CS768: Learning With Graphs Course Project
Learning MLPs on Dynamic Graphs for Node Classification

Yash Salunkhe - 210020156
Ravi Kumar - 210010052

GitHub Link: https://github.com/ravioli1369/cs768-project

1 Abstract

This project explores the possibility of an extension of NOSMOG [13] on temporal dynamic graphs that leverages time aggregation over evolving snapshots to adapt its noise-robust, structure-aware MLP framework to temporal settings. By performing time aggregation on dynamic graph snapshots and incorporating temporal distillation alongside adversarial feature augmentation, Temporal-NOSMOG achieves efficient, robust, and structure-aware inference on temporal graph benchmarks.

2 Problem Statement

Node classification on dynamic graphs involves predicting labels for nodes whose connectivity and features evolve over time as the network changes. Traditional static GNNs, designed for fixed graph structures, fail to capture the temporal dependencies and shifting neighborhoods crucial for accurate classification in evolving settings. Temporal aggregation across discrete snapshots or continuous-time events enables models to integrate historical context and update node representations dynamically. NOSMOG's noise-robust, structure-aware MLP architecture has demonstrated strong performance on static node classification benchmarks by leveraging position features, representational distillation, and adversarial feature augmentation to approximate GNN accuracy with far greater inference efficiency. However, without explicit temporal adaptation, NOSMOG struggles to maintain classification accuracy as graph connectivity patterns shift and feature distributions drift over time. By integrating temporal aggregation, our approach adapts NOSMOG to dynamic node classification tasks, enabling the MLP to capture both structural and temporal dependencies in a lightweight and efficient end-to-end framework.

3 Literature Review

3.1 Node Classification

In the past few years, node classification is one of the most typical research directions of graph analysis due to the widespread application scenarios. In particular, the objective of the node classification task is to predict a particular class for each unlabeled node in the graph based on the graph information [8]. Various GNN models have been developed to tackle this problem [9, 6] Embedding methods have been found to perform the best [11, 5] for node classification tasks as they consistently outperform traditional approaches such as spectral clustering and matrix factorization [3, 12].

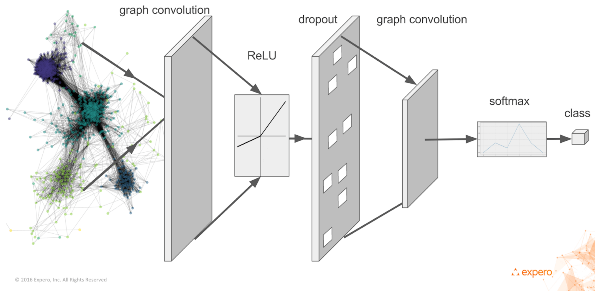

Figure 1: An illustration of a traditional GNN performing node classification. Source - [4]

3.2 Dynamic Graphs

Previous research [10] revealed that temporal motifs unfold across varying timescales in evolving graphs. Thus, in recent years, systems with dynamic networks demand models that can perform node classification on dynamic graphs. [2, 15] are exhaustive benchmarks in this subproblem on various datasets. However, some of these models are computationally expensive and struggle when the data is noisy. In real-world applications, it's better to build models that can handle noise well while still running efficiently in a timely manner.

4 Methodology

4.1 Temporal-NOSMOG

Temporal-NOSMOG extends the original NOSMOG framework to temporal graphs by encoding temporal dynamics directly into the graph's adjacency matrix. Starting from a set of timestamped interactions (u, v, t), each edge is assigned a time-decayed weight using an exponential function $w_{uv} = \alpha^{(1-\tilde{t})}$, where $\tilde{t} \in [0, 1]$ is the normalized timestamp and α is a decay constant (e.g., 0.9), ensuring that recent edges are weighted more heavily. The resulting weighted adjacency matrix $\mathbf{A}_t$ reflects the temporal importance of connections and is paired with a node feature matrix $\mathbf{X} \in \mathbb{R}^{N \times F}$, aligned by a consistent node indexing scheme. These are then cached and passed through the standard NOSMOG preprocessing: extracting the largest connected component, normalizing $\mathbf{A}_t$ and $\mathbf{X}$, binarizing the label matrix $\mathbf{Y} \in \{0, 1\}^{N \times C}$, and splitting nodes into training, validation, and test sets. Then teacher GNNs are trained and distill knowledge to student MLPs following the NOSMOG framework

4.2 Mathematical Framework for Temporal-NOSMOG

Let

$$\mathcal{E} = \{(u_i, v_i, t_i)\}_{i=1}^{m}$$

be the set of time-stamped directed edges. We construct a time weighted adjacency via the tie-decay [1] model, in which each event at time t_i instantaneously boosts the edge weight, and it then decays exponentially in the absence of further interactions. If one normalizes timestamps to $[0, 1]$ via

$$\tilde{t}_i = \frac{t_i - t_{\min}}{t_{\max} - t_{\min} + \epsilon},$$

then a discrete analogue is

$$A_{uv} \;=\; \sum_{i:(u_i=u, v_i=v)} \alpha^{\,1-\tilde{t}_i}, \tag{0.1}$$

with $\alpha = 0.9$ for our case.

Finally, we remove self-loops, symmetrize and normalize, but A forms the structure for downstream embedding.

4.3 NOSMOG: Noise-Robust, Structure-Aware MLPs on Graphs

Given node content features $X \in \mathbb{R}^{n \times d}$ and the graph encoded by A, NOSMOG seeks an MLP embedding

$$f_{\text{MLP}} \colon (X_v, P_v) \;\mapsto\; z_v \in \mathbb{R}^h,$$

where P_v are *position features* (DeepWalk embeddings) that encode v's location in the graph.

Let $f_{\text{GNN}}(X_v, S_v)$ be a teacher GNN embedding using v's multi-hop neighborhood S_v. NOSMOG optimizes three coupled objectives:

1. **Supervised loss** on labels y_v:

$$\mathcal{L}_{\text{CE}} = \sum_{v=1}^{n} \ell\big(f_{\text{MLP}}(X_v, P_v),\, y_v\big). \tag{Cross-Entropy}$$

2. **Representational Similarity Distillation** [7] to transfer structural knowledge:

$$\mathcal{L}_{\text{distill}} = \sum_{v=1}^{n} \ell_{\text{sim}}\big(f_{\text{MLP}}(X_v, P_v),\, f_{\text{GNN}}(X_v, S_v)\big), \tag{Distill}$$

   where ℓ_{sim} measures distance (e.g. KL-Divergence) between student and teacher embeddings

3. **Adversarial Feature Augmentation** for noise robustness:

$$\mathcal{L}_{\text{adv}} = \sum_{v=1}^{n} \max_{\|\delta_v\| \le \epsilon} \ell\big(f_{\text{MLP}}(X_v + \delta_v,\, P_v),\, y_v\big), \qquad \text{(Adv. Augment)}$$

where each δ_v is a worst-case perturbation of bounded norm.

The full NOSMOG loss is a weighted sum:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{CE}} \;+\; \lambda_2 \mathcal{L}_{\text{distill}} \;+\; \lambda_3 \mathcal{L}_{\text{adv}}, \qquad (0.2)$$

5 Results

Table 1: Performance comparison between models on the Wikipedia and MOOC datasets
The results are mean ± variance for 10 runs. Best performance is in bold, 2nd best is underlined

Method	**Wikipedia**	**MOOC**
Only SAGE [6]	0.8528±0.0021	**0.6823±0.003**
Only GCN [9]	0.8366±0.0001	0.6522±0.0026
Only MLP	0.6321±0.0011	0.4080±0.0025
Teacher (SAGE) + MLP	<u>0.8591±0.0035</u>	<u>0.6764±0.0023</u>
Teacher (GCN) + MLP	0.8078±0.0012	0.6471±0.0017
DyRep [14]	**0.873±0.002**	0.661±0.012
TGAT [15]	0.8±0.01	0.673±0.006

6 Conclusions

In this work, we introduced ***Temporal-NOSMOG***, an extension of the NOSMOG framework for dynamic graphs. Our approach incorporates temporal aggregation mechanisms that effectively capture the evolving nature of node interactions over time. We evaluated Temporal-NOSMOG on two benchmark temporal graph datasets—Wikipedia and MOOC—and demonstrated that our method achieves competitive performance in node classification tasks. The results highlight the strength of our temporal modeling approach and establish Temporal-NOSMOG as a viable alternative to state-of-the-art dynamic graph learning models.

7 Contributions

Table 2: Contribution of each member

Name	Contribution
Yash Salunkhe (210020156)	Temporal aggregation implementation, Experiments
Ravi Kumar (210010052)	Data curation, Dataloader implementation, Report Writing

References

[1] Walid Ahmad, Mason A. Porter, and Mariano Beguerisse-Díaz. Tie-decay networks in continuous time and eigenvector-based centralities, 2021.

[2] Mohammad Ali Alomrani, Mahdi Biparva, Yingxue Zhang, and Mark Coates. Dyg2vec: Efficient representation learning for dynamic graphs. *arXiv preprint arXiv:2210.16906*, 2022.

[3] Lei Chen, Shunwang Gong, Joan Bruna, and Michael Bronstein. Attributed random walk as matrix factorization. In *neural information processing systems, graph representation learning workshop*, 2019.

[4] Expero Inc. Node classification by graph convolutional network, 2020. Accessed: 2025-04-19.

[5] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks, 2016.

[6] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs, 2018.

[7] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

[8] Przemyslaw Kazienko and Tomasz Kajdanowicz. Label-dependent node classification in the network. *Neurocomputing*, 75(1):199–209, 2012.

[9] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2017.

[10] Ashwin Paranjape, Austin R Benson, and Jure Leskovec. Motifs in temporal networks. In *Proceedings of the tenth ACM international conference on web search and data mining*, pages 601–610, 2017.

[11] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

[12] Lei Tang and Huan Liu. Leveraging social media networks for classification. *Data mining and knowledge discovery*, 23:447–478, 2011.

[13] Yijun Tian, Chuxu Zhang, Zhichun Guo, Xiangliang Zhang, and Nitesh Chawla. Learning MLPs on graphs: A unified view of effectiveness, robustness, and efficiency. In *International Conference on Learning Representations*, 2023.

[14] Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. Dyrep: Learning representations over dynamic graphs. In *International Conference on Learning Representations*, 2019.

[15] Dongkuan Xu, Wei Cheng, Dongsheng Luo, Yameng Gu, Xiao Liu, Jingchao Ni, Bo Zong, Haifeng Chen, and Xiang Zhang. Adaptive neural network for node classification in dynamic networks. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 1402–1407. IEEE, 2019.