Predicting COVID-19 pandemic by spatio-temporal graph neural networks: A New Zealand's study

Viet Bach Nguyen *
FPT Software AI Center
Hanoi, Vietnam

Truong Son Hy * †
Indiana State University
Terre Haute, USA

Long Tran-Thanh
University of Warwick
Coventry, United Kingdom

Nhung Nghiem
University of Otago
Wellington, New Zealand

Abstract

Modeling and simulations of pandemic dynamics play an essential role in understanding and addressing the spreading of highly infectious diseases such as COVID-19. In this work, we propose a novel deep learning architecture named Attention-based Multiresolution Graph Neural Networks (ATMGNN) that learns to combine the spatial graph information, i.e. geographical data, with the temporal information, i.e. timeseries data of number of COVID-19 cases, to predict the future dynamics of the pandemic. The key innovation is that our method can capture the multiscale structures of the spatial graph via a learning-to-cluster algorithm in a data-driven manner. This allows our architecture to learn to pick up either local or global signals of a pandemic, and model both the long-range spatial and temporal dependencies. Importantly, we collected and assembled a new dataset for New Zealand. We established a comprehensive benchmark of statistical methods, temporal architectures, graph neural networks along with our spatio-temporal model. We also incorporated socioeconomic cross-sectional data to further enhance our prediction. Our proposed model have shown highly robust predictions and outperformed all other baselines in various metrics for our new dataset of New Zealand. Our data and source code are publicly available at `https://github.com/HySonLab/pandemic_tgnn`.

1 Introduction

The Coronavirus Disease started in 2019 (COVID-19) has been and currently is a major global pandemic, challenging every country's population and public health systems. As a fairly water-isolated island country, New Zealand mostly contained the spread of COVID-19 until early 2022, when infection cases surged to more than 2 million confirmed cases by the end of the year (WHO data, `https://covid19.who.int/region/wpro/country/nz`). While New Zealand responded promptly, contained and effectively vaccinated the population to keep the case number low, the sudden rise in infections posed certain challenges to the healthcare system.

In the wake of the spread of COVID-19, many epidemiological modeling and prediction models emerged, seeking to project the progression of the pandemic and inform public health authorities to take measures when appropriate. To model non-linear disease growth functions, artificial neural networks and deep learning models have been developed and trained to predict the infection case time

*Co-first authors
†Correspondent author

Temporal Graph Learning Workshop @ NeurIPS 2023, New Orleans.

series of each health area. The most common types of deep learning models for epidemic modeling are Long Short-Term Memory (LSTM)-based models, in which the architecture is specially designed to learn and represent historical or temporal information [49].

It is shown that incorporating geospatial information, including but not limited to movement and connectedness information, helps with the forecasting performance of LSTM-based and deep learning model [33]. One of the classes of deep learning models that can seamlessly embed geospatial information is Graph Neural Networks (GNNs), neural network deep learning models that can capture topological information in graph- and network-based data [54]. Following in the footsteps of previous efforts at COVID-19 forecasting with GNNs and spatial disease features [41], we propose improved spatiotemporal graph neural network models that can accurately learn and forecast COVID-19 case progression in New Zealand. To this end, we gathered and reformated New Zealand COVID-19 case data, and constructed day-to-day disease graphs based on geographical information; graph disease representations are then fed to a hierarchical, multi-resolution temporal graph neural network model that can automatically group multiple disease areas to learn large-scale disease properties [19, 23].

2 Related works

Various traditional statistical and linear models have been employed to forecast the spread of COVID-19 cases. Among these traditional models are the Susceptible, Infectious, or Recovered (SEIR) models [51, 43, 17]. While mathematical models such as SEIR can estimate the effect of control measures even before the start of the pandemic, these models cannot make accurate predictions due to a lack of data and their inherent assumptions restricting the class of available learnable disease functions [44, 6]. ARIMA models are applied the case of COVID-19 [28, 2, 53]; Meta-developed Prophet model and its variants are also used in forecasting the number of cases in India [45] and generally for any country using day level case information [29, 4].

The vanilla neural networks without any additional component have been tested as predictors of COVID-19 outbreaks across several countries, owing to their high capacity modeling of disease patterns and functions when certain assumptions (e.g., disease incubation period) are encoded [38]. Recurrent neural networks (RNNs) are frequently used in the context of COVID-19 modeling. LSTM-based models were used to simulate and forecast the COVID-19 pandemic in several other countries, either independently or in conjunction with various distinct statistical models incorporating spatial features [40, 34, 48], albeit without incorporating spatial information of the pandemic.

Graph neural networks (GNNs) utilizing various ways of generalizing the concept of convolution to graphs [47, 39, 30] have been widely applied to many learning tasks, including modeling physical systems [3], finding molecular representations to estimate quantum chemical computation [8, 26, 13, 21, 22], and protein interface prediction [11]. One of the most popular types of GNNs is message passing neural nets (MPNNs) [13] that are constructed based on the message passing scheme in which each node propagates and aggregates information, encoded by vectorized messages, to and from its local neighborhood. In order to capture the dynamic nature of evolving features or connectivity over time, temporal graph neural networks (TGNN) have been proposed by [46, 23] as a generic, efficient deep learning framework that combines graph encoding (e.g., MPNNs) with time-series encoding architectures (e.g., LSTM, Tranformers, etc.). Applications of TGNN include traffic prediction [7, 31, 37] and learning on brain networks [37], etc.

3 Methods

3.1 Graph Neural Networks

3.1.1 Graph construction

We process the input disease data as graphs, a form of non-Euclidean irregular data that is *permutation invariant* in nature (i.e., changing the ordering of the nodes in a graph does not change the data that the graph represents). To represent the New Zealand pandemic data as graphs, the entirety of the country is formatted as a single graph $G = (V, E)$, where $n = \mathsf{V}$ is the number of nodes, and each node represents a single district health board in New Zealand. We create a series of graphs $G^{(1)}, G^{(2)}, ..., G^{(T)}$ corresponding to each day in the case dataset of New Zealand, where the current day t is within the available day case data for every district health board. The topology

(i.e., connecting edges and adjacency matrix) of the graphs remains constant over all time steps. The adjacency matrix $\mathbf{A}$ represents the connection between edges in the disease graph; we constructed the connections between nodes based on geographical adjacency between any two district health boards. Between any two district health boards u and v, the edge (u, v) from u to v is $A_{u,v} = 2$ if two district health boards share any border length, and $A_{u,v} = 1$ otherwise. For each node or district health board, we denote the features, or the number of cases in the last d days in the region u, as the vector $\mathbf{x}_u^{(t)} = (c_u^{(t-d)}, ..., c_u^{(t)})^\top \in \mathbb{R}^d$. The number of cases over multiple previous days is used to account for irregular case reporting and the length of the incubation period.

3.1.2 Message-passing neural networks

We model the spatial and geographical spread of COVID-19 in New Zealand using a well-known family of GNNs known as message-passing neural networks (MPNNs) [13]. Across multiple layers and to account for the vectorization of the node embeddings, we define the neighborhood aggregation scheme as

$$H^{(k)} = \sigma(\tilde{A}H^{(k-1)}W^{(k)}) \tag{1}$$

where $H^{(k-1)}$ is a matrix containing the generated node embeddings from the previous layer, $H^{(k)} = (h_1^{(k)}, h_2^{(k)}, ..., h_n^{(k)})^\top$ denotes the matrix arrangement of the node embeddings of all nodes in the graph ($H^{(0)} = X$), and $\tilde{A}$ denotes the aforementioned normalized graph Laplacian, with hidden embedding $h_u^{(k)}$ representing each node/district health board $u \in V$. The time index is omitted from both equations; the model is in fact applied to all input graphs $G^{(1)}, G^{(2)}, ..., G^{(T)}$ in the time series separately. Since the connectivity and adjacency of the disease graphs are constant over time, the matrix $\tilde{A}$ is shared across all temporal graphs alongside the weight matrices $W^{(1)}, ..., W^{(K)}$ for K message-passing layers, while the node embeddings $H^0, ..., H^K$ are unique for each disease day graph in the time series.

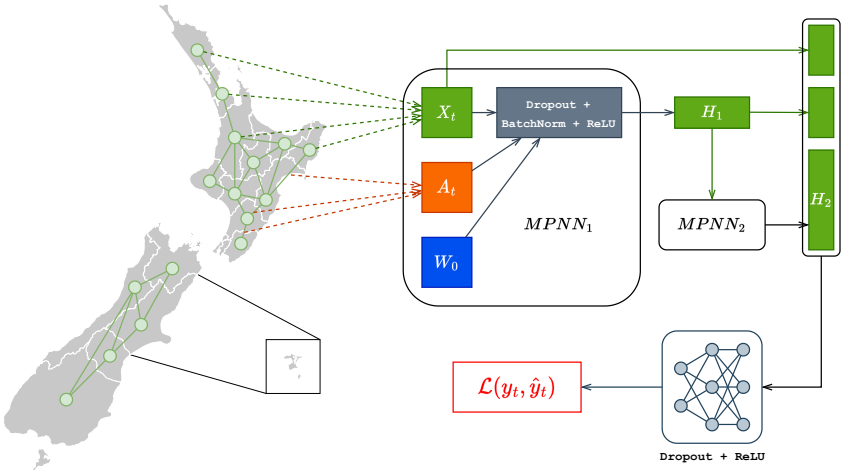

Figure 1: Overview of the proposed Message-Passing Neural Network architecture on the graph representation of New Zealand. Note that dotted green arrows represent the extraction of historical case counts as node features, and dotted orange arrows represent the geospatial location between two regions extracted as edge features.

3.1.3 Multiresolution Graph Neural Networks

In field of graph learning, it is important to build a graph neural network that can capture the multiscale and hierarchical structures of graphs. Multiresolution Graph Neural Networks (MGNN) was originally proposed by [20] as a graph encoder in the context of graph generation via variational autoencoder, and adopted by [23] in combination with a temporal architecture to learn and predict the dynamics of an epidemic or a pandemic. Instead of a fixed coarse-graining process, MGNN introduces a learnable clustering algorithm that iteratively constructs a hierarchy of coarsening graphs, also called multiresolution or multiple levels of resolutions (see Def. 3.2):

1. Based on the node embeddings, we cluster a graph into multiple partitions. Each partition is coarsened into a single node, and all the edges connecting between two partitions are aggregated into a single edge (see Def. 3.1). This process results into a smaller coarsened graph.
2. We continue to apply message passing on the coarsened graph to produce its node embeddings, and then cluster it further. On each resolution, all the node embeddings are pooled into a single graph-level vectorized representation, i.e. latent. The hierarchy of latents allows us to capture both local information (in the lower levels) and global information (in the higher levels) of a graph.

Definition 3.1. A k-cluster partition on a graph $G = (V, E)$ partitions its set of nodes into k disjoint sets $\{V_1, V_2, .., V_k\}$. A coarsening of G is a graph $\tilde{G} = (\tilde{V}, \tilde{E})$ of k nodes in which node $\tilde{v}_i \in \tilde{V}$ corresponds to a induced subgraph of G on V_i. The weighted adjacency matrix $\tilde{A} \in \mathbb{N}^{k \times k}$ of $\tilde{G}$ is defined as:

$$\tilde{A}_{ij} = \begin{cases} \frac{1}{2} \sum_{u,v \in V_i} A_{uv}, & \text{if } i = j, \\ \sum_{u \in V_i, v \in V_j} A_{uv}, & \text{if } i \neq j, \end{cases}$$

where the diagonal of $\tilde{A}$ denotes the number of edges inside each cluster, while the off-diagonal denotes the number of edges between two clusters.

Definition 3.2. An L-level of resolutions, i.e. multiresolution, of a graph G is a series of L graphs $\tilde{G}_1, .., \tilde{G}_L$ in which: **(i)** $\tilde{G}_L$ is G itself; and **(ii)** For $1 \leq \ell \leq L - 1$, $\tilde{G}_\ell$ is a coarsening graph of $\tilde{G}_{\ell+1}$ as defined in Def. 3.1. The number of nodes in $\tilde{G}_\ell$ is equal to the number of clusters in $\tilde{G}_{\ell+1}$. The top level coarsening $\tilde{G}_1$ is a graph consisting of a single node.

The key innovation of MGNN is how the model can learn to cluster graph $\tilde{G}_{\ell+1}$ into $\tilde{G}_\ell$ in a data-driven manner. Without the loss of generality, we suppose that the number of nodes in $\tilde{G}_\ell$ is K, i.e. $\|\tilde{V}_\ell\| = K$, meaning that we cluster $\tilde{G}_{\ell+1}$ into K partitions. First, we employ a GNN to produce a K-channel node embedding for each node of $\tilde{G}_{\ell+1}$. Then, we apply a softmax over the node embedding to compute the probability of assigning each node to one of the K clusters. However, we want each node to be in a single cluster, i.e. hard clustering, thus we employ the Gumbel-max trick [14, 24, 35] to sample/select the cluster based on the assignment probability while maintaining differentiability for back-propagation. This results into an assignment matrix $P \in \{0, 1\}^{\|\tilde{V}_{\ell+1}\| \times K}$. The adjacency matrix of $\tilde{G}_\ell$ can be computed as $\tilde{A}_\ell = P^T \tilde{A}_{\ell+1} P$. We repeat this clustering process iteratively in order to build multiple resolutions of coarsening graphs.

3.1.4 Spatio-temporal graph neural networks

In this section, we build our spatio-temporal GNNs by combining all the previously defined modules. Suppose that we are given a historical data of T timesteps which can be modeled by T input graphs $G^{(1)}, G^{(2)}, .., G^{(T)}$. The simplest combination is MPNN+LSTM in which we employ MPNN (see Section 3.1.2) to encode each $G^{(t)}$ into a graph-level vectorized representation and then feed it into an LSTM backbone (as formulated in [18]). Furthermore, we want to capture the multiscale information, i.e. local to global, that is essential in modeling the long-range spatial and temporal dependencies. Thus, instead of MPNN, we apply MGNN (see Section 3.1.3) to construct a hierarchy of latents (i.e. each latent is a graph-level representation for a resolution) for each graph $G^{(t)}$. At the t-th timestep, a multi-head self-attention mechanism (as formulated in [50]) is applied to encode the hierarchy of latents into a single vector that will be fed further into a temporal architecture. Finally, another multi-head self-attention mechanism block is used, instead of LSTM, as the temporal backbone. We call this novel architecture as Attention-based Multiresolution Graph Neural Networks or ATMGNN.

3.2 Data preprocessing

New Zealand daily new cases with graphs Official data originally obtained is in tabular form with information regarding the sex, age group, district health board location, case status and travel of each COVID-19 infected patient. All cases are filtered so that only cases in 2022 and cases that are confirmed are included in the dataset. For each district health board, on each day, all confirmed cases regardless of sex or age group are aggregated and counted toward the daily new cases count. From the geographical map of the district health boards, an adjacency matrix that represents the topology

Model	Next 3 Days			Next 7 Days			Next 14 Days			Next 21 Days		
	MAE	RMSE	R^2	MAE	RMSE	R^2	MAE	RMSE	R^2	MAE	RMSE	R^2
AVG	247.20	325.92	-3.22	258.95	340.95	-3.58	277.16	362.98	-4.22	292.23	379.98	-4.87
AVG_WINDOW	80.88	111.15	0.76	104.09	142.37	0.55	144.88	196.63	-0.02	176.82	238.39	-0.79
LAST_DAY	118.81	158.56	0.64	73.65	102.09	0.84	120.99	164.78	0.47	156.17	211.44	-0.08
LIN_REG	182.46	284.61	0.31	213.53	336.56	-0.01	272.77	440.60	-0.77	335.95	551.23	-1.81
GP_REG	331.43	471.17	-0.89	332.08	472.45	-0.98	325.55	464.20	-0.97	322.08	460.23	-0.96
RAND_FOREST	98.97	152.96	0.80	72.85	111.69	0.89	112.02	168.81	0.74	140.77	210.73	0.59
XGBOOST	109.68	165.36	0.77	**68.51**	**105.91**	**0.90**	108.45	165.17	0.75	137.45	208.13	0.60
PROPHET	119.32	642.78	-0.24	148.58	770.55	-1.50	222.01	526.58	-0.59	292.54	407.66	-0.17
ARIMA	132.49	534.26	0.14	155.44	523.57	-0.15	204.51	472.95	-0.28	239.06	423.17	-0.26
LSTM	186.86	242.62	-0.97	168.43	222.65	-0.39	140.69	192.39	0.38	128.04	182.35	0.59
MPNN	80.33	110.75	0.84	87.45	121.23	0.79	121.41	168.34	0.53	153.62	210.69	0.15
MGNN	80.87	111.67	0.83	89.77	124.56	0.74	125.30	172.46	0.46	156.25	213.55	0.06
MPNN+LSTM	**75.25**	**104.64**	**0.86**	85.14	117.92	0.84	**88.28**	**121.71**	**0.85**	**99.85**	**137.74**	**0.83**
ATMGNN	77.49	106.96	**0.86**	86.85	119.68	0.84	90.43	124.89	0.84	101.87	140.33	0.82

Table 1: New Zealand: Performance of all experimental model evaluated based on the metrics specified in Section 4.4

of the disease graph is generated by connecting each board to itself with a unit weighted edge, and each board to every other board that shares any part of its border with edges weighted 2. Original data is imported and transformed using the Python packages Pandas and NumPy [52, 16], while disease graphs are built with the included code and the NetworkX [15] package. All data that was preprocessed and converted to graph form between March 4, 2022 and November 4, 2022 is available on GitHub.

New Zealand economic features Official categorical GDP data is obtained from NZStats [1], with the original data containing GDP information in terms of NZ dollars for each predefined administrative region and for every 22 available economic industry categories. All economic feature vectors are normalized (via mean and standard deviation) to allow the models to learn properly and mitigate exploding/vanishing gradients.

4 Experiments

4.1 Experiment task description

We comprehensively evaluate the forecasting effectiveness of the models in short-, mid-, and long-term prediction windows. The models are trained and assessed on their predictions 3, 7, 14, and 21 days from the input data. Data from day 1 to day T is used to train one model at a time, and then predictions are obtained from the model from day $T + 1$ to day $T + d$, where d is the prediction window size and $0 < d < 22$. Note that each model within a single class of models is trained separately and specifically for a single fixed time window. In other words, two different models are trained to predict days $T + a$ and $T + b$, where $a, b > 0$ and $a \neq b$. The size of the training set gradually increases as time progresses, and for each value of T the best model is identified via a validation set with no overlapping day with the test set. We trained and validated the models on the time series data from March 4, 2022 to September 4, 2022, and performed further model evaluations to examine generalization performance on an out-of-distribution starting from September 4, 2022 to November 4, 2022.

4.2 Baselines and Comparisons

We compare the different spatio-temporal models with traditional statistical prediction and neural network-based regression models that have been recently applied to the problem of COVID-19 forecasting.

Simple statistical models The class of most rudimentary statistical models for forecasting. The models examined include (1) AVG: The average number of cases for one region up to the time of the test day, (2) AVG_WINDOW: The average number of cases in the past d-day window for one region, and (3) LAST_DAY: The prediction for the next day in one region is the same as the number of cases on the previous day.

Traditional machine learning models The input format for all models in this class is the case history up to the prediction date of each district health board. The models examined include (4) LIN_REG [25]: Ordinary least squares Linear Regression, (5) GP_REG [27]: Gaussian Process Regressor, (6) RAND_FOREST [12]: A random forest regression model, and (7) XGBOOST [9]: An improved version of the random forest regression model using gradient boosting.

Parameterized regression time-series forecasting models The class of linear regression models with specific components represented as parameters. The models examined include (8) PROPHET [36]: A forecasting model for various types of time series that has also seen extensive use in forecasting COVID-19 where the input is the entire time-series historical number of cases of one region up to before the testing day; (9) ARIMA [28]: A simple autoregressive moving average model, which the input is similar to PROPHET; and (10) LSTM [5]: A two-layer bidirectional LSTM model.

Graph neural network-based models The proposal graph models to be compared to the previous baseline models, with and without temporal components. The models examined include (11) MPNN [13]: Message-passing neural network model with separate layers for each day in the case time series; (12) MGNN [23, 20]: Message-passing neural network model similar to MPNN, but with multiple graph resolutions and learned clustering of different regions; (13) MPNN + LSTM: Message-passing neural network model with long-short term memory neural time series model; and (14) ATMGNN: Multiresolution graph model based on the MGNN model combined with Transformers for modeling time series. All models in this category are described in detail in Section 3.

4.3 Experimental setup

We detail the hyperparameters setup of the deep learning prediction models in our experiments. For all graph-based models (MPNN, MPNN+LSTM, ATMGNN), training lasts for a maximum of 300 epochs with early stopping patience of 50 epochs, and early stopping is only enabled from epoch 100 onward. Models are trained with the Adam optimizer ($lr = 10^{-3}$), batch size 128. For the neighborhood aggregation layers of the graph models, batch normalization is applied to the output of all layers with dropout applied to 0.5 times the total number of nodes. The LSTM component of the MPNN+LSTM model is implemented with a hidden state size of 64. The multiresolution component of the ATMGNN is configured for two additional coarsening layers with 10- and 5-node clusters, respectively; self-attention is configured with a single head for all regions. The models with the lowest validation loss at each prediction day shift are saved as parameter checkpoints for the sake of further evaluation, and validation information is outputted for further examination. All models are implemented with PyTorch [42] and PyTorch geometric [10].

4.4 Evaluation metrics

We measured the performance of all models with three evaluation metrics: Mean absolute error (MAE), root mean squared error (RMSE), and the coefficient of determination (R^2-score).

4.5 Observations

Performance measurement Across the board, MPNN+LSTM is the highest-performing model, with relatively low mean error and root means square error, alongside accurate trend prediction at R^2-score consistently over 0.8. Other baseline methods performed inconsistently across different time ranges, with massive fluctuations in heuristic statistical methods (AVG, AVG_WINDOW, and especially LAST_DAY), owning to these baselines simply forecasting based on rudimentary statistics of the data. The class of traditional machine learning models performed reasonably well, with tree-based methods RAND_FOREST and XGBOOST outperforming simple statistical methods and parameterized models aside from LSTM. In New Zealand from 14 days prediction length onwards, graph-based temporal models on average see a 20.31% and 25.37% relative reduction in MAE and RMSE, respectively; while correlation metric R^2 relatively improving 9.43%. Similar results are obtained from cross-examining Italy and England COVID datasets, with graph-based temporal models (e.g., MPNN+LSTM, ATMGNN) generally outperforming other baseline models and LSTM models coming second on all metrics (see Appendix).

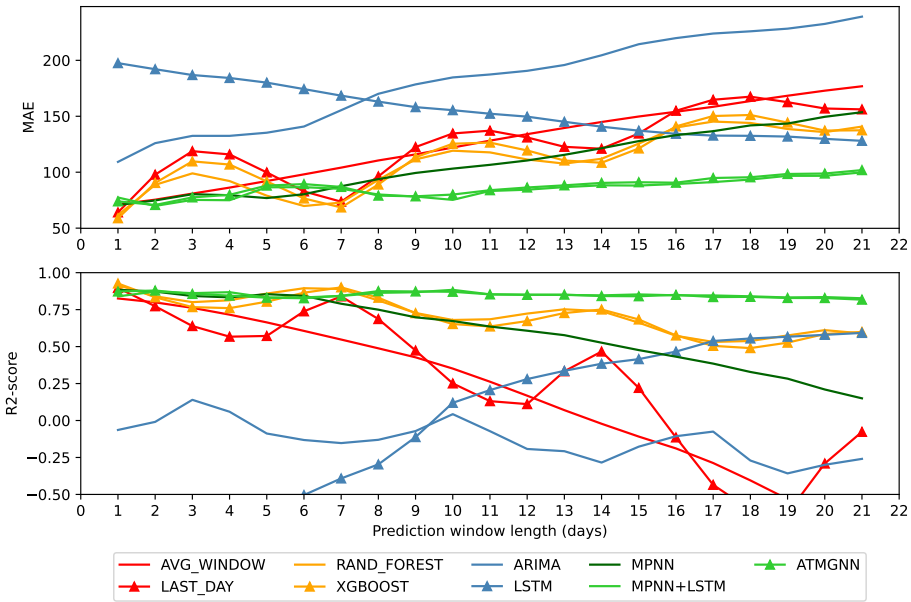

Figure 2: Performance decay with respect to MAE and R^2 metrics. Models with performance worse than the defined y-axis range are excluded.

Performance decay over long forecasting windows Across all models, the AVG model performed the worst when it comes to performance decay relative to the length of forecasting windows, concerning both absolute error and correlation metrics. On the other hand, the two other heuristic statistical methods, AVG_WINDOW and LAST_DAY, outperformed regression-based methods ARIMA and PROPHET with respect to the rate of decay and error increment over longer forecasting windows (Figure 2). Graph- and temporal-hybrid models MPNN+LSTM and ATMGNN maintained a stable performance decay profile with a low decay rate on both error and correlation compared to every other model aside from the LSTM exception, alongside lower values in both metrics across the board. Both temporal graph models started with relatively high performance in terms of all metrics compared to other baselines and mostly maintained the same performance when predicting longer time ranges with minimal decay, resulting in them outperforming all other baseline models. We specifically demonstrated the relative metric and decay stability of the two graph-based temporal models by averaging over several runs and computing the deviation, showing the performance and decay similarities between these models over time.

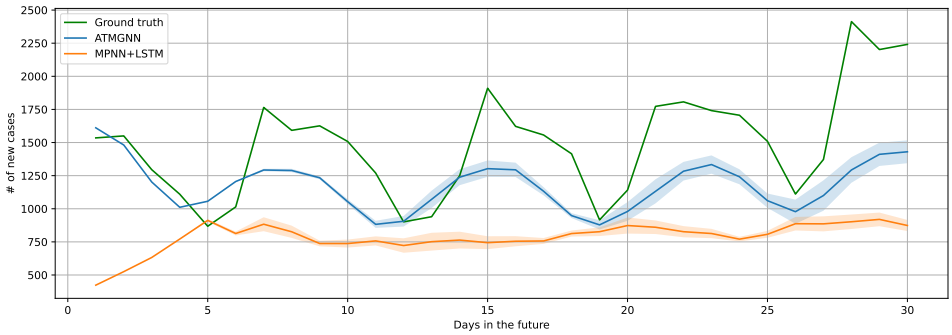

Figure 3: Sample out-of-distribution new case predictions for two graph-based models.

Out-of-distribution forecasting We examined the out-of-distribution performances of two of our best-performing models, the MPNN+LSTM and ATMGNN. The evaluation is done on the number of new cases between September 4, 2022 and November 4, 2022, with no overlapping between the evaluation set and the train/validation sets. All models are evaluated as autoregressive models,

meaning for the 30-day prediction window the models use the prediction output of the previous day as an input feature for predicting the current day. As demonstrated in Figure 3, ATMGNN outperformed MPNN+LSTM when it comes to prediction error and emulating the spiking dynamics of the number of new cases. The predictions retrieved from the outputs of ATMGNN showed that the model can fairly precisely simulate the case spiking dynamics even when tested on a dataset completely unrelated and separate from the training dataset, demonstrating good generalization performance of the model. All other baseline models are not included in the evaluation after extensive testing showing that their performances are not remotely comparable to the two models demonstrated above. Further testing with different information windows with ground truth case information feed into the model ranging between 3 and 9 days before the target day to predict showed that both models maintained similar forecasting patterns with ATMGNN better conforming to the ground truth and the models maintaining relatively stable predictions given different case information levels.

5 Discussion

Interpretation of the main results We provided a comprehensive evaluation of four classes of COVID-19 forecasting models, with a detailed analysis of the models' performance, decay over time, out-of-distribution forecasting, and economic features addition. Generally, graph neural network-based models, specifically the temporal variant of graph-based models outperformed every other baseline model in terms of performance metrics and performance decay over time. This trend is not shared by non-temporal graph-based models, indicating the importance of temporal mechanisms in forecasting models, whether it is attention-based on recurrent network-based (i.e., LSTM). The results suggest that the spatiotemporal approach to modeling the spread of COVID-19 based on the number of new cases is effective compared to other traditional modeling methods. Intuitively, graph-based models can accurately simulate the change in the number of new cases in one region when given that region's traffic connectivity with its neighbors. Since the spread of COVID-19 in every country, not only in New Zealand, is movement-based in nature, by modeling such geographical connectivity we can find latent information by accounting for human contacts with graph-based models. Moreover, the out-of-distribution performance of multiresolution temporal graph models also demonstrates the utility of modeling the problem of COVID-19 forecasting as a hierarchical system, with spreads localized in adjacent regions that have significant traffic volume.

Limitations and future directions While certain metrics of the proposed models are satisfactory, we have identified several weaknesses of the models that were tested. Graph-based models, while powerful, still require a certain amount of computational resources and adequate time for the process of training the models. Data inputs also have to be well-structured and preprocessed carefully to suit the formatting of the models, though this is less of a concern given the availability and accuracy of case datasets such as the New Zealand COVID-19 public dataset. Furthermore, data features can be further enriched with more detailed movement data between regions, traffic density information for all traveling modalities (e.g., land, sea, or air travel), and local movement details within each region. Most importantly, graph-based networks and deep learning models in general are black-boxes, offering little insight into the precise mechanisms of forecasting and modeling disease dynamics for the sake of studying the exact nature of epidemic spread. Further research may focus on additional data features and enrichment features readily incorporable into the model as node features (e.g., more fine-grained socioeconomic features) or as edge features (e.g., mobility data), or interpretation methods designed for graph neural networks [32] for the sake of understanding the inner workings of such prediction models.

6 Conclusion

In this paper, we propose a combined multiresolution graph neural network model for forecasting evolving COVID-19 pandemic graphs constructed from New Zealand data. We also provided a comprehensive evaluation of the model's performance with respect to high-performing traditional and machine learning approaches to the task of epidemic forecasting. Our study suggested that graph neural network-based models outperformed every other baseline model in terms of performance metrics and performance decay over time. Furthermore, our graph neural network-based models can effectively predict the number of COVID-19 cases up to 30 days and therefore can assist with public health policy planning in order to control the COVID-19 outbreaks. Our work can be further expanded

with richer graph features such as mobility between infected regions or policy representations. Finally, our results in terms of model structures and frameworks can be generalized to other countries with similar settings.

References

[1] NZStats: new zealand's official data agency. https://www.stats.govt.nz/. Accessed: 2022-11-30.

[2] K. E. ArunKumar, D. V. Kalaga, C. M. Sai Kumar, G. Chilkoor, M. Kawaji, and T. M. Brenza. Forecasting the dynamics of cumulative COVID-19 cases (confirmed, recovered and deaths) for top-16 countries using statistical machine learning models: Auto-Regressive integrated moving average (ARIMA) and seasonal Auto-Regressive integrated moving average (SARIMA). *Appl. Soft Comput.*, 103(107161):107161, May 2021.

[3] P. Battaglia, R. Pascanu, M. Lai, D. Jimenez Rezende, and k. kavukcuoglu. Interaction networks for learning about objects, relations and physics. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/file/3147da8ab4a0437c15ef51a5cc7f2dc4-Paper.pdf.

[4] G. Battineni, N. Chintalapudi, and F. Amenta. Forecasting of covid-19 epidemic size in four high hitting nations (usa, brazil, india and russia) by fb-prophet machine learning model. *Applied Computing and Informatics*, 2020.

[5] R. Chandra, A. Jain, and D. Singh Chauhan. Deep learning via lstm models for covid-19 infection forecasting in india. *PLOS ONE*, 17(1):1–28, 01 2022. doi: 10.1371/journal.pone.0262708. URL https://doi.org/10.1371/journal.pone.0262708.

[6] L. Cuesta-Herrera, L. Pastenes, F. Córdova-Lepe, A. D. Arencibia, H. Torres-Mantilla, and J. P. Gutiérrez-Jara. Analysis of seir-type models used at the beginning of covid-19 pandemic reported in high-impact journals. *Medwave*, 2022.

[7] Z. Diao, X. Wang, D. Zhang, Y. Liu, K. Xie, and S. He. Dynamic spatial-temporal graph convolutional neural networks for traffic forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):890–897, Jul. 2019. doi: 10.1609/aaai.v33i01.3301890. URL https://ojs.aaai.org/index.php/AAAI/article/view/3877.

[8] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/f9be311e65d81a9ad8150a60844bb94c-Paper.pdf.

[9] Z.-G. Fang, S.-Q. Yang, C.-X. Lv, S.-Y. An, and W. Wu. Application of a data-driven XGBoost model for the prediction of COVID-19 in the USA: a time-series study. *BMJ Open*, 12(7): e056685, July 2022.

[10] M. Fey and J. E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[11] A. Fout, J. Byrd, B. Shariat, and A. Ben-Hur. Protein interface prediction using graph convolutional networks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6533–6542, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[12] J. Galasso, D. M. Cao, and R. Hochberg. A random forest model for forecasting regional COVID-19 cases utilizing reproduction number estimates and demographic data. *Chaos Solitons Fractals*, 156(111779):111779, Mar. 2022.

[13] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1263–1272. PMLR, 06–11 Aug 2017. URL `https://proceedings.mlr.press/v70/gilmer17a.html`.

[14] E. J. Gumbel. Statistical theory of extreme values and some practical applications: a series of lectures. *US Govt. Print. Office*, Number 33, 1954.

[15] A. A. Hagberg, D. A. Schult, and P. J. Swart. Exploring network structure, dynamics, and function using networkx. In G. Varoquaux, T. Vaught, and J. Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.

[16] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, Sept. 2020. doi: 10.1038/s41586-020-2649-2. URL `https://doi.org/10.1038/s41586-020-2649-2`.

[17] S. Hendy, N. Steyn, A. James, M. J. Plank, K. Hannah, R. N. Binny, and A. Lustig. Mathematical modelling to inform new zealand's covid-19 response. *Journal of the Royal Society of New Zealand*, 51(sup1):S86–S106, 2021. doi: 10.1080/03036758.2021.1876111. URL `https://doi.org/10.1080/03036758.2021.1876111`.

[18] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, nov 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL `https://doi.org/10.1162/neco.1997.9.8.1735`.

[19] T. S. Hy and R. Kondor. Multiresolution equivariant graph variational autoencoder, 2021. URL `https://arxiv.org/abs/2106.00967`.

[20] T. S. Hy and R. Kondor. Multiresolution equivariant graph variational autoencoder. *Machine Learning: Science and Technology*, 4(1):015031, mar 2023. doi: 10.1088/2632-2153/acc0d8. URL `https://dx.doi.org/10.1088/2632-2153/acc0d8`.

[21] T. S. Hy, S. Trivedi, H. Pan, B. M. Anderson, , and R. Kondor. Predicting molecular properties with covariant compositional networks. *The Journal of Chemical Physics*, 148, 2018.

[22] T. S. Hy, S. Trivedi, H. Pan, B. M. Anderson, and R. Kondor. Covariant compositional networks for learning graphs. In *Proc. International Workshop on Mining and Learning with Graphs (MLG)*, 2019.

[23] T. S. Hy, V. B. Nguyen, L. Tran-Thanh, and R. Kondor. Temporal multiresolution graph neural networks for epidemic prediction. In P. Xu, T. Zhu, P. Zhu, D. A. Clifton, D. Belgrave, and Y. Zhang, editors, *Proceedings of the 1st Workshop on Healthcare AI and COVID-19, ICML 2022*, volume 184 of *Proceedings of Machine Learning Research*, pages 21–32. PMLR, 22 Jul 2022. URL `https://proceedings.mlr.press/v184/hy22a.html`.

[24] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. In *ICLR*, 2017. URL `https://arxiv.org/abs/1611.01144`.

[25] G. Kaur, P. Kaur, N. Kaur, and P. Kaur. Forecasting prediction of covid-19 outbreak using linear regression. In I. J. Jacob, S. Kolandapalayam Shanmugam, and I. Izonin, editors, *Data Intelligence and Cognitive Informatics*, pages 195–221, Singapore, 2023. Springer Nature Singapore. ISBN 978-981-19-6004-8.

[26] S. Kearnes, K. McCloskey, M. Berndl, V. Pande, and P. Riley. Molecular graph convolutions: moving beyond fingerprints. *Journal of Computer-Aided Molecular Design*, 30(8):595–608, Aug 2016. ISSN 1573-4951. doi: 10.1007/s10822-016-9938-8.

[27] S. Ketu and P. K. Mishra. Enhanced gaussian process regression-based forecasting model for COVID-19 outbreak and significance of IoT for its detection. *Appl. Intell.*, 51(3):1492–1512, 2021.

[28] T. Kufel. Arima-based forecasting of the dynamics of confirmed covid-19 cases for selected european countries. *Equilibrium. Quarterly Journal of Economics and Economic Policy*, 15 (2):181–204, 2020. URL https://EconPapers.repec.org/RePEc:pes:ierequ:v:15:y:2020:i:2:p:181-204.

[29] N. Kumar and S. Susan. COVID-19 pandemic prediction using time series forecasting models. In *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. IEEE, jul 2020. doi: 10.1109/icccnt49239.2020.9225319. URL https://doi.org/10.1109%2Ficccnt49239.2020.9225319.

[30] Y. Li, R. Zemel, M. Brockschmidt, and D. Tarlow. Gated graph sequence neural networks. In *Proceedings of ICLR'16*, April 2016.

[31] Y. Li, R. Yu, C. Shahabi, and Y. Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SJiHXGWAZ.

[32] Y. Li, J. Zhou, S. Verma, and F. Chen. A survey of explainable graph neural networks: Taxonomy and evaluation metrics, 2022.

[33] B. Lucas, B. Vahedi, and M. Karimzadeh. A spatiotemporal machine learning approach to forecasting covid-19 incidence at the county level in the usa. *International Journal of Data Science and Analytics*, 2022. doi: 10.1007/s41060-021-00295-9. URL https://doi.org/10.1007/s41060-021-00295-9.

[34] J. Luo, Z. Zhang, Y. Fu, and F. Rao. Time series prediction of covid-19 transmission in america using lstm and xgboost algorithms. *Results in Physics*, 27:104462 – 104462, 2021.

[35] C. J. Maddison, D. Tarlow, and T. Minka. A* sampling. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[36] S. Mahmud. Bangladesh covid-19 daily cases time series analysis using facebook prophet model. *PSN: Disease & Illness (Topic)*, 2020. URL https://ssrn.com/abstract=3660368.

[37] D. T. Nguyen, M. D. T. Nguyen, T. S. Hy, and R. Kondor. Fast temporal wavelet graph neural networks. *arXiv preprint arXiv:2302.08643*, 2023.

[38] H. R. Niazkar and M. Niazkar. Application of artificial neural networks to predict the covid-19 outbreak. *Global Health Research and Policy*, 5, 2020.

[39] M. Niepert, M. Ahmed, and K. Kutzkov. Learning convolutional neural networks for graphs. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2014–2023, New York, New York, USA, 20–22 Jun 2016. PMLR.

[40] B. Nikparvar, M. M. Rahman, F. Hatami, and J.-C. Thill. Spatio-temporal prediction of the covid-19 pandemic in us counties: modeling with a deep lstm neural network. *Scientific Reports*, 11, 2021.

[41] G. Panagopoulos, G. Nikolentzos, and M. Vazirgiannis. Transfer graph neural networks for pandemic forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(6): 4838–4845, May 2021. doi: 10.1609/aaai.v35i6.16616. URL https://ojs.aaai.org/index.php/AAAI/article/view/16616.

[42] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.

[43] R. C. Poonia, A. K. J. Saudagar, A. Altameem, M. Alkhathami, M. B. Khan, and M. H. A. Hasanat. An enhanced SEIR model for prediction of COVID-19 with vaccination effect. *Life (Basel)*, 12(5):647, Apr. 2022.

[44] W. C. Roda, M. B. Varughese, D. Han, and M. Y. Li. Why is it difficult to accurately predict the COVID-19 epidemic? *Infect. Dis. Model.*, 5:271–281, Mar. 2020.

[45] R. Rodiah, E. Patriya, D. T. Susetianingtias, and E. Sutanty. Implementation of the prophet model in covid-19 cases forecast. *ILKOM Jurnal Ilmiah*, 14(2):99–111, 2022. ISSN 2548-7779. doi: 10.33096/ilkom.v14i2.1219.99-111. URL `https://jurnal.fikom.umi.ac.id/index.php/ILKOM/article/view/1219`.

[46] E. Rossi, B. Chamberlain, F. Frasca, D. Eynard, F. Monti, and M. Bronstein. Temporal graph networks for deep learning on dynamic graphs. *arXiv preprint arXiv:2006.10637*, 2020.

[47] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009. doi: 10.1109/TNN.2008.2005605.

[48] S. Shastri, K. Singh, S. Kumar, P. Kour, and V. Mansotra. Time series forecasting of covid-19 using deep learning models: India-usa comparative case study. *Chaos, Solitons, and Fractals*, 140:110227 – 110227, 2020.

[49] R. C. Staudemeyer and E. R. Morris. Understanding lstm – a tutorial into long short-term memory recurrent neural networks, 2019. URL `https://arxiv.org/abs/1909.09586`.

[50] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf`.

[51] Y. Y. Wei, Z. Z. Lu, Z. C. Du, Z. J. Zhang, Y. Zhao, S. P. Shen, B. Wang, Y. T. Hao, and F. Chen. Fitting and forecasting the trend of COVID-19 by SEIR(+CAQ) dynamic model. *Zhonghua Liu Xing Bing Xue Za Zhi*, 41(4):470–475, Apr. 2020.

[52] Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61, 2010. doi: 10.25080/Majora-92bf1922-00a.

[53] M. Yousaf, S. Zahir, M. Riaz, S. M. Hussain, and K. Shah. Statistical analysis of forecasting covid-19 for upcoming month in pakistan. *Chaos, Solitons & Fractals*, 138:109926, 2020. ISSN 0960-0779. doi: https://doi.org/10.1016/j.chaos.2020.109926. URL `https://www.sciencedirect.com/science/article/pii/S0960077920303258`.

[54] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020. ISSN 2666-6510. doi: https://doi.org/10.1016/j.aiopen.2021.01.001. URL `https://www.sciencedirect.com/science/article/pii/S2666651021000012`.

Appendix

Model	Next 3 Days			Next 7 Days			Next 14 Days			Next 21 Days		
	MAE	RMSE	R^2	MAE	RMSE	R^2	MAE	RMSE	R^2	MAE	RMSE	R^2
AVG	8.15	11.39	-0.14	8.50	11.77	-0.35	8.97	12.14	-0.80	9.32	12.60	-1.39
AVG_WINDOW	6.33	8.79	0.40	7.94	10.87	-0.07	11.04	14.91	-1.52	14.17	18.77	-4.06
LAST_DAY	7.12	10.45	0.19	7.33	10.49	0.19	9.83	14.13	-0.90	12.76	17.85	-3.01
LIN_REG	13.13	17.77	-0.32	17.19	22.95	-1.32	26.28	34.53	-6.34	37.11	47.94	-18.85
GP_REG	14.95	21.48	-0.93	14.13	20.65	-0.88	12.00	17.50	-0.89	10.04	14.72	-0.87
RAND_FOREST	6.64	9.87	0.59	7.16	10.17	0.54	10.01	14.08	-0.22	13.03	17.88	-1.76
XGBOOST	7.32	10.93	0.50	7.46	10.82	0.48	10.00	14.50	-0.30	12.82	18.06	-1.82
PROPHET	10.79	20.78	-0.06	14.45	29.28	-1.29	23.43	34.29	-2.99	33.59	31.72	-3.66
ARIMA	8.95	20.28	-0.01	9.51	13.77	0.49	9.63	13.02	0.43	9.77	11.62	0.37
LSTM	8.61	11.88	-0.36	8.20	11.24	0.12	7.86	10.66	0.47	7.09	9.95	0.65
MPNN	6.51	9.41	0.55	7.54	10.63	0.39	10.12	14.14	-0.42	12.84	17.77	-1.82
MGNN	6.87	9.48	0.55	8.18	10.93	0.35	11.07	14.67	-0.60	14.14	18.67	-2.16
MPNN+LSTM	6.73	9.55	0.57	7.08	10.13	0.57	7.68	10.89	0.57	7.95	11.36	0.58
ATMGNN	**6.24**	**8.82**	**0.63**	**6.44**	**9.04**	**0.66**	**6.80**	**9.57**	**0.68**	**6.70**	**9.53**	**0.73**

Table 2: England: Performance of all experimental model evaluated based on the metrics specified in Section 4.4

Model	Next 3 Days			Next 7 Days			Next 14 Days			Next 21 Days		
	MAE	RMSE	R^2	MAE	RMSE	R^2	MAE	RMSE	R^2	MAE	RMSE	R^2
AVG	21.13	42.80	0.53	20.31	41.88	0.49	20.28	43.23	0.43	**19.19**	41.35	0.39
AVG_WINDOW	**17.69**	**33.48**	**0.66**	19.75	**37.30**	0.53	23.75	44.90	0.30	26.88	50.00	-0.01
LAST_DAY	21.21	41.99	0.45	21.83	43.36	0.37	25.45	49.97	0.13	27.73	50.85	0.03
LIN_REG	28.15	54.53	0.29	35.35	69.38	-0.29	50.02	99.11	-1.74	65.12	132.16	-4.61
GP_REG	37.41	74.72	-0.33	35.72	70.72	-0.34	33.12	68.38	-0.31	30.13	63.40	-0.29
RAND_FOREST	19.42	40.80	0.60	20.41	42.38	0.52	24.80	50.01	0.30	27.41	52.76	0.11
XGBOOST	21.79	47.09	0.47	22.49	48.63	0.37	26.41	55.60	0.14	28.64	57.06	-0.05
PROPHET	23.03	55.65	0.46	29.18	71.44	0.15	40.95	92.95	-0.76	51.92	100.06	-1.64
ARIMA	22.90	76.56	-0.03	27.01	68.94	0.21	28.62	56.27	0.36	25.57	45.51	0.45
LSTM	20.98	42.01	0.51	19.80	40.17	0.59	**19.56**	39.91	0.60	20.18	39.44	0.66
MPNN	18.09	36.67	0.64	21.45	43.56	0.49	26.07	51.72	0.21	28.94	59.26	-0.16
MGNN	19.14	37.17	0.64	22.69	42.99	0.51	27.33	51.73	0.20	29.87	58.18	-0.14
MPNN+LSTM	18.50	38.43	0.60	**19.48**	39.98	0.59	19.72	41.89	0.56	19.84	41.22	0.58
ATMGNN	18.05	36.94	0.65	19.63	39.06	**0.63**	19.80	**39.49**	**0.63**	18.55	**37.07**	**0.67**

Table 3: Italy: Performance of all experimental model evaluated based on the metrics specified in Section 4.4

Model	Next 3 Days			Next 7 Days			Next 14 Days			Next 21 Days		
	MAE	RMSE	R^2	MAE	RMSE	R^2	MAE	RMSE	R^2	MAE	RMSE	R^2
AVG	7.65	14.41	-462.73	7.55	14.22	-591.09	7.92	15.20	-872.30	8.49	16.56	-1187.11
AVG_WINDOW	**5.24**	**9.47**	-102.43	5.69	10.14	-215.84	7.88	14.55	-666.37	10.56	19.55	-2235.75
LAST_DAY	7.29	13.94	-140.97	5.05	**9.84**	-181.85	7.12	14.01	-970.65	10.05	19.45	-6024.04
LIN_REG	9.89	19.71	-0.55	12.59	24.65	-3.63	18.57	35.83	-21.64	26.00	49.31	-101.23
GP_REG	8.20	17.55	-0.22	7.51	13.45	-0.10	8.97	18.67	-5.34	9.79	21.43	-18.70
RAND_FOREST	6.91	16.15	-0.04	5.01	10.94	0.09	7.13	15.60	-3.29	9.76	20.83	-17.25
XGBOOST	7.80	18.15	-0.31	**5.00**	12.56	-0.20	**7.03**	17.15	-4.19	9.71	22.87	-20.98
PROPHET	11.12	42.21	-1.41	13.86	44.01	-2.13	21.25	40.86	-3.10	27.88	39.83	-8.04
ARIMA	9.09	19.95	**0.46**	9.08	20.54	**0.32**	8.78	16.03	**0.37**	**8.13**	**13.37**	**-0.03**
LSTM	7.95	14.98	-590.28	6.12	11.62	-41.90	7.89	14.42	-29.37	8.93	16.34	-35.69
MPNN	6.41	12.14	-28.73	5.61	10.70	-41.77	8.11	15.09	-254.37	10.85	20.89	-1358.29
MGNN	7.04	11.83	-27.92	7.44	11.66	-69.09	10.17	16.27	-424.27	13.21	22.09	-1363.33
MPNN+LSTM	6.92	12.72	-19.68	7.55	13.71	-30.45	7.46	13.10	-6.30	8.26	15.20	-2.66
ATMGNN	7.44	13.21	-13.40	7.16	12.59	-27.34	7.28	**12.69**	-9.79	8.25	14.18	-1.61

Table 4: France: Performance of all experimental model evaluated based on the metrics specified in Section 2.5

Model	Next 3 Days			Next 7 Days			Next 14 Days			Next 21 Days		
	MAE	RMSE	R^2	MAE	RMSE	R^2	MAE	RMSE	R^2	MAE	RMSE	R^2
AVG	48.71	111.19	-9.68	52.60	122.82	-12.68	60.01	149.53	-20.01	68.19	178.26	-31.68
AVG_WINDOW	**32.56**	**59.57**	-0.51	40.09	79.83	-3.08	53.03	121.30	-12.42	63.15	159.29	-25.10
LAST_DAY	35.20	63.98	-0.34	37.63	70.57	-1.23	52.60	112.27	-8.45	63.23	155.97	-22.17
LIN_REG	50.73	104.21	0.11	62.34	126.85	-0.39	87.19	190.79	-5.48	120.59	267.01	-29.84
GP_REG	53.22	121.87	-0.22	51.28	119.04	-0.23	43.12	86.49	-0.33	**31.58**	**57.52**	-0.43
RAND_FOREST	33.27	65.77	**0.64**	37.05	74.41	**0.52**	51.72	117.37	-1.45	61.38	155.99	-9.53
XGBOOST	35.41	69.85	0.60	38.18	76.67	0.49	52.58	117.90	-1.47	62.70	159.38	-9.99
PROPHET	60.60	351.20	-2.49	75.86	320.15	-2.88	114.87	192.33	-0.91	149.51	167.25	-1.09
ARIMA	41.89	112.37	**0.64**	40.54	75.95	0.78	48.46	82.08	0.65	56.45	119.47	-0.18
LSTM	43.39	84.78	-3.69	44.25	85.54	-3.48	36.50	**66.01**	0.33	35.74	60.87	**0.76**
MPNN	33.26	65.51	0.24	39.91	82.70	-0.70	50.42	106.45	-4.64	61.95	142.19	-9.40
MGNN	34.71	66.16	0.26	42.35	84.48	-0.68	54.31	109.22	-5.09	67.11	148.25	-10.40
MPNN+LSTM	34.60	69.55	0.37	**35.03**	**67.52**	0.45	37.95	81.06	0.56	39.68	82.58	0.72
ATMGNN	34.08	69.37	0.40	35.88	72.06	0.40	**34.73**	69.56	**0.70**	38.13	82.72	0.71

Table 5: Spain: Performance of all experimental model evaluated based on the metrics specified in Section 2.5

Model	Next 3 Days			Next 7 Days			Next 14 Days			Next 21 Days		
	MAE	RMSE	R^2	MAE	RMSE	R^2	MAE	RMSE	R^2	MAE	RMSE	R^2
MPNN+LSTM	128.79	139.09	-3.65	149.44	160.53	-2.73	174.22	185.44	-2.81	203.13	214.37	-3.17
ATMGNN	132.83	143.38	-4.04	150.52	161.47	-2.77	178.19	189.93	-2.88	208.49	220.03	-3.20
MPNN+LSTM (ECON)	128.17	139.26	-3.22	150.13	161.30	-3.00	165.94	178.23	-2.62	199.19	211.00	-3.17
ATMGNN (ECON)	131.59	142.21	-4.23	154.19	165.13	-3.25	173.04	184.48	-2.71	203.36	214.87	-3.16

Table 6: New Zealand: Limited DHB map results for graph-based models with and without economic features

Model	Next 3 Days			Next 7 Days			Next 14 Days			Next 21 Days		
	MAE	RMSE	R^2	MAE	RMSE	R^2	MAE	RMSE	R^2	MAE	RMSE	R^2
MPNN+LSTM (MO)	86.85	119.97	0.84	99.89	137.90	0.81	114.84	158.12	0.75	128.79	177.60	0.70
ATMGNN (MO)	85.67	118.27	0.84	100.91	138.11	0.81	114.43	157.87	0.76	129.89	178.97	0.70
MPNN+LSTM (MO+ET)	109.82	150.23	0.76	121.45	166.33	0.73	124.55	171.37	0.71	136.98	188.01	0.66
ATMGNN (MO+ET)	104.44	143.88	0.76	113.65	155.83	0.76	128.81	177.27	0.70	133.34	183.91	0.68

Table 7: New Zealand: Multiple outputs demographical graph models results, with and without explicit training weighting. MO denotes the multiple output graph models, and ET denotes custom age group weighting during graph model training.