Leveraging Cross-Snapshot Attention for Identifying Graph Propagation Patterns in Dynamic Real-World Networks

Til Schniese[*][0009−0008−5301−7881]
Christian Medeiros Adriano[0000−0003−2588−9937]
Holger Giese[0000−0002−4723−730X]

Hasso Plattner Institute, University of Potsdam, Germany

Abstract. Dynamic real-world networks, encompassing both digital and physical realms, inherently display complex spatio-temporal phenomena. A common manifestation is the propagation of node states, where information is disseminated through network edges via everyday human and system interactions. Given the potential threats like virus spread and fake news dissemination, it is critical to quickly and effectively identify propagation patterns and their harmful instances. Although various approaches exist for classifying spatio-temporal graphs, we argue that current methods overlook essential characteristics of propagation behavior, such as the causal-effect relationships between node state transitions. To address this gap, we propose a novel cross-snapshot attention method that leverages the unique features of propagations originating from specific nodes over time. The novelty lies in the element-wise attention weight calculations across consecutive snapshots, linking changes in propagation states to local network regions. Our method surpasses a set of established graph neural network techniques in accuracy across datasets designed to simulate complex real-world propagation dynamics. We performed a series of ablation studies to confirm the positive impact of the cross-snapshot attention module and its robustness to missing snapshots, which shows that our method experiences smoother performance degradation compared to the state-of-the-art.

Keywords: Learning on Graphs · Graph Neural Networks · Propagation

1 Introduction

Dynamic real-world networks are subject to complex dynamics [30], such as human interactions, social media postings, financial transactions, or inter-node computer communication. A specific kind of dynamics is the propagation of node states across the network's edges, mirroring a variety of these phenomena. While some propagations depict typical network trends, others may represent detrimental behaviors, such as the spread of diseases [32] or fake news [35].

[*] Corresponding author: til.schniese@gmail.com

In order to effectively identify harmful propagation patterns, various graph-based approaches exist, for instance, Graph Neural Networks (GNNs) and graph kernel methods. Among GNNs, the attention-based approaches [24,37,22,17] allow expressing the spatio-temporal features via representative low-dimensional embeddings. Graph kernel methods compute distances in a metric space by relying on symmetric, positive semi-definite functions, like graph diffusion models [6] and temporal graphlet kernels [21]. While GNNs and graph kernel methods ascertain node similarities based on spatio-temporal features, they neglect the causal-effect relationships between the node states, which are strongly governed by domain-specific propagation properties [32,31,35].

To investigate this gap, we design a cross-snapshot attention method (*CSA*) that derives attention weights from successive propagation snapshots that associate changes in the node states to particular regions in the networks. By design, these weights inherently estimate the underlying causal-effect relationships. Unlike existing methods, these weights are determined element-wise, offering better scalability than conventional dot product attention operators, while sidestepping traditional message passing across nodes. We synthesize datasets that mirror real-world network dynamics that make the task of classifying propagation patterns particularly challenging. We compare our model to popular traditional GNNs such as a graph convolutional network [10] and a graph attention network [28] (both temporally enhanced), as well as state-of-the-art dynamic GNNs, like DySAT [24] and Roland [33]. *CSA* outperforms traditional approaches on all datasets and remains competitive with state-of-the-art methods, while surpassing them in particular cases.

To increase confidence in our results[1], we perform ablation studies and a robustness evaluation against missing snapshots (a frequent condition in real-world networks [18]). While the ablation studies showcase the positive impact of the novel attention mechanism, the robustness evaluation reveals that *CSA* degrades more smoothly than alternative methods. To mitigate degradation even further, we propose a self-supervised *prompting* model that recovers missing information from snapshots. Ultimately, our contributions are two-fold: (1) a set of synthetic datasets that emulate the complex network dynamics of real-world propagations and (2) a novel cross-snapshot attention method that exploits the causal-effect relationships of the propagation phenomenon.

2 Preliminaries

We consider a graph $G = (V, E)$, where each node $u \in V$ possesses a specific state $s(u) \in \{0, 1\}$. A subset of nodes $A \subseteq V$ are identified as *anchor nodes* where initially $s(u) = 1$, whereas for the remaining nodes $s(u) = 0$. A propagation is defined as the subsequent transfer of the anchor nodes' state via the edges E over a specified period, such that the state for a node in $V \backslash A$ transitions from 0 to 1. When $s(u) = 1$ for a node u, it is referred to as *covered*, while nodes of state $s(u) = 0$ are considered *uncovered*.

[1] Reproduction package https://github.com/hpi-sam/cross-snapshot-attention-gnn

The propagations in a graph are observed at discrete intervals, modeled through a spatio-temporal (dynamic) graph (STG), according to [19], such that a STG is a collection of graph snapshots

$$\mathbb{G} = \{G_t\}_{t=0}^{T} G_t = (V_t, E_t, X_{V_t}, X_{E_t}), \tag{1}$$

where V_t is a set of nodes at time t, E_t is a set of edges at time t, X_{V_t} is a set of node attributes at time t, and X_{E_t} is a set of edge attributes at time t. Specifically, a dedicated node attribute represents the state s in each snapshot and an additional edge attribute indicates the related propagation edge. This definition implies causality in state propagations, i.e., *causal-effect relationships* between nodes, excluding state transitions not induced by a neighboring node.

3 The State of the Art

Recently, graph neural networks enabled classification tasks by transforming the graph features into structural preserving embeddings. Because traditional GNNs are not tailored to dynamic networks, new embedding methods were proposed where spatio-temporal features are used to transform changes (snapshots) into a static graph, thus enabling the reuse of conventional GNNs [9,23]. As this strategy potentially compromises the sequential integrity of the snapshots, additional modules are added by dynamic GNN approaches [22,17,26], e.g., long short-term memory (LSTMs) [11,7] and gated recurrent units (GRUs) [34,2]. Meanwhile, new methods rely on self-attention strategies to capture the evolution of the graph structure [24,37] and related node and edge attributes [5], for instance, by integrating transformers [36]. While current approaches ascertain pair-wise node similarity scores based on pertinent attributes [16], they neglect the particular behavior of the propagation phenomenon, i.e., that evolutions originate from specific nodes and their propagation is mediated by a combination of node and edge features, and the network topology.

An alternative to GNNs are the graph kernel methods [20], which consist of symmetric, positive semi-definite functions defined on the space of graphs and that can be expressed as an inner product in a Hilbert space. These methods are applied to estimate temporal dynamics on networks [6] and are shown to be competitive with GNNs [21]. Nonetheless, we did not prioritize them as baselines due to their limitation of fixing the set of features before training [20].

4 Data

Current spatio-temporal graph datasets [8] (e.g., Enron [2], Bitcoin-Alpha [3], or LastFM [4]) do not contain the ground truth of the causal-effects that origi-

[2] Shetty et al. The Enron email dataset database schema and brief statistical report (2004)

[3] Kumar et al. Edge weight prediction in weighted signed networks (2016)

[4] Kumar et al. Predicting dynamic embedding trajectory in temporal interaction networks (2019)

nated the nodes' attributes changes. Hence, we carefully generated synthetic samples, mimicking realistic characteristics of real-world propagations. The underlying graph structures are generated via random graph generators including Erdős–Rényi, Watts-Strogatz, Barabási-Albert, and Expander graphs.

Regarding the propagation phenomenon, we devise a sampling strategy inspired by the spread of pathogens within a population. On one hand, certain diseases are more infectious, e.g., due to higher viral loads [12]. Conversely, specific populations may be more susceptible to particular diseases, e.g., because of immune evasion mechanisms [1]. Accordingly, we model *emission* to represent the state dissemination as covered nodes, and *absorption* to constitute the susceptibility of uncovered nodes. Finally, we establish *transmission* to configure the likelihood that edges, combining infectious and susceptible connections, induce further dissemination. Intuitively, a sample is produced through a four-step process: (1) sampling anchor nodes, (2) sampling adjacent edges of covered nodes, (3) sampling adjacent edges of uncovered nodes, and (4) determining propagation ability for matching edges sampled in steps (2) and (3). Each step can be parameterized through time-dependent node and edge functions to create diverse scenarios mirroring desired real-world characteristics.

We introduce five datasets: COV-19, replicating different COVID-19 mutation infection chains with variable transmissibility [32,14] and incubation time [31]; FAKE, based on social media reposting behavior [29,35], distinguishing between true and fake news; BTC-BP, representing block propagations in a Bitcoin network with variations in edge broadcasting speed [3,25]; DDoS, incorporating nodes that go offline during information exchange; and WAV, featuring propagations with peak transmission shifted over the observation time.

5 Approach

Our general idea of cross-snapshot attention (CSA) is to learn the broad concepts and inner workings around the propagation phenomenon autonomously, to automatically locate and exploit granular differences to classify propagations more accurately. Such a model must directly examine the changes between successive snapshots, holding crucial information like who propagated to whom, how many new nodes became covered, and which nodes possess covered neighbors so they could become covered next. We incorporate an attention module that emphasizes nodes pertinent to such propagation dynamics across snapshots, aiming to learn structural and temporal propagation characteristics.

This cross-snapshot attention model is inspired by [13], who developed graph matching networks to generate representations for graph pairs, that can be used to calculate similarity scores between graphs. However, unlike [13], we focus on the dissimilarities between graph snapshots in order to adequately capture propagation dynamics. Specifically, our cross-snapshot attention model comprises three main components: (1) a snapshot encoder, (2) a cross-snapshot attention layer, and (3) a set of aggregators.

Snapshot Encoder. Initially, an encoder is deployed individually per snapshot, mapping the node attributes to higher dimensional node embeddings via a linear layer, where H_t represents the transformed embeddings for all nodes at the time t. Assuming a number of snapshots T, the initial node attributes for a node n that will be covered at time t correspond to a T-dimensional vector, reflecting the propagation state. For each entry i where $i < t$, the value is set to 0. Once $i >= t$, the value changes to 1, indicating the propagation at time t.

Cross-Snapshot Attention Layer. Utilizing the snapshot-specific node embeddings, we first compute the differences $\varDelta$ between two consecutive snapshots.

$$\varDelta_t = H_t - H_{t-1}, \quad \forall t \in \{1, 2, ..., T-1\} \tag{2}$$

We formulate attention queries Q and attention keys K by mapping H_t and H_{t-1} through another linear layer and apply cross-snapshot attention as follows:

$$\varDelta'_t = \varDelta_t \odot (A_{t-1} \cdot \text{softmax}(Q_t \odot K_t)), \quad \forall t \in \{1, 2, ..., T-1\} \tag{3}$$

Unlike traditional attention [27], which applies a softmax to the dot product of the query and key vectors to derive a similarity score between pairs, we compute the element-wise multiplication (Hadamar product) between queries and keys. This adaptation ensures that interactions are only captured between matching dimensions in Q and K, corresponding to the computation of attention scores over individual features of the same node as they interact from one point in time to the next. The dot product of attention weights and the adjacency matrix A_{t-1} at the prior snapshot then yields updated attention weights for each node, derived from the sum of interaction scores within their local neighborhood. This way, each node accumulates the attention weights from its neighbors and gains higher attention when its neighborhood received high attention. These context-dependent node importance scores are multiplied with the snapshot differences using another element-wise operation. As a result, inter-snapshot changes of nodes are associated with neighboring nodes, weighted by attention. Intuitively, the model is tasked to explain attribute changes in snapshots by emphasizing specific regions in the preceding snapshot. Finally, we combine the weighted snapshot differences $\varDelta'$ with the previous snapshot embeddings (represented via $\oplus$) to receive node embeddings H' for each snapshot delta, introducing non-linearity through an additional ϕ function (ReLU).

$$H'_t = \phi(W_{concat} \cdot (H_{t-1} \oplus \varDelta'_t) + b_{concat}), \quad \forall t \in \{1, 2, ..., T-1\} \tag{4}$$

Aggregators. Following the cross-snapshot attention, a propagation aggregator takes the node embeddings H' for each snapshot delta and computes propagation-level node embeddings for the complete sequence of snapshots. At present, a lightweight mean operation is used, which may be replaced with more sophisticated aggregation methods like an LSTM or GRU. On top, we employ an aggregator to generate a lower-dimensional embedding from the propagation's node embeddings using gating vectors for each node, similar to [13].

Table 1: Mean accuracy and standard deviation for each model in the transductive setting on the proposed datasets. Highlighted values represent the best-performing model, and "c" denotes the number of classes in the datasets.

Model	COV-19 (c=4)	FAKE (c=3)	BTC-BP (c=2)	DDoS (c=2)	WAV (c=2)
GCN	85.04 ± 9.58	39.29 ± 8.05	56.13 ± 5.80	65.11 ± 10.39	82.15 ± 8.06
GAT	93.99 ± 2.09	**86.59 ± 2.19**	68.08 ± 3.10	81.65 ± 2.70	89.03 ± 2.85
Roland	92.80 ± 0.57	78.91 ± 1.71	64.10 ± 7.89	59.80 ± 13.88	81.30 ± 4.60
DySAT	93.90 ± 1.14	85.17 ± 0.96	68.60 ± 0.65	**87.70 ± 1.48**	**90.80 ± 1.44**
CSA	**94.70 ± 0.45**	82.49 ± 3.86	**73.00 ± 2.15**	83.80 ± 1.96	89.20 ± 2.05

Runtime Complexity. Given d node feature dimensions, an average number of nodes n, and an average node degree of k across T snapshots, CSA has a theoretical runtime complexity of $O((T-1)nkd)$, dominated by the multiplication of the attention weights with the adjacency matrix within the *CSA* layer.

6 Evaluation

6.1 Transductive Tests

We evaluate our model's performance in a transductive setting using the proposed datasets, where propagation graphs in the training and test sets are sampled from the same distribution. We compare our model to popular traditional GNNs, specifically the Graph Convolutional Network (GCN) [10] and the Graph Attention Network (GAT) [28]. As these models were designed for static graphs, we utilized their temporal versions, which were obtained by injecting propagation states as sparse node features equivalent to the ones used by the CSA.

Additionally, we compare our model to a subset of the state-of-the-art temporal GNNs [16]. Among them, we evaluate a dynamic GNN [4] based on the Roland framework [33], which updates node embeddings hierarchically across snapshots. This model employs multiple GCN layers, each with subsequent updates along the snapshots using an MLP. We also evaluate the DySAT [24] model, which uses dedicated attention layers for both the structural and temporal dimensions of graphs. Both models are run with their default configurations as suggested by the authors. Because these models are not inherently designed for graph classification tasks, we post-process their outputs using mean aggregation of the node embeddings to yield a graph-level embedding, followed by a linear classification layer to map to the dataset classes. Table 1 displays the accuracy of the models on the different datasets. We chose accuracy instead of metrics such as F1-Score because the classes in our synthetic datasets are balanced.

6.2 Ablation Studies

We conduct ablation studies (see Table 2) to account for variations in the model configurations and architectures. CSA, Roland, and DySAT leverage only

Table 2: Ablations showing mean accuracy and standard deviation for each model and dataset. The best-performing values within each group are highlighted

Model	COV-19	FAKE	BTC-BP	DDoS	WAV
	(c=4)	(c=3)	(c=2)	(c=2)	(c=2)
GAT	**93.99 ± 2.09**	**86.59 ± 2.19**	68.08 ± 3.10	**81.65 ± 2.70**	**89.03 ± 2.85**
GAT_{NE}	92.95 ± 1.41	83.21 ± 2.17	67.60 ± 3.23	80.00 ± 2.59	87.90 ± 3.24
$\text{GAT}_{\text{NE, 1L}}$	84.30 ± 1.40	84.12 ± 1.11	**68.30 ± 0.52**	77.20 ± 0.25	80.55 ± 2.35
Roland	92.80 ± 0.57	78.91 ± 1.71	64.10 ± 7.89	59.80 ± 13.88	**81.30 ± 4.60**
$\text{Roland}_{\text{1L}}$	**93.67 ± 1.89**	**79.10 ± 1.32**	**67.00 ± 0.50**	**77.50 ± 3.97**	81.00 ± 3.97
CSA	**94.70 ± 0.45**	**82.49 ± 3.86**	**73.00 ± 2.15**	**83.80 ± 1.96**	89.20 ± 2.05
CSA_{NA}	89.50 ± 1.00	76.61 ± 0.86	66.50 ± 6.06	74.50 ± 10.68	**89.33 ± 2.02**

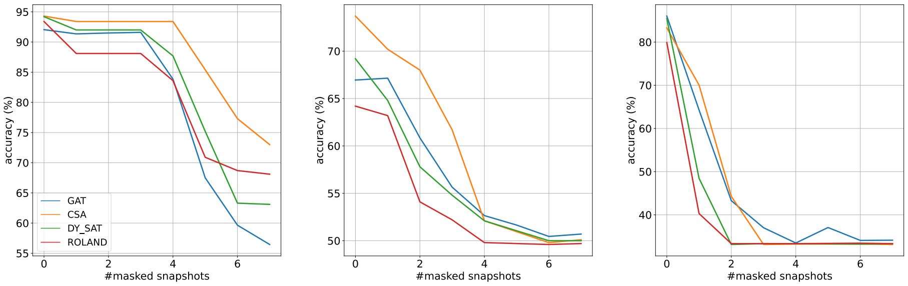
node features and the graph structure, ignoring propagation edge information, whereas the GAT implementation includes these features in its message passing. Additionally, CSA focuses solely on the direct neighborhood of nodes, while GAT performs three-hop and Roland two-hop message passing on the structure. Therefore, we also compare a GAT model without edge information (GAT_{NE}), a GAT model without edge information and with a single message passing layer ($\text{GAT}_{\text{NE, 1L}}$), and a Roland implementation with a single message passing layer ($\text{Roland}_{\text{1L}}$). To assess the efficacy of CSA's novel attention module, we finally report the performance of a model without its attention mechanism (CSA_{NA}), which simply assigns equal weight to all node embeddings when matching snapshot differences.

6.3 Inductive Tests

In our proposed synthetic datasets, all samples represent complete recordings from the onset of a propagation. However, in real-world scenarios, capturing the entire life-cycle of a propagation is often impossible. For instance, the spread of a virus might only be recognized after a significant number of infections. To simulate this, we delineate an experimental setup that masks propagation states of specific snapshots within the test set samples, exemplifying an inductive setting where training and test distributions diverge. Specifically, we mask an increasing number of consecutive snapshots from the beginning of the propagation. Instead of using a specific masking label, we substitute the propagation states of masked snapshots with the last recorded one. For a masked snapshot at time t, we reassign the state function $s(t)$ to $s(t-1)$, assuming the snapshot at time $t-1$ remains unmasked. Figure 1 illustrates the results of this masking experiment for a selected set of datasets.

6.4 Self-Supervised Tests

As a robustness evaluation, we evaluate the performance of CSA in scenarios where propagations cannot be accurately labeled. Take for instance, the indi-



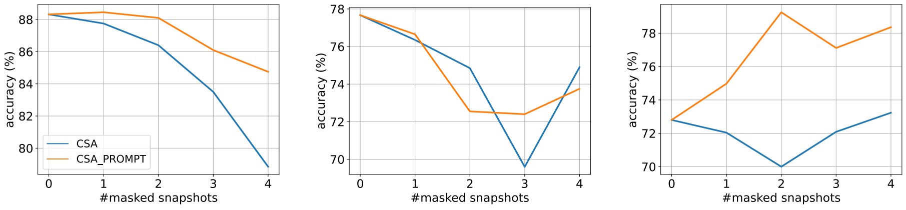
(a) COV-19 (b) BTC-BP (c) FAKE

Fig. 1: Accuracy trajectory of the models while masking an increasing number of snapshots within the test samples for COV-19, BTC-BP, and FAKE. The snapshots are consecutively masked from the beginning of propagations by substituting the propagation states with those from the last recorded snapshot.

viduals who have not been tested for a particular COVID-19 strain and, hence, their infection chains cannot be associated with a certain virus mutation. To counteract this potential information scarcity within propagation samples, we formulate an informative self-supervised pretext task, enabling GNN training without labeled data. This approach entails learning embeddings through self-supervision, which was shown [15] to elicit the recognition of latent patterns, thus yielding more robust embeddings.

As part of this experiment, we propose a predictive model termed the *prompting* model, which accepts a list of snapshots and aims to predict the propagation states of nodes at the next snapshot. The model incorporates an encoder-decoder architecture. An initial CSA layer encodes the preceding propagation snapshots, and the decoder reconstructs the resulting node embeddings using a lightweight version of the cross-snapshot attention utilizing the graph structure in the next snapshot. By training with a cross-entropy loss on the predicted binary propagation states, the model learns to represent propagation graphs in the embedding space, i.e., becomes capable of capturing the propagation behavior between subsequent snapshots.

The model is intentionally trained on a set of generic propagation graphs that do not align with the downstream datasets, tasked to recognize more general patterns among propagations, such as the causal-effect relationships. To evaluate this approach, we conduct another masking experiment. We use the pre-trained prompting model to fill the masked snapshots within the sample, before passing it into another CSA model serving as a classifier. In this case, the snapshots at the end of a propagation are masked in order to enact the task of predicting node states based on the initial propagation prompt. The results are shown in Figure 2.



(a) COV-19 (b) WAV (c) FAKE

Fig. 2: Comparison of CSA and a *prompting*-based counterpart that uses an upstream prompting model to recover maskings. Snapshots within the test set of COV-19, WAV, and FAKE are masked from the end of the propagations.

7 Discussion

7.1 Implications

Transductive Tests. CSA significantly outperforms the temporally-enhanced GCN and GAT, as well as the more modern Roland. However, DySAT surpasses CSA on FAKE and DDoS datasets, but underperforms on BTC-BP (compare Table 1). These results demonstrate that CSA is competitive with the state-of-the-art solutions, while suggesting that CSA has the potential of improving prediction accuracy in certain propagation scenarios.

Ablation Studies. The performance decrease of GAT when ignoring edge features, shown in Table 2, suggests that including such information in CSA could further enhance its accuracy. Notably, additional message passing layers are decisive for GAT performance. Conversely, removing layers improves accuracy of Roland. With a single layer, Roland is guided to update embeddings solely based on direct neighbors, proving more effective within the propagation context. In contrast, it is crucial for GAT to combine embeddings from distant nodes, as it otherwise fails to match states from remote snapshots due to operating on a static graph. Finally, removing the novel attention mechanism within the CSA layer leads to a significant drop in performance, underscoring the importance of the attention in identifying meaningful temporal patterns among nodes and snapshots.

Inductive Tests. Masking snapshots at the start of propagations results in severe performance degradation across models and datasets (see Figure 1). However, while GAT, Roland, and DySAT degrade similarly with the number of masked snapshots, CSA exhibits a smoother degradation and withstands one additional masking. This finding suggests that CSA is able to produce embeddings that are more resilient to missing information, which is remarkable given that gathering continuous snapshots throughout the entire observation may be infeasible in the propagation domain, e.g., consider non-symptomatic infections.

Self-Supervised Tests. When recovering information from masked snapshots using the predictive prompting model before classification, Figure 2 illustrates enhanced accuracy on COV-19. Interestingly, CSA is insensitive to maskings on FAKE, while the prompting even increases accuracy. This result potentially positions prompting as a viable alternative to mitigate information scarcity in propagations. It presents an interesting trade-off between the cost and feasibility of self-supervision versus saving resources during data collection.

7.2 Threats to Validity

External validity. While we demonstrate solid performance of CSA in transductive and inducting environments, the instances evaluated might not be representative of realistic domains. We mitigated this threat by sampling diverse scenarios derived from each domain. However, future work is needed to evaluate if our findings can be generalized to real-world settings. Because we evaluate CSA with a subset set of baselines, further experiments are recommended, for instance, with GNNs of the event-based category [9] and graph kernels [21]. Note that the architecture of CSA allows adaptation to event-based networks by transforming the lists of complete graph snapshots into sequences of events.

Internal validity. During evaluation, we did not investigate potential biases induced by hidden confounders. In future studies, we plan to examine whether the attention mechanism learns significant patterns during training that benefit the classification task. Additionally, further investigation is required to discern the causal-effects behind the robustness increase when recovering snapshots. Specifically, we need to determine if this robustness is due to the model accurately predicting snapshots or if it is merely the result of additional noise in the data.

8 Conclusion and Future Work

This work presents a novel cross-snapshot method that utilizes element-wise attention weights that measure the causal-effect relationships within propagations, by associating network regions with changes in the node states. Our approach enables the classification of propagation patterns in complex dynamic graphs. We reproduce these patterns by synthesizing datasets that mirror real-world network dynamics. Results are promising in a sense that we outperform traditional approaches and remain competitive with two state-of-the-art methods.

In future work, we plan to investigate the robustness of the method and the representativity of the synthetic dataset for cases where CSA underperforms. To verify the increased robustness observed in CSA, we plan to generate more sophisticated maskings that may produce more severe perturbations to the snapshots. Beyond the classification task, we aim to study the CSA application in a broader spectrum of tasks, such as transfer learning and fine-tuning across domains by combining real and synthetic datasets.

References

1. Alcami, A., Koszinowski, U.H.: Viral mechanisms of immune evasion. Trends in microbiology **8**(9), 410–418 (2000)
2. Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder–decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1724–1734 (2014)
3. Decker, C., Wattenhofer, R.: Information propagation in the bitcoin network. In: IEEE P2P 2013 Proceedings. pp. 1–10. IEEE (2013)
4. Dileo, M., Zignani, M., Gaito, S.: Temporal graph learning for dynamic link prediction with text in online social networks. Machine Learning **113**, 2207–2226 (2024)
5. Fan, Y., Yao, Y., Joe-Wong, C.: Gcn-se: Attention as explainability for node classification in dynamic graphs. Industrial Conference on Data Mining (2021)
6. Gomez-Rodriguez, M., Song, L., Du, N., Zha, H., Schölkopf, B.: Influence estimation and maximization in continuous-time diffusion networks. ACM Transactions on Information Systems (TOIS) **34**(2), 1–33 (2016)
7. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**, 1735–80 (12 1997)
8. Huang, S., Poursafaei, F., Danovitch, J., Fey, M., Hu, W., Rossi, E., Leskovec, J., Bronstein, M.M., Rabusseau, G., Rabbany, R.: Temporal Graph Benchmark for Machine Learning on Temporal Graphs. In: Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track (2023)
9. Kazemi, S.M., Goel, R., Jain, K., Kobyzev, I., Sethi, A., Forsyth, P., Poupart, P.: Representation learning for dynamic graphs: A survey. The Journal of Machine Learning Research (2020)
10. Kipf, T.N., Welling, M.: Semi-Supervised Classification with Graph Convolutional Networks. In: ICLR (2017)
11. La Gatta, V., Moscato, V., Postiglione, M., Sperli, G.: An epidemiological neural network exploiting dynamic graph structured data applied to the covid-19 outbreak. IEEE Transactions on Big Data **7**(1), 45–55 (2020)
12. Leung, N.H.: Transmissibility and transmission of respiratory viruses. Nature Reviews Microbiology **19**(8), 528–545 (2021)
13. Li, Y., Gu, C., Dullien, T., Vinyals, O., Kohli, P.: Graph matching networks for learning the similarity of graph structured objects. International Conference on Machine Learning (2019)
14. Liu, Y., Rocklöv, J.: The effective reproductive number of the omicron variant of sars-cov-2 is several times relative to delta. Journal of Travel Medicine **29** (2022)
15. Liu, Y., Jin, M., Pan, S., Zhou, C., Zheng, Y., Xia, F., Yu, P.: Graph self-supervised learning: A survey. IEEE Transactions on Knowledge and Data Engineering (2022)
16. Longa, A., Lachi, V., Santin, G., Bianchini, M., Lepri, B., Lio, P., Scarselli, F., Passerini, A.: Graph neural networks for temporal graphs: State of the art, open challenges, and opportunities. arXiv (2023)
17. Manessi, F., Rozza, A., Manzo, M.: Dynamic graph convolutional networks. Pattern Recognition **97**, 107000 (2020)
18. Marisca, I., Cini, A., Alippi, C.: Learning to reconstruct missing data from spatiotemporal graphs with sparse observations. Advances in Neural Information Processing Systems **35**, 32069–32082 (2022)
19. Michail, O.: An introduction to temporal graphs: An algorithmic perspective (2015)

20. Nikolentzos, G., Siglidis, G., Vazirgiannis, M.: Graph kernels: A survey. Journal of Artificial Intelligence Research **72**, 943–1027 (2021)
21. Oettershagen, L., Kriege, N.M., Jordan, C., Mutze, P.: A temporal graphlet kernel for classifying dissemination in evolving networks. In: Proceedings of the 2023 SIAM International Conference on Data Mining (SDM). pp. 19–27. SIAM (2023)
22. Pareja, A., Domeniconi, G., Chen, J., Ma, T., Suzumura, T., Kanezashi, H., Kaler, T., Schardl, T.B., Leiserson, C.E.: Evolvegcn: Evolving graph convolutional networks for dynamic graphs. AAAI Conference on Artificial Intelligence (2019)
23. 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)
24. Sankar, A., Wu, Y., Gou, L., Zhang, W., Yang, H.: Dysat: Deep neural representation learning on dynamic graphs via self-attention networks. In: Proceedings of the 13th international conference on web search and data mining. pp. 519–527 (2020)
25. Shahsavari, Y., Zhang, K., Talhi, C.: A theoretical model for block propagation analysis in bitcoin network. IEEE Transactions on Engineering Management **69**(4), 1459–1476 (2020)
26. Taheri, A., Gimpel, K., Berger-Wolf, T.: Learning to represent the evolution of dynamic graphs with recurrent models. In: Companion Proceedings of The 2019 World Wide Web Conference. p. 301–307. WWW '19, Association for Computing Machinery, New York, NY, USA (2019)
27. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems (2017)
28. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. International Conference on Learning Representations (2017)
29. Vosoughi, S., Roy, D., Aral, S.: The spread of true and false news online. science **359**(6380), 1146–1151 (2018)
30. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature **393**(6684), 440–442 (1998)
31. Wu, Y., Kang, L., Guo, Z., Liu, J., Liu, M., Liang, W.: Incubation period of covid-19 caused by unique sars-cov-2 strains: a systematic review and meta-analysis. JAMA network open **5**(8) (2022)
32. Yang, Z., Zhang, S., Tang, Y.P., Zhang, S., Xu, D.Q., Yue, S.J., Liu, Q.L.: Clinical characteristics, transmissibility, pathogenicity, susceptible populations, and re-infectivity of prominent covid-19 variants. Aging and disease **13**(2), 402 (2022)
33. You, J., Du, T., Leskovec, J.: Roland: Graph learning framework for dynamic graphs. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. p. 2358–2366 (2022)
34. Yu, S., Xia, F., Li, S., Hou, M., Sheng, Q.Z.: Spatio-temporal graph learning for epidemic prediction. ACM Transactions on Intelligent Systems and Technology **14**(2), 1–25 (2023)
35. Zhao, Z., Zhao, J., Sano, Y., Levy, O., Takayasu, H., Takayasu, M., Li, D., Wu, J., Havlin, S.: Fake news propagates differently from real news even at early stages of spreading. EPJ data science **9**(1), 7 (2020)
36. Zhong, Y., Huang, C.: A dynamic graph representation learning based on temporal graph transformer. Alexandria Engineering Journal **63**, 359–369 (2023)
37. Zhou, B., Liu, X., Liu, Y., Huang, Y., Lio, P., Wang, Y.G.: Well-conditioned spectral transforms for dynamic graph representation. In: Learning on Graphs Conference. pp. 12–1. PMLR (2022)