LLM-based Online Prediction of Time-varying Graph Signals (Student Abstract)

Dayu Qin*, Yi Yan*, Ercan Engin Kuruoglu†

Tsinghua-Berkeley Shenzhen Institute, Tsinghua Shenzhen International Graduate School, Tsinghua University
kuruoglu@sz.tsinghua.edu.cn

Abstract

In this paper, we propose a novel framework that leverages Large Language Models (LLMs) for predicting missing values in time-varying graph signals by exploiting spatial and temporal smoothness. We leverage the power of LLM to achieve a message-passing scheme. For each missing node, its neighbors and previous estimates are fed into and processed by LLM to infer the missing observations. Tested on the task of the online prediction of wind-speed graph signals, our model outperforms online graph filtering algorithms in terms of accuracy, demonstrating the potential of LLMs in effectively addressing partially observed signals in graphs.

Introduction

The application of Large Language Models (LLMs) in graph machine learning tasks has increasingly gained research attention. For example, traditional graph tasks such as substructure counting and finding the shortest path can be achieved by LLM (Chai et al. 2023; Wang et al. 2024). There are many applications of graph learning, such as forecasting weather recordings, predicting market crises in financial data, and identification of autism patients (Yan, Kuruoglu, and Altinkaya 2022; Qin and Kuruoglu 2024; Qin, Chen, and Kuruoglu 2024). However, these tasks often are time-varying multi-variate numeric signals containing missing values, which is different from the language sequences commonly seen in LLM-related tasks. In our work, we developed an online framework that leverages LLMs to predict missing values in time-varying graph data. By online framework, we mean that this framework operates in real-time, making predictions as new observations arrive, exploiting both spatial and temporal smoothness in the time-varying graph signal. The LLM model aggregates signals from neighboring nodes and prior time steps, outperforming traditional methods in accuracy.

Methodology

The proposed framework leverages the smoothness assumption in Graph Signal Processing (GSP) on an online (real-time) time-varying regression task. For a graph $\mathcal{G}$ with N nodes, we assume that the time-varying graph signals nodes $\boldsymbol{x}[t]$ exhibits smoothness over time and space. This implies that for an observation $\boldsymbol{o}[t]$ containing missing node observations, the ground truth $\boldsymbol{x}[t]$ can potentially be inferred from the temporal trends or spatial proximity to other observed values. In the context of graph-based formulation, a transductive setting naturally appears where observed data and unobserved data coexist, enabling the LLM to exploit the intrinsic structure of the graph for more accurate predictions on the unobserved nodes when the topological context is fed into LLM along with the data (Liu and Wu 2023). In GSP, the conventional method is to design a filter in a graph convolution to process graph signals: $\hat{\boldsymbol{x}}[t+1] = \hat{\boldsymbol{x}}[t] + \Delta_x$, with $\Delta_x = \sum_0^P \theta_p \mathbf{L}^p \boldsymbol{o}[t] \approx \mathbf{U}\boldsymbol{\Sigma}\mathbf{U}^T\boldsymbol{o}[t]$. Here $\hat{\boldsymbol{x}}[t]$ is the estimated $\boldsymbol{x}[t]$, $\mathbf{L}$ is the graph Laplacian matrix used to represent the graph, $\mathbf{U}$ is the eigenvector matrix of $\mathbf{L}$, θ_p are the spatial parameters, and $\boldsymbol{\Sigma}$ are the spectral parameters. However, the inference power relies heavily on the redefinition of the weight parameters, which is restricted by prior knowledge of the underlying data and the operations defined by the graph representation $\mathbf{L}$. Additionally, conventional GSP techniques process all the nodes at once using global operations, often neglecting the localities within individual node neighborhoods.

To predict missing observations, our LLM framework mimics a message-passing mechanism represented by localized aggregation at each individual node:

$$\text{agg}(x_v) = \Omega\left(\{(x_v[t], x_u[t]) \mid u \in \mathcal{N}_v\}\right), \tag{1}$$

where for the node v in $\mathcal{G}$, Ω is the aggregation function, $x_v[t]$ is the signal on v and it has $\mathcal{N}_v$ 1-hop neighboring nodes (including self-loop) denoted with the subscript $u = 1 \dots N_v$. In GSP, the localized node adjacencies used can be deducted from the adjacency matrix (Yan, Peng, and Kuruoglu 2024). In our context, message passing techniques to divide the global node observations $\boldsymbol{o}[t]$ and the estimation $\hat{\boldsymbol{x}}[t]$ into localized representation. Using message passing, data representation is on the local level: each node v knows only its own signal and its one-hop neighboring node signals (Gilmer et al. 2017). This localized representation of graph signals is fed into the context and the prompt of LLM. At each time step t, the current partial observation $o_v[t]$ of a node, along with historical data (either observed or reconstructed), is provided as the input to the LLM. The LLM

*These authors contributed equally.

†Corresponding author.

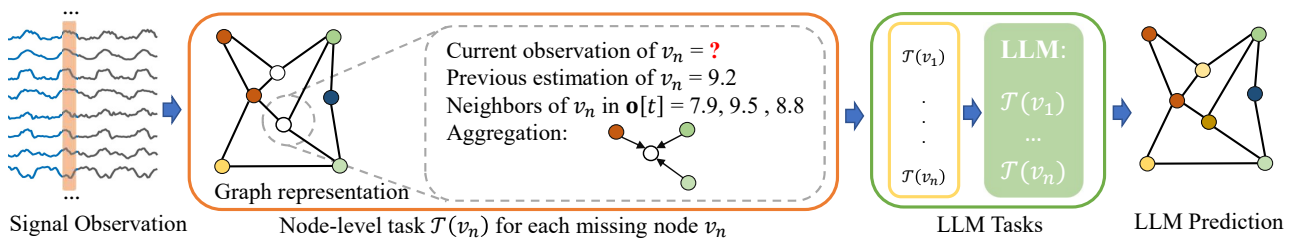

Figure 1: One single iteration of the proposed algorithm

serves as a localized aggregation where its task $\mathcal{T}(v_n)$ is to interpret input from the neighbors of a missing node v and the previous estimate of $\hat{x}_v[t-1]$, then inferring the missing observation $\hat{x}_v[t]$. The online learning setup means that our framework has to make quick predictions based on limited observation, so we are making only 1-step predictions. The algorithmic procedure is shown in Algorithm 1. Since this aggregation is defined locally on each node, the aggregation process is fed iterative to LLM for all N nodes, which in turn is mapped back onto each missing node observation.

In Algorithm 1, the process initiates by identifying any new observations at time t. For each missing node v_n in $\mathcal{G}$, values from its own previous estimate and those of its neighbors in the current observations $\boldsymbol{o}[t]$ are collected. This collected data, including neighbor values from past estimations at $t-1$, informs the LLM task $\mathcal{T}(v_n)$, which then predicts the current value $\hat{x}_v[t]$ for the missing node. This prediction is iteratively performed for all missing nodes, culminating in a comprehensive state estimation $\hat{\boldsymbol{x}}[t]$ for time t. By employing Algorithm 1, the LLM iteratively learns the patterns from spatial neighbors and temporal trends to predict the missing data accurately. The approach also allows for the integration of LLM predictions with graph-based structures, enhancing the prediction process.

Experiment Results and Discussion

In our experiments, we employed time-varying hourly wind speed graph signals with $N = 197$ nodes and $T = 95$ time points to validate the proposed methodology from (Yan, Kuruoglu, and Altinkaya 2022). The data is uniformly set 30% of the nodes to be missing. The GPT-3.5-turbo model from the OpenAI API was utilized to predict these missing values based on the temporal and spatial patterns in the data (Brown et al. 2020). GPT-3.5-turbo is an advanced LLM, part of the GPT-3 family, that excels in understanding and generating human-like text. It can handle complex prompts and generate highly accurate predictions, making it suitable for tasks involving pattern recognition and missing data prediction.

Algorithm 1: Online Prediction of Unobserved Nodes

while new observations $\boldsymbol{o}[t]$ are available **do**
  for each missing node v_n in unlabelled node set **do**
    Collect the previous estimation of v_n
    Collect observed neighbor signals of v_n in $\boldsymbol{o}[t]$
    Form LLM task $\mathcal{T}(v_n)$ from aggregation (1)
    Feed $\mathcal{T}(v_n)$ into LLM and let LLM predict $\hat{x}_v[t]$
  end for
  Collect all $\hat{x}_v[t]|_{n=1\ldots M}$ and form $\hat{\boldsymbol{x}}[t]$
end while

The GPT-3.5-turbo model was provided with the observations from the previous node to predict the missing values for the same node at the next time point. By inputting only one time point at a time, we can achieve online prediction and prevent the LLM from peeking into future data. Additionally, in the prompt, we explicitly instruct it not to use chat memory. Each algorithm is repeated 5 times. Then we can calculate the MSE $= \frac{1}{5NT}\sum_{i=1}^{N}\sum_{t=1}^{T}(x_i[t]-\hat{x}_i[t])^2$ of the predicted value and evaluate the results. In the experiments, we used GLMS (D. Lorenzo et al. 2016) and G-Sign (Yan, Kuruoglu, and Altinkaya 2022) as baseline GSP algorithms for comparison. The result is shown as follows:

Based on the MSE comparison between different algorithms, the results demonstrate that the GPT-3.5-turbo model significantly outperforms both the GLMS and G-Sign algorithms in handling missing data within graph structures. This may be attributed to the fact that adaptive filters are primarily designed for noise reduction and are less effective for pure noiseless spatiotemporal prediction tasks compared to powerful models like GPT, which suggests that the GPT model holds substantial potential in addressing challenges related to partially observed time-varying graph signals.

Limitations There are rare occasions (occurred on only one node during our 5 experiment runs) where the model outputs NaN values and the cause remains unclear. In such cases, a practical workaround is employed: the missing values are substituted with the average of the previous time points. Additionally, batch-feeding N nodes tasks into LLM often results in the number of returns not being N. While the GPT model demonstrates substantial potential, addressing these limitations through complementary strategies remains necessary to further improve its robustness in scenarios involving noisy, time-varying, and incomplete data.

Model	MSE
GLMS Algorithm	3.396
G-Sign Algorithm	3.718
GPT-3.5-turbo	**1.560**

Table 1: MSE Comparison of Different Algorithms

Acknowledgments

This work is supported by the Tsinghua Shenzhen International Graduate School Start-up fund under Grant QD2022024C, Shenzhen Science and Technology Innovation Commission under Grant JCYJ20220530143002005, and Shenzhen Ubiquitous Data Enabling Key Lab under Grant ZDSYS20220527171406015.

References

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language models are few-shot learners. *NeurIPS*.

Chai, Z.; Zhang, T.; Wu, L.; Han, K.; Hu, X.; Huang, X.; and Yang, Y. 2023. Graphllm: Boosting graph reasoning ability of large language model. *arXiv preprint arXiv:2310.05845*.

D. Lorenzo, P.; Barbarossa, S.; Banelli, P.; and Sardellitti, S. 2016. Adaptive Least Mean Squares Estimation of Graph Signals. *IEEE Trans. Signal Inf. Process. Netw.*, 2(4): 555 – 568.

Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; and Dahl, G. E. 2017. Neural message passing for quantum chemistry. In *International conference on machine learning*, 1263–1272. PMLR.

Liu, C.; and Wu, B. 2023. Evaluating large language models on graphs: Performance insights and comparative analysis. *arXiv preprint arXiv:2308.11224*.

Qin, D.; Chen, Y.; and Kuruoglu, E. E. 2024. Exploring Neurofunctional Phase Transition Patterns in Autism Spectrum Disorder: A Thermodynamics Parameters Analysis Approach. *arXiv preprint arXiv:2409.01039*.

Qin, D.; and Kuruoglu, E. E. 2024. Graph learning based financial market crash identification and prediction. In *2024 IEEE Conference on Artificial Intelligence (CAI)*, 650–651. IEEE.

Wang, H.; Feng, S.; He, T.; Tan, Z.; Han, X.; and Tsvetkov, Y. 2024. Can language models solve graph problems in natural language? *NeurIPS*, 36.

Yan, Y.; Kuruoglu, E. E.; and Altinkaya, M. A. 2022. Adaptive sign algorithm for graph signal processing. *Signal Process.*, 200: 108662.

Yan, Y.; Peng, C.; and Kuruoglu, E. E. 2024. Graph Signal Adaptive Message Passing. *arXiv*.