporal Multi-Head Attention:** A multi-head attention-based graph representation convolution learning framework to capture the spatial and temporal dependencies present among the graph entities.

2 PRELIMINARIES

2.1 COVID-19 INFECTION FORECASTING FOR MULTIPLE CITIES

Disease forecasting is a typical time-series prediction problem in which we try to predict the number of infections for the next Q days given the information available from the previous P days. Mathematically,

$$\hat{v}_{t+1}, \cdots, \hat{v}_{t+Q} = \underset{v_{t+1},\cdots,v_{t+Q}}{\arg\max} \log \mathbb{P}\Big(v_{t+1}, \cdots, v_{t+Q} \,|\, v_{t-P+1}, \cdots, v_t\Big) \tag{1}$$

where $v_t \in \mathbb{R}^N$ is an observation vector of N cities at time step t and records the daily COVID-19 infections data.

2.2 MODELING DATA INTO A GRAPH STRUCTURE

In this paper, we argue that the daily new infections are dependent not only on a single city but also on neighboring cities because of the disease spread dynamics. Hence, instead of predicting the infections for each city separately, we model all the cities on a graph where each node denotes the city and each edge denotes the connection between the cities based on a pruned weighted adjacency matrix.

Mathematically, we denote a COVID-19 spread network as a weighted undirected graph $\mathcal{G} = (V, E, W)$. Where V denotes a set of N cities which are represented as nodes, E denotes the set of edges, and W represents the weighted adjacency matrix which is calculated as follows:

$$w_{ij} = \begin{cases} \exp(-\frac{d_{ij}^2}{\sigma^2}) & \text{, if } \exp(-\frac{d_{ij}^2}{\sigma^2}) \geq \epsilon \\ 0 & \text{, otherwise} \end{cases} \tag{2}$$

Here, d_{ij} denotes the actual physical distance between the cities i and j, and σ, ϵ are thresholds to control the sparsity of the weighted adjacency matrix. Therefore, the data point v_t can be considered as a graph signal that is defined on an undirected graph $\mathcal{G}$ with weights W.

2.3 CONVOLUTIONS ON GRAPHS

The convolution operation is generalized to the graph structure in both spatial and spectral domains (Balcilar et al., 2020). However, for time series forecasting, we discuss the spectral convolution that uses the graph Laplacian. Denote $x \in \mathbb{R}^n$, a kernel Θ, and $L = I_n - D^{-\frac{1}{2}} L D^{\frac{1}{2}}$ the normalized graph Laplacian with Fourier basis $U \in \mathbb{R}^{n \times n}$, Λ the diagonal matrix of eigen values of L. We define the spectral graph convolution as follows:

$$\Theta *_{\mathcal{G}} x = \Theta(L)x = \Theta(U\Lambda U^\top)x = U\Theta(\Lambda)U^\top x \tag{3}$$

[2]https://covid19.who.int/data

[3]You can find our code here

3 PROPOSED METHOD

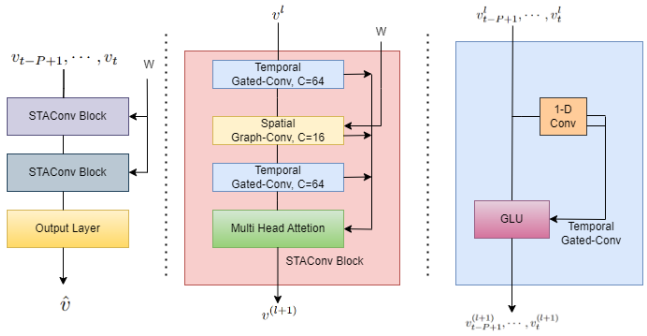

Figure 1: **Proposed STAGCN Architecture**. We stack mutiple STAConv layers. Here $\hat{v}$ denotes the final predictions. The output layer is the same as that in STGCN (Yu et al., 2018)

3.1 NETWORK ARCHITECTURE

In this section, we propose the architecture for STAGCN. As shown in the figure 1, STAGCN consists of multiple spatio-temporal attention convolutional (STAConv) blocks. Each of these blocks has two temporal gated convolutional layers and one spatial graph convolutional layer sandwiched between them. The outputs of these three layers are concatenated and combined using an attention mechanism across the channels of these layers. Denoting P as the length of the history of temporal data in the train set and the input to each block as $\{v_{t-P+1}, \cdots, v_t\}$

3.2 GCNS FOR SPATIAL EMBEDDINGS

We use the 1st-order approximation of graph Laplacian for the graph convolutional layers (Kipf & Welling, 2016) to reduce the computational complexity and also the number of parameters of the model.

We can extend the graph convolutional operation defined in equation (3) to multi-channel graph signals. Denoting a signal $X \in \mathbb{R}^{n \times C_i}$ with C_i channels, the generalized graph convolution is given by

$$y_j = \sum_{i=1}^{C_i} \Theta_{i,j}(L) x_i \ \ , 1 \leq j \leq C_0 \tag{4}$$

where C_i and C_0 are input and output embedding with $\Theta \in \mathbb{R}^{(K \times C_i)}$

3.3 GATED CNNS FOR TEMPORAL EMBEDDING

This is composed of the temporal convolutional layer that contains 1-D casual convolutional followed by GLU for non-linearity (Gehring et al., 2017).

$$\Gamma *_{\mathcal{T}} Y = P \odot \sigma(Q) \ \in \mathbb{R}^{(M-K_t+1) \times C_0} \tag{5}$$

3.4 STACONV BLOCK

Taking inspiration from Vaswani et al. (2017) and Cordonnier et al. (2020), we propose to combine the outputs of each of the spatial and temporal using multi-head attention. For the input v^l of block

l, the output v^{l+1} is computed by

$$v_i^{(l+1)} = \Big\|_{m=1}^{M} \Big[\sum_{v_k \in v^l \setminus \{v_i\}} \alpha_{i,k}^{(m)} \, c_k^{(l)} \Big] \tag{6}$$

$$c^{(l)} = \Big\| \Big[\Gamma_0^l *_{\mathcal{T}} v^l, \; \Theta^l *_{\mathcal{G}} \left(\Gamma_0^l *_{\mathcal{T}} v^l \right), \; \Gamma_1^l *_{\mathcal{T}} \text{ReLU}\Big(\Theta^l *_{\mathcal{G}} \left(\Gamma_0^l *_{\mathcal{T}} v^l \right) \Big) \Big] \tag{7}$$

where $\Big\|$ denotes the concatenation operation, M is the number of attention heads and $f_1^{(m)}$, $f_2^{(m)}$ are some neural networks used to compute the score function $s_{i,k}^{(m)}$, a_k indicates the kth coordinate of the vector a, and $\alpha_{i,k}^{(m)}$ denote the attention weights. Refer to appendix A.2 for detailed equations.

4 Experiments and Results

4.1 Dataset Information

California COVID-19 Dataset The raw data from California State Government's Open Data Portal (California Open Data Portal, 2020) was taken pre-processed using the Z-Score method, for the period 1st Feb 2020 to 17th Jan 2023. More information about data pre-processing can be found in appendix A.3.

4.2 Experimental Setup

We use a historical time window of 30 days to forecast the new cases for the next 10 days.

Metrics To measure and evaluate the performance of different methods, we use Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE), Weighted Mean Absolute Percentage Error (WMAPE).

Baselines We compare STAGCN with the following baselines methods (1) STGCN with ChebConv (2) STGCN with GraphConv (3) STAGCN with ChebConv (4) STAGCN with GraphConv.

4.3 Experimental Results

The table 1 demonstrates the results of our experiments. It can be seen that STAGCN out-performs STGCN with by a significant margin, which demonstrate that our proposal of modelling the COVID-19 forecasting on a graph and using multi-head attention is valid.

Method	Test Loss ↓	MAE ↓	RMSE ↓	WMAPE ↓	MAPE ↓
STGCN (GraphConv)	0.065590	58.798785	297.323978	0.70136494	0.107144
STGCN (ChebConv)	0.057346	45.506019	195.012181	0.54280588	0.070275
STAGCN (GraphConv)	**0.043253**	**34.245379**	**109.544507**	**0.40848647**	**0.039475**
STAGCN (ChebConv)	0.050865	38.478212	132.962695	0.45897664	0.047914

Table 1: **Comparison of performances of STAGCN and STGCN**

5 Conclusion

We propose a STAGCN network to predict the number of COVID-19 cases for time steps ahead by taking into account the proximity linked to the reported cases. We model a graph architecture further supplementing it with multi-head attention to attain early forecasting results about the number of cases that are likely to spread among the residents of the neighbourhood. Our experiments show that we outperform the baselines by a significant margin which validates the assumption behind using the attention mechanism.

REFERENCES

Muhammet Balcilar, Guillaume Renton, Pierre Heroux, Benoit Gauzere, Sebastien Adam, and Paul Honeine. Bridging the gap between spectral and spatial domains in graph neural networks, 2020. URL https://arxiv.org/abs/2003.11702.

California Open Data Portal. Statewide covid-19 cases deaths tests, 2020. URL https://data.ca.gov/dataset/covid-19-time-series-metrics-by-county-and-state/resource/30331e8f-4679-4ee9-908b-df4512065563.

Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. Multi-head attention: Collaborate instead of concatenate. *CoRR*, abs/2006.16362, 2020. URL https://arxiv.org/abs/2006.16362.

Songgaojun Deng, Shusen Wang, Huzefa Rangwala, Lijing Wang, and Yue Ning. Cola-gnn: Cross-location attention based graph neural networks for long-term ili prediction. In *Proceedings of the 29th ACM International Conference on Information amp; Knowledge Management*, CIKM '20, pp. 245–254, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450368599. doi: 10.1145/3340531.3411975. URL https://doi.org/10.1145/3340531.3411975.

Grzegorz Dudek, Slawek Smyl, and Paweł Pełka. Recurrent neural networks for forecasting time series with multiple seasonality: A comparative study, 2022. URL https://arxiv.org/abs/2203.09170.

Junyi Gao, Rakshith Sharma, Cheng Qian, Lucas M. Glass, Jeffrey Spaeder, Justin Romberg, Jimeng Sun, and Cao Xiao. Stan: Spatio-temporal attention network for pandemic prediction using real world evidence, 2020. URL https://arxiv.org/abs/2008.04215.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *International conference on machine learning*, pp. 1243–1252. PMLR, 2017.

Amol Kapoor, Xue Ben, Luyang Liu, Bryan Perozzi, Matt Barnes, Martin Blais, and Shawn O'Banion. Examining covid-19 forecasting using spatio-temporal graph neural networks, 2020. URL https://arxiv.org/abs/2007.03113.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016. URL http://arxiv.org/abs/1609.02907.

Jia Li, Zhichao Han, Hong Cheng, Jiao Su, Pengyun Wang, Jianfeng Zhang, and Lujia Pan. Predicting path failure in time-evolving graphs, 2019. URL https://arxiv.org/abs/1905.03994.

Ferozuddin Riaz and Khidir M. Ali. Applications of graph theory in computer science. In *2011 Third International Conference on Computational Intelligence, Communication Systems and Networks*, pp. 142–145, 2011. doi: 10.1109/CICSyN.2011.40.

Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. Structured sequence modeling with graph convolutional recurrent networks, 2016. URL https://arxiv.org/abs/1612.07659.

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Yuankai Wu and Huachun Tan. Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework. *arXiv preprint arXiv:1612.01022*, 2016.

Bin Xu, Jiayuan Li, and Mengqiao Wang. Epidemiological and time series analysis on the incidence and death of aids and hiv in china. *BMC Public Health*, 20(1):1–10, 2020.

Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, jul 2018. doi: 10.24963/ijcai.2018/505. URL https://doi.org/10.24963%2Fijcai.2018%2F505.

A Appendix

A.1 Related Work

Deep learning-based approaches: These work revolve around using deep learning algorithms to analyze time series data. Time series data is often complex and non-linear, making it difficult to analyze using traditional methods. All the previous methods typically involve using recurrent neural networks (RNNs) and long short-term memory-less (Dudek et al., 2022), these can perform effectively well with sequential time-series data. However, they suffer from the problem of error accumulation during iterative training. More recent works propose the use of convolutional neural networks (CNNs) (Wu & Tan, 2016) based architectures to take advantage of spatial structure along with an RNN for the time domain. One of the key advantages of using CNNs for time series forecasting is that they can handle variable-length time series data. The other advancements over this include convolutional LSTM (Shi et al., 2015). Despite them being the first attempts at mutually combining spatial and temporal modalities, these can be applied to only grid-like structures and take into consideration only the local patterns in the data.

Graph Neural Network (GNNs): Graph Neural networks treat every data point as nodes in the graph taking into account the relationship between data points instead of modeling them individually. Graph Convolutional Neural Networks (GCNs) (Kipf & Welling, 2016) operate on the graph's adjacency matrix and apply convolutional operations to extract features from the graph, one of the significances of using GCNs for time series forecasting is that they can handle missing data in the time series, which is a common issue in time series forecasting. Many GCN-based methods were proposed for spatiotemporal analysis by Deng et al. (2020), Kapoor et al. (2020) and Gao et al. (2020). Other works combining recurrent networks and GNNs include GConvGRU and GConvLSTM (Seo et al., 2016), GC-LSTM (Seo et al., 2016), LRGCN (Li et al., 2019) etc.,

A.2 STAConv Details

For the input v^l of block l, the output v^{l+1} is computed by

$$v_i^{(l+1)} = \Big\|_{m=1}^{M} \Big[\sum_{v_k \in v^l \setminus \{v_i\}} \alpha_{i,k}^{(m)} \, c_k^{(l)} \Big] \tag{8}$$

$$s_{i,k}^{(m)} = \big\langle \, f_1^{(m)}(c_i^{(l)}), \, f_2^{(m)}(c_k^{(l)}) \, \big\rangle \tag{9}$$

$$\alpha_{i,k}^{(m)} = \frac{\exp(s_{i,k}^{(m)})}{\sum_j \exp(s_{i,j}^{(m)})} \tag{10}$$

$$c^{(l)} = \Big\| \Big[\Gamma_0^l *_{\mathcal{T}} v^l, \, \Theta^l *_{\mathcal{G}} \big(\Gamma_0^l *_{\mathcal{T}} v^l\big), \, \Gamma_1^l *_{\mathcal{T}} \text{ReLU}\Big(\Theta^l *_{\mathcal{G}} \big(\Gamma_0^l *_{\mathcal{T}} v^l\big)\Big) \Big] \tag{11}$$

The STAConv block can jointly process the outputs from both the spatial and temporal domains. This layer is flexible and can dynamically stack any number of blocks and can be set accordingly by the user. Our major work lies in this block we bring in a multi-head-based attention mechanism that can parallelly respond to the outputs of multiple spatio-temporal blocks.

A.3 DATASET PREPROCESSING

This dataset contains daily new infections count for every county in California, apart from cumulative cases, deaths, tests, etc. We then construct a complete graph with nodes as these counties and edge weights as the average geodesic distance between every pair of counties. We further alter the weighted adjacency matrix as indicated in section 2.2. We used $\sigma = 100$ and $\epsilon = 0.5$.

A.4 TRAINING SETTINGS

We train our STAGCN-ChebConv using the spectral graph convolution layers with 1st-order Chebyshev polynomials approximation. For every model in the experiments, mean squared error is used as the loss function with RMSProp optimizer, trained using early stopping with the patience of 30 and batch size of 32. Both the graph convolution kernel K_s and temporal convolution kernel K_t are set to 3. Learning rate decay is used with initial value 10^{-3} with decay rate of 0.7 after. We divided the dataset of size 1083 samples in the ratio $70 : 15 : 15$ for the train, validation and test split. We used 3 STGCN blocks in our experiments and found to be giving us the best results.

A.5 PREDICTIONS

We plot the average predictions for each of the 58 counties of California for STAGCN and STGCN. They are as follows:

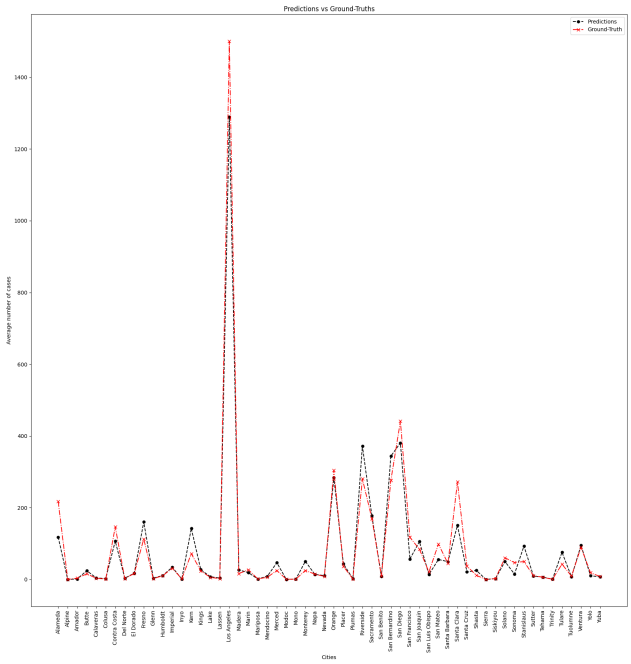

Figure 2: STAGCN with ChebConv, trained using early stopping with of patience 10 epochs

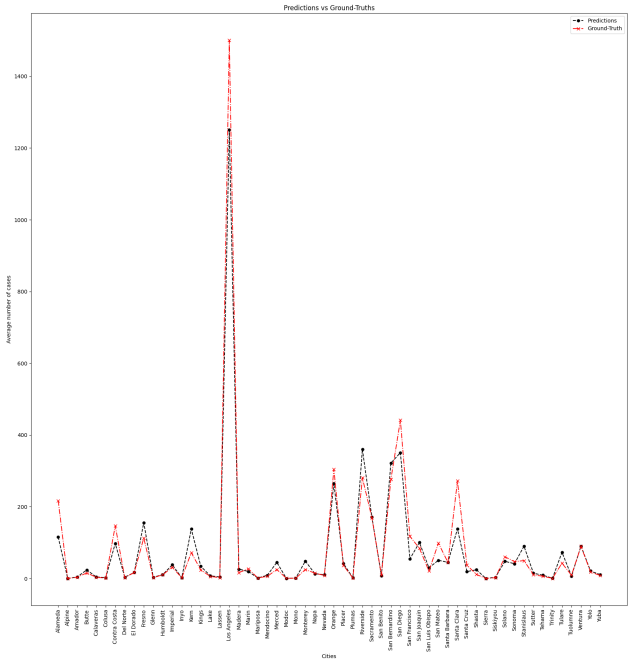

Figure 3: STAGCN with GraphConv, trained using early stopping with of patience 20 epochs

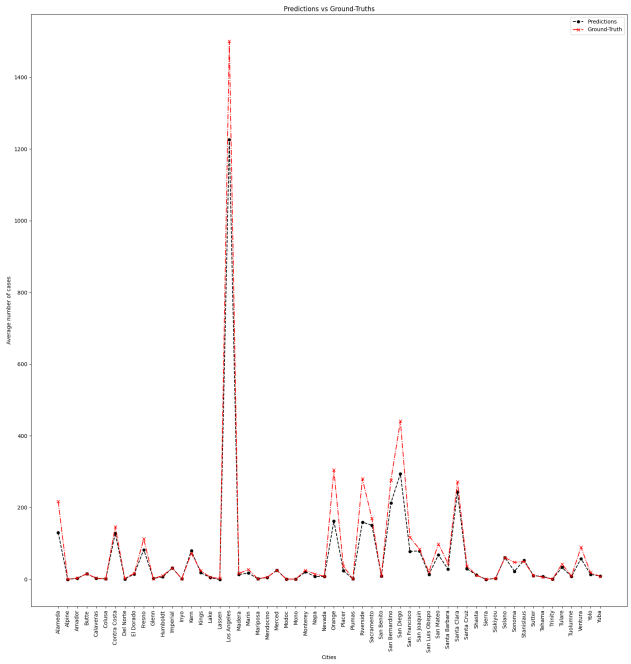

Figure 4: STGCN with ChebConv, trained using early stopping with of patience 20 epochs

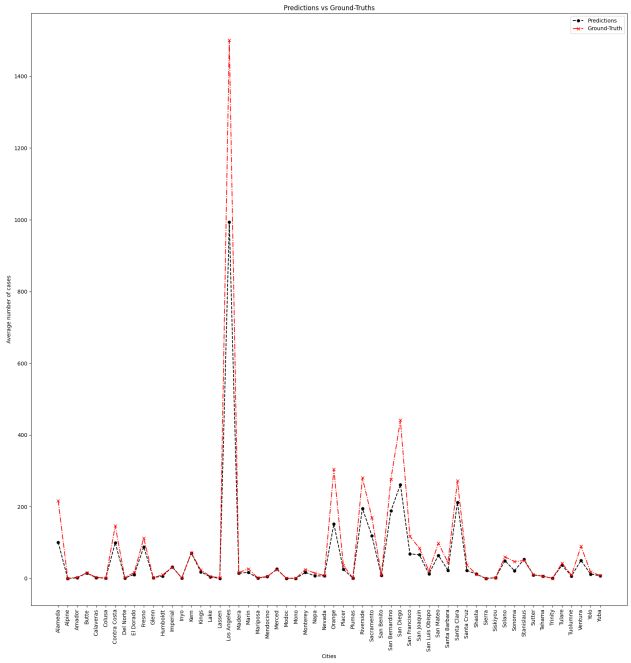

Figure 5: STGCN with GraphConv, trained using early stopping with of patience 10 epochs