GraphCoReg: Co-Training for Regression on Temporal Graphs

Harshith Mohan Kumar[1], Vishruth Veerendranath[1], Vibha Masti[1], Divya Shekar[1], and Bhaskarjyothi Das[1]

Department of Computer Science and Engineering,
PES University, Bengaluru, Karnataka, India
{harshithmohankumar,vishruth,vibha,divya}@pesu.pes.edu,
bhaskarjyoti01@gmail.com

Abstract. Graphs are widely in use to model related instances of data attributed with properties providing rich spatial information. Although a lot of classical graph-related problems have been solved with the advent of Graph Neural Networks (GNN), Spatio-Temporal data poses a new challenge. We propose GraphCoReg: a novel methodology to perform regression on spatio-temporal data, in a Semi-Supervised Learning (SSL) setting using co-training. Our co-training approach exploits two model-based views of the dataset using two temporal Graph Neural Networks (GNNs) - an Attention-based GNN (ASTGCN) and a Long Short Term Memory GNN (GCLSTM). Additionally, methodologies to incrementally add the pseudo-targets to training data have been described. We finally compare the performance of the semi-supervised model with equivalent supervised models. This approach has been tested on the MetrLA dataset for traffic forecasting. This is a work-in-progress to investigate the performance of GraphCoReg on multiple benchmark spatio-temporal datasets for the task of regression on temporal graphs.

Keywords: Co-training · Temporal Graphs · Dynamic Graphs · Semi-Supervised Learning · Graph Neural Network · Traffic Forecasting · Pseudo-values · Regression

1 Introduction

A graph, in its most basic sense, is a collection of nodes and the edges that connect them. Graphs are abundantly useful in representing collections of data, such as social networks, wireless sensor networks, maps, molecules, etc. In recent years, there has been an increase in the usage of graph-based learning methods using the *message passing* framework in variations of *Graph Neural Networks* [18] [27] to solve the problems of node classification, edge prediction and graph classification.

Dynamic graphs are graphs whose components such as node attributes, links, and the number of nodes change over time. Such graphs can be viewed as a time-series of different graph snapshots. One class of dynamic graphs, as defined

in *PyTorch Geometric Temporal* [17], consists of *Static Graphs with Temporal Signals*. Their graph structure (nodes and edges) remains invariant, but the signals (node and edge attributes) change with time. The temporal signals can be viewed as state transitions, hence providing a time-series of "states", similar to a standard time-series problem.

Additionally, in real-world spatio-temporal data, true values might be scarcely available due to issues like sensor failures within the network. Performing regression with limited ground truth values in a Semi-Supervised setting has previously been a challenge. In this case, we posit that the spatial information of these graphs can be exploited to handle the unavailability of ground truth values.

To handle this problem of regression in a Semi-Supervised Learning (SSL) setting that exploits spatial and temporal information, we propose a novel methodology *GraphCoReg*. We use co-training [1] that exploits two views of the graph generated by two temporal GNNs — namely the Graph Convolution Embedded LSTM (GCLSTM) [2] and the Attention Spatial-Temporal Graph Convolutional Network (ASTGCN) [6] — to efficiently generate pseudo-targets. These pseudo-targets are incrementally added to the data that is used to train the models to learn an optimal regression function. We also propose a novel method to pick the most *confident pseudo-targets*.

In this study, we explore the applications of *GraphCoReg* for the problem of traffic forecasting, using the *MetrLA* benchmark dataset [10] and compare its performance to the two individual models in a supervised manner. The rest of the paper is organized as follows: Section 2 summarizes related work, Section 3 elaborates on the experimental setting, Section 4 details the methodology, Section 5 discusses the results and analysis, and Section 6 covers the conclusion and future scope.

2 Related Work

2.1 Co-training and Semi-Supervised Learning

SSL algorithms refer to a class of machine learning techniques that exploit the availability of an abundance of unlabelled data to enhance the learning capability of labelled data. Co-training [1] as an SSL technique has been around for over two decades, but several recent improvements have been made. Want et al. [22] proposed random subspace co-training (RASCO) which eased up the necessary condition of multi-view learning by artificially generating it via a sufficiently large random division of the attribute set. Han et al. [8] propose co-teaching, which aims to curb the effect of noisy labels by co-training with two deep neural networks that "teach" each other. With exponential developments in deep learning algorithms, researchers have explored fusing deep learning with co-training [15].

2.2 Semi-Supervised Regression

Semi-Supervised Regression (SSR) is a statistical technique that generates real-valued output vectors whereas semi-supervised classification generates discrete

output variables [12]. A majority of the research output in the field of SSL pertains to classification. In a review conducted by Ning et al. [15], SSL classification held 9% of the proportion of keywords related to SSL. Regression on the other hand had less than 1%. Zhou and Li [28] sparked a new interest in SSL regression with their co-training style semi-supervised regression algorithm named COREG. In our paper, we greatly reduce the complexity of the neighbour-based confidence estimator proposed in COREG while retaining the accuracy of our model.

2.3 Graph-based Semi-Supervised Learning

In [3], the authors describe both transductive and inductive algorithms for graph-based SSL. Inductive learning is mainly concerned with methods related to the *Label Propagation Algorithm* [29], either based on embeddings or neural networks, whereas transductive learning consists of *sparse*, *low-rank* models and semi-supervised *neural network* models. Wang et. al. [23] provided an intuitive perspective of looking at co-training and multi-view learning as a label propagation over a combinative graph. Extending this idea, [13] introduces *Co-GCN* which provides a new aggregation strategy for Graph Convolution Networks (GCN) that is trained on a combinative model of two views. *Mutual Teaching* [25] describes a more conventional co-training approach where pseudo-labels are generated from the two views. The authors also introduce a *pseudo-label loss* and *consistency loss* to encourage consistency among the two views' embeddings. Similarly in [11], spectral Graph Convolutions have been used for semi-supervised node classification.

3 Experimental Setting

To simulate a semi-supervised setting, we assume a hypothetical scenario where a number of these speed detectors have failed in the sensor network. This would lead to the unavailability of true targets at all failed nodes, and the success of conventional supervised and time-series methods in such a scenario is questionable due to their inability to circumvent missing data.

A detailed description of the MetrLA dataset [10] used has been provided under Appendix C. Formally, we refer to the set of unlabelled/failed nodes as set U. U is also referred to as the set of *masked nodes* as their signals have been masked across all time-steps. The nodes in set U are picked at random with no bias towards a node's neighbourhood. The remaining set of nodes (*labelled or non-masked nodes*) are referred to as set L. The ratio of $U/(L+U)$ or *mask-ratio* is a hyperparameter and has been chosen to be 0.7, which would mean true-targets are unavailable at 70% of the nodes. This choice of hyperparameter simulates the relevant realistic scenario of the high availability of unlabelled data coupled with a small pool of labelled data. Drawing a parallel to the MetrLA dataset, it allows for a high margin of speed detector failures in the sensor network.

4 Methodology

4.1 Compatibility with multi-view co-training

An extension of the original co-training algorithm [1] proposed by Goldman and Zhou [5], demonstrates that the requirement of sufficient and redundant views can be substituted through the use of a largely diverse base of learning algorithms. The GC-LSTM offers a long-term dependency view while the AST-GCN offers an attention-based weighted view. The two base learners are conditionally independent owing to their difference in architecture and underlying algorithms, a detailed description of which has been provided in Appendix B. The GC-LSTM model was designed with the intention for link prediction whereas the AST-GCN has been built for node prediction. Although our primary task pertains to node label prediction, we assume that the spatio-temporal features from a link prediction GNN would improve that of a node prediction GNN. In combination, the two independent views allow our model to train on and leverage heterogeneous input features owing to each model in turn learning from the other. Since the two base learners are conditionally independent, it would imply that their target outputs are also conditionally independent, demonstrated in Eq. 1.

$$(G_{lstm} \perp\!\!\!\perp G_{ast}) \Rightarrow (\hat{y}_{lstm} \perp\!\!\!\perp \hat{y}_{ast}) \tag{1}$$

The architecture of the resultant overall co-training model is similar to the AGC-LSTM classification framework proposed by [20] in a semi-supervised setting for regression. Thus we propose a novel approach for semi-supervised regression on spatio-temporal graphs using multi-view co-training.

4.2 GraphCoReg

The two temporal graph neural networks, i.e G_{lstm} and G_{ast}, are initially trained on the labelled data L and later refined with the benefit of pseudo-targets P generated from the unlabelled example set U. The pseudo-targets P are computed by averaging the values predicted by the different models G_{lstm} and G_{ast}. This enables the inclusion of implicit information extracted by both views in the pseudo-targets. The selection of the set of nodes (top_k) whose pseudo-targets are added to training data Y at the end of every epoch has been described further in Section 4.3. Over each epoch, the pool of labelled data is incremented until eventually GraphCoReg exhausts the U set and continues to train on the set $L+P$. The base models are backpropagated during every time-step. The loss is calculated using the *mean squared error* shown in Eq. 2. The GraphCoReg training algorithm in its entirety is provided in Algorithm 1 and is explained in Fig. 1. We have included a link to the GitHub Repository [1].

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{2}$$

[1] https://github.com/Deep-Co-Training/GraphCoReg

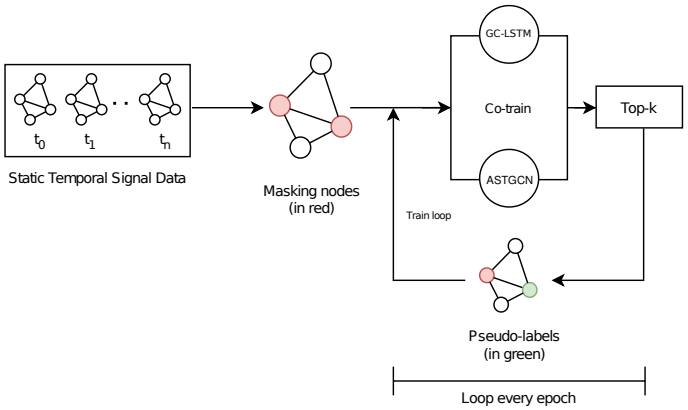

Fig. 1. CoTraining

4.3 Pseudo-target Confidence Estimation

Co-training relies on the ability to choose the top k candidates from set U to add to set L based on the confidence of their predictions. When performing co-training on classification, the top k targets are typically chosen corresponding to the k highest softmax values. In the case of regression, however, the possible values have no bounds and a different method must be employed. Zhou et al. [28] use two kNNs with different distance metrics as the two models and calculate the top k based on the influence of each prediction on others.

We introduce a method to calculate the top k candidates by finding the k nodes in the graph with the least difference in the two predicted values from the two models. Every epoch, k nodes from U get added to L. If at any point k exceeds $\|U\|$, we simply add all nodes in U to L. The algorithm is described in Algorithm 2.

5 Results and Analysis

Our results have been obtained by training the models using an RTX Nvidia 3080 GPU with PyTorch-GPU dependencies. The data was sourced and the models were built using PyTorch Geometric Temporal [17]. The models were trained on 80% of the graph dataset which approximately took 9 hours to complete.

5.1 Co-training vs Supervised

For purposes of equal and fair comparison between the two learning methods, the losses of co-training and the supervised algorithms were calculated on only the unmasked nodes, i.e the nodes which don't have a "sensor failure". We computed and recorded the sum of mean-squared error across all time-steps during each epoch. The results from training for 30 epochs are shown in Fig. 3.

Algorithm 1 GraphCoReg Training

Input: Two Temporal GNNs - the GCLSTM G_{lstm} and astGCN G_{ast} , Set of labelled nodes L and unlabelled nodes U, Temporal Signals X, Targets Y, Number of pseudo-targets to append each epoch k, Number of epochs E

for epoch $e = 0$ to E **do**
  for timestep $t = 0$ to T **do**
    $\hat{y}^t_{lstm} \leftarrow \mathrm{G}_{lstm}(x^t)$; $\hat{y}^t_{ast} \leftarrow \mathrm{G}_{ast}(x^t)$
    $loss_L \leftarrow \mathrm{MSE}(\hat{y}^t_L, y^t_L)$ ▷ $\hat{y}^t_L$ are predictions of nodes in L
    Backpropogate $loss_L$ ▷ Tune the weights of G_{lstm} and G_{ast}
  end for
  $\hat{Y}_{lstm} \leftarrow \mathrm{G}_{lstm}(X)$; $\hat{Y}_{ast} \leftarrow \mathrm{G}_{ast}(X)$
  k $\leftarrow$ min(k,$\|U\|$)
  top_k $\leftarrow$ findTopK($\hat{y}^t_{lstm}, \hat{y}^t_{ast}, U$)
  $L \leftarrow \mathrm{L} \bigcup top_k$; $U \leftarrow$U – top$_k$
  $Y[top_k] \leftarrow \frac{1}{2} \times (\hat{Y}_{lstm}[top_k] + \hat{Y}_{ast}[top_k])$ ▷ Add k pseudo-targets to Y
end for

Algorithm 2 FindTopK

Input: Predictions $\hat{Y}_{lstm}$ from GCLSTM and $\hat{Y}_{ast}$ from astGCN, unlabelled set U
Output: Set of top k nodes top_k

$Y_{diff} \leftarrow \mathrm{abs}(\hat{Y}_{lstm} - \hat{Y}_{ast})$
$Y_{diff_{total}} \leftarrow \sum_{t=0}^{T} \mathrm{Y}^t_{diff}$ ▷ Add differences over all timesteps for each node
Set top_k as the k nodes in U with lowest $Y^n_{diff_{total}}$
return top_k

From a quick visual inspection, we notice that the AST-GCN model performs slightly worse than the GC-LSTM. However, during co-training, the AST-GCN converges while the GC-LSTM performs worse. The models were then evaluated on a validation set and the losses were calculated across all time-steps. The results are shown in Table. 1.

Table 1. Train and test loss for each model

Models	Test loss	Train loss
Supervised - LSTM	**3101.61**	2873.19
Supervised - AST	5130.56	4211.11
Co-Training - LSTM	3973.35	**1402.35**
Co-Training - AST	7644.07	2408.27
Co-Training - Combined	5808.71	1905.31

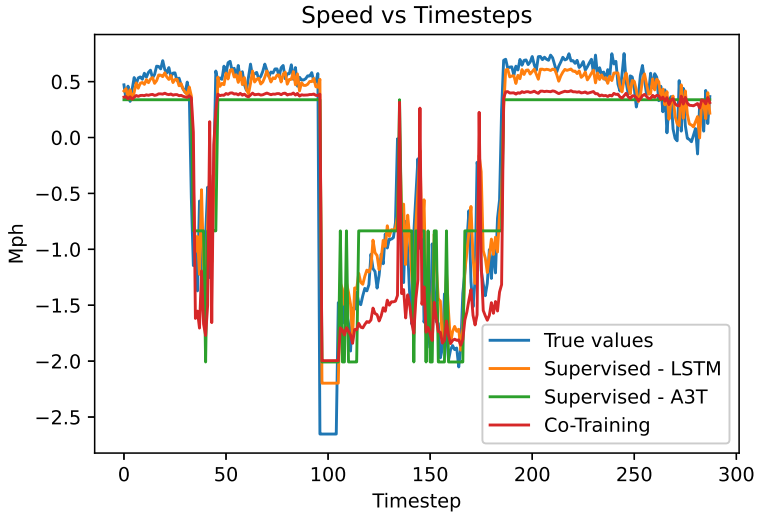

Fig. 2. Speed vs Timesteps for a particular node

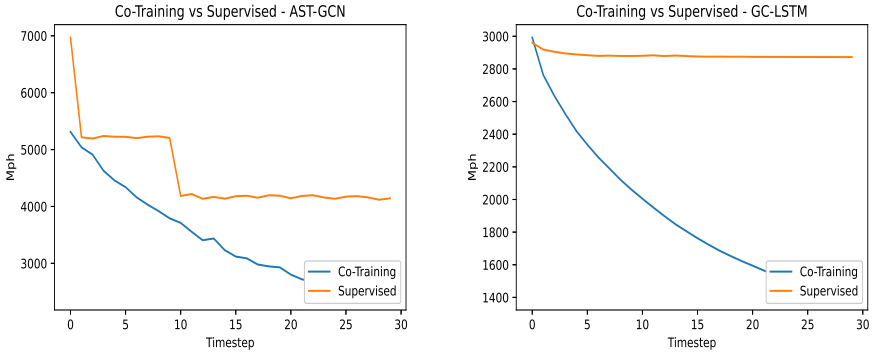

Fig. 3. Co-training vs Supervised Training Loss

5.2 GC-LSTM vs AST-GCN

In the supervised setting, GC-LSTM performs much better than the AST-GCN. This could be attributed to the fact that only a short window has been made available as input to both models. The short window limits the ability of attention weights in the AST-GCN to forecast optimal values, whereas the hidden state of the GC-LSTM is better suited to encoding the window values.

After co-training, the performance of AST-GCN and GC-LSTM become comparable. In Fig. 3. it is noticeable that the training loss decreases significantly in both AST-GCN and GC-LSTM after utilizing the co-training algorithm. Additionally, we report the standardized miles per hour for one sensory node across all time-steps which represents the range for one entire day, the results of which are displayed in Fig. 2.

6 Conclusions and Future Scope

In this paper, we propose a novel algorithm for forecasting traffic speeds in a spatial graph with temporal signals in the case of sensor failure. A custom co-training algorithm is implemented using two models which incorporate two conditionally independent views of the MetrLA dataset - a Graph Convolution embedded LSTM (GC-LSTM) and an Attention Temporal Graph Convolution Network (AST-GCN). We demonstrated that the link prediction features from GC-LSTM were able to boost the performance of the node-based AST-GCN. We observed a significant improvement in training loss after utilizing the co-training framework.

This solution mitigates the problem of scarcity of labelled data by simulating sensor failure in the traffic network. We establish that *GraphCoReg* has the potential to perform better than both equivalent supervised models. Both models show similar performance in the semi-supervised setting.

For the future, we outline three aspects to further refine and bolster our fundamental research. Firstly, we are in the process of tuning our base co-training model parameters by experimenting with other classes of temporal GNNs as well as hyper-tuning the window sizes for effective learning. Secondly, inspired by *Co-teaching* [8] and *Co-teaching+* [24] we intend to extend our proposed GraphCoReg top-k pseudo-target selection algorithm to select a certain range of loss examples and cross feed them during backpropagation. We speculate that loss adjustment would be a necessary addition to robustly train the base learners [21]. Last but not least, we hope to carry over our work across other benchmark datasets with an emphasis on spatio-temporal social media data. In doing so we wish to portray and evaluate the general performance of the *GraphCoReg* model across a variety of conditions.

References

1. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the eleventh annual conference on Computational learning theory. pp. 92–100 (1998)
2. Chen, J., Wang, X., Xu, X.: Gc-lstm: Graph convolution embedded lstm for dynamic link prediction. arXiv preprint arXiv:1812.04206 (2018)
3. Chong, Y., Ding, Y., Yan, Q., Pan, S.: Graph-based semi-supervised learning: A review. Neurocomputing **408**, 216–230 (2020)
4. Fritz, C., Dorigatti, E., Rügamer, D.: Combining graph neural networks and spatio-temporal disease models to improve the prediction of weekly covid-19 cases in germany. Scientific Reports **12**(1), 1–18 (2022)
5. Goldman, S.A., Zhou, Y.: Enhancing supervised learning with unlabeled data. In: Proceedings of the Seventeenth International Conference on Machine Learning. p. 327–334. ICML '00, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2000)
6. Guo, S., Lin, Y., Feng, N., Song, C., Wan, H.: Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. Proceedings of the AAAI Conference on Artificial Intelligence **33**(01), 922–929 (Jul 2019). https://doi.org/10.1609/aaai.v33i01.3301922, `https://ojs.aaai.org/index.php/AAAI/article/view/3881`
7. Guo, S., Lin, Y., Feng, N., Song, C., Wan, H.: Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 922–929 (2019)
8. Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., Sugiyama, M.: Co-teaching: Robust training of deep neural networks with extremely noisy labels. Advances in neural information processing systems **31** (2018)
9. He, Y., Zhao, Y., Wang, H., Tsui, K.L.: Gc-lstm: A deep spatiotemporal model for passenger flow forecasting of high-speed rail network. In: 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC). pp. 1–6. IEEE (2020)
10. Jagadish, H.V., Gehrke, J., Labrinidis, A., Papakonstantinou, Y., Patel, J.M., Ramakrishnan, R., Shahabi, C.: Big data and its technical challenges. Communications of the ACM **57**(7), 86–94 (2014)
11. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
12. Kostopoulos, G., Karlos, S., Kotsiantis, S., Ragos, O.: Semi-supervised regression: A recent review. Journal of Intelligent & Fuzzy Systems **35**(2), 1483–1500 (2018)
13. Li, S., Li, W.T., Wang, W.: Co-gcn for multi-view semi-supervised learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 4691–4698 (2020)
14. Li, Y., Yu, R., Shahabi, C., Liu, Y.: Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv:1707.01926 (2017)
15. Ning, X., Wang, X., Xu, S., Cai, W., Liping, Z., Yu, L., Li, W.: A review of research on co-training. Concurrency and Computation: Practice and Experience (03 2021). https://doi.org/10.1002/cpe.6276
16. Rossi, E., Chamberlain, B., Frasca, F., Eynard, D., Monti, F., Bronstein, M.: Temporal graph networks for deep learning on dynamic graphs. arXiv preprint arXiv:2006.10637 (2020)

17. Rozemberczki, B., Scherer, P., He, Y., Panagopoulos, G., Riedel, A., Astefanoaei, M., Kiss, O., Beres, F., , Lopez, G., Collignon, N., Sarkar, R.: PyTorch Geometric Temporal: Spatiotemporal Signal Processing with Neural Machine Learning Models. In: Proceedings of the 30th ACM International Conference on Information and Knowledge Management. p. 4564–4573 (2021)

18. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE transactions on neural networks **20**(1), 61–80 (2008)

19. Seo, Y., Defferrard, M., Vandergheynst, P., Bresson, X.: Structured sequence modeling with graph convolutional recurrent networks. In: International conference on neural information processing. pp. 362–373. Springer (2018)

20. Si, C., Chen, W., Wang, W., Wang, L., Tan, T.: An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1227–1236 (2019)

21. Song, H., Kim, M., Park, D., Shin, Y., Lee, J.G.: Learning from noisy labels with deep neural networks: A survey (2020). https://doi.org/10.48550/ARXIV.2007.08199, `https://arxiv.org/abs/2007.08199`

22. Wang, J., Luo, S.w., Zeng, X.h.: A random subspace method for co-training. vol. 36, pp. 195 – 200 (07 2008). https://doi.org/10.1109/IJCNN.2008.4633789

23. Wang, W., Zhou, Z.H.: A new analysis of co-training. In: ICML (2010)

24. Yu, X., Han, B., Yao, J., Niu, G., Tsang, I., Sugiyama, M.: How does disagreement help generalization against label corruption? In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 7164–7173. PMLR (09–15 Jun 2019), `https://proceedings.mlr.press/v97/yu19b.html`

25. Zhan, K., Niu, C.: Mutual teaching for graph convolutional networks. Future Generation Computer Systems **115**, 837–843 (2021)

26. Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., Deng, M., Li, H.: T-gcn: A temporal graph convolutional network for traffic prediction. IEEE Transactions on Intelligent Transportation Systems **21**(9), 3848–3858 (2019)

27. Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., Sun, M.: Graph neural networks: A review of methods and applications. AI Open **1**, 57–81 (2020)

28. Zhou, Z.H., Li, M.: Semi-supervised regression with co-training. In: Proceedings of the 19th International Joint Conference on Artificial Intelligence. p. 908–913. IJCAI'05, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2005)

29. Zhu, X., Ghahramani, Z.: Learning from labeled and unlabeled data with label propagation. Tech. rep. (2002)

A Related Work

A.1 Spatio-Temporal Data Mining

Spatio-temporal data is classically dealt with by processing the spatial information (using graph representation learning) and temporal information (using time-series methods) separately and then combining them by processing a time-series of spatial embeddings. However, recently with the introduction of *Recurrent GNNs* [19] and *Temporal Graph Networks* [16], and the explosion of spatio-temporal data, effective processing of such data has come into the spotlight. Spatio-temporal data mining has also been widely used in various applications such as Covid-19 forecasting [4] and traffic forecasting [26] [7].

B Model Descriptions

B.1 GC-LSTM

A *Graph Convolution embedded Long Short-Term Memory (GC-LSTM)* Network [9] [2] employs a GCN to learn spatial features in combination with an LSTM network capable of learning long-term feature dependencies, similar to a conventional LSTM. The model can thus make effective predictions on spatial-temporal data. For a dynamic network, this results in retention of traffic speeds over multiple time-steps, owing to it being a continuous phenomenon.

B.2 AST-GCN

A *Attention based Spatial Temporal Graph Convolution Network (ASTGCN)* proposed by Guo et al. [7] for traffic flow forecasting consist of three components based on the length of time intervals considered for temporal data: recent, daily-periodic and weekly-periodic. Each component is constituted, firstly, by an attention mechanism - both spatial attention and temporal attention - to capture spatio-temporal correlation within the data. Secondly, it consists of a convolution module - graph convolutions to capture spatial features of the traffic network as well as temporal convolutions to capture traffic flow from various time-steps.

C Data Description

For the purposes of this preliminary study, we check the validity of GraphCoReg on the MetrLA benchmarking dataset [10], discretized and formulated as a sensor network as in [14]. The distribution of sensors is shown in Fig. 4. This dataset consists of data from 207 loop detectors in the Los Angeles highway network over 4 months. Each data sample is collected at 5-minute intervals. The goal is to forecast the traffic speed (target) at each of these detectors, given the temporal signals which are features of traffic flow like traffic volume at the detectors.

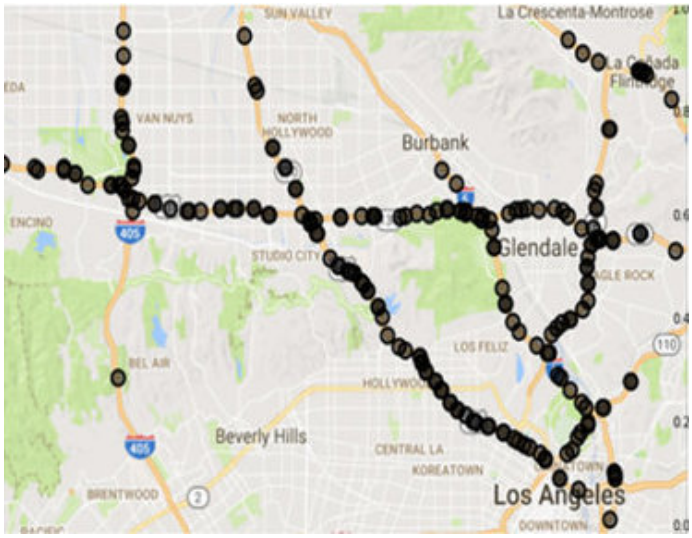

Fig. 4. Sensor distribution in METR-LA Dataset [14]