Valid Conformal Prediction for Dynamic GNNs

Ed Davis[1] **Ian Gallagher**[1] **Daniel John Lawson**[1] **Patrick Rubin-Delanchy**[2]
[1]University of Bristol, U.K. [2]The University of Edinburgh, U.K.
{edward.davis, ian.gallagher, dan.lawson}@bristol.ac.uk,
prubind@ed.ac.uk

Abstract

Graph neural networks (GNNs) are powerful black-box models which have shown impressive empirical performance. However, without any form of uncertainty quantification, it can be difficult to trust such models in high-risk scenarios. Conformal prediction aims to address this problem, however, an assumption of exchangeability is required for its validity which has limited its applicability to static graphs and transductive regimes.

We propose to use unfolding, which allows any existing static GNN to output a dynamic graph embedding with exchangeability properties. Using this, we extend the validity of conformal prediction to dynamic GNNs in both transductive and semi-inductive regimes. We provide a theoretical guarantee of valid conformal prediction in these cases and demonstrate the empirical validity, as well as the performance gains, of unfolded GNNs against standard GNN architectures on both simulated and real datasets.

1 Introduction

Graph neural networks (GNNs) have seen formidable success in a wide variety of application domains including prediction of protein structure [1] and molecular properties [2, 3], drug discovery [4], computer vision [5], natural language processing [6, 7], recommendation systems [8–10], estimating time of arrival (ETA) in services like Google Maps [11], and advancing mathematics [12]. They often top leaderboards in benchmarks relating to machine-learning on graphs [13], for a range of tasks such as node, edge, or graph property prediction. Useful resources for getting started with GNNs include the introductions [14, 15] and the Pytorch Geometric library [16].

Conformal prediction, advanced by Vovk and co-authors [17], and later expanded on by various researchers in the Statistics community [18–26], is an increasingly popular paradigm for constructing provably valid prediction sets from a 'black-box' algorithm with minimal assumptions and at low cost. Several recent papers have brought this idea to quantifying uncertainty in GNNs [27–29].

There are many applications of significant societal importance, e.g. cyber-security [30–32], human-trafficking prevention [33, 34], social media & misinformation [35], contact-tracing [36], patient care [37, 38] where we would like to be able to use GNNs, with uncertainty quantification, on dynamic graphs.

The added time dimension in dynamic graphs causes serious issues with existing GNN + conformal methodology, broadly relating to de-alignment between embeddings across time points (see Figure 1), resulting in large or invalid prediction sets in even simple and stable dynamic regimes.

Inspired by a concurrent line of statistical research on spectral embedding and tensor decomposition [39–45], we propose to use one of the *unfoldings* [46] of the tensor representation of the dynamic graph as input to a standard GNN, and demonstrate validity in both transductive and semi-inductive regimes.

Preprint. Under review.

Related work. The only paper we found studying conformal inference on dynamic graphs is [29], but the paper considers a strictly growing graph, which is not what we mean by a 'dynamic graph', where edges appear and disappear in complex and random ways, as in all of our examples and applications alluded to above. Our work naturally builds on earlier contributions combining GNNs and conformal inference [27–29] but our theory is distinct from those contributions, with arguments closer to [24], and the use of unfoldings for GNNs is new. In the dynamic graph literature, unfoldings have exclusively been proposed for unsupervised settings [43–45, 47, 48], particularly spectral embedding, and have never been applied to conformal inference.

2 Theory & Methods

Problem setup. In this paper, a dynamic graph G is a sequence of T graphs over a globally defined nodeset $[n] = \{1, \ldots, n\}$, represented by adjacency matrices $\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(T)}$. It may help to assume the $\mathbf{A}^{(t)}$ are $n \times n$ binary symmetric matrices corresponding to undirected graphs, but in fact all we require is that the $\mathbf{A}^{(t)}$ have the same number of rows, n_r, allowing for directed, bipartite and weighted graphs.

Along with the dynamic graph G, we are given labels for certain nodes at certain points in time, which could describe a state or behaviour. Our task is to predict the label for a different node/time pair, whose label is hidden or missing. Our prediction must take the form of a set which we can guarantee contains the true label with some pre-specified probability $1 - \alpha$ (e.g. 95%).

We track the labelled and unlabelled pairs in this problem using a fixed sequence ν where for each $(i, t) \in \nu$ we assume existence of

1. a class label $Y_i^{(t)} \in \mathcal{Y}$, which may or may not be observed, where $\mathcal{Y}$ is some discrete set of cardinality d;
2. a set of attributes, which is observed, and represented by a feature vector $X_i^{(t)} \in \mathbb{R}^p$;
3. a corresponding column in $\mathbf{A}^{(t)}$, denoted $\mathbf{A}_i^{(t)}$.

For a pair $w = (i, t) \in \nu$, we will use the notation $X_w \coloneqq X_i^{(t)}$, $Y_w \coloneqq Y_i^{(t)}$, $\mathbf{A}_w \coloneqq \mathbf{A}_i^{(t)}$.

Transductive regime. For some $m < |\nu|$, the missing label is missing completely at random among the first $m + 1$ node/time pairs of ν.

Semi-inductive regime. For some $m < |\nu|$, the missing label corresponds to a fixed pair among the first $m + 1$ node/time pairs of ν.

Let ν_{test} denote the missing node/time pair, $\nu^{(\text{training})} \coloneqq \{\nu_\ell, \ell > m + 1\}$, and $\nu^{(\text{calibration})} \coloneqq \{\nu_\ell, \ell \leq m + 1\} \setminus \{\nu_{\text{test}}\}$.

In the transductive regime, a valid confidence set can be constructed with no further assumptions (Algorithm 1). This may seem surprising and could be very useful in view of the following implication: the pairs $\nu^{(\text{training})}$ can correspond to historical datapoints, at times $t < T$, and the pairs $\{\nu_{\text{test}}\} \cup \nu^{(\text{calibration})}$ can correspond to datapoints in the present, T. We only need the missing label to be uniformly chosen *among the labels at time* T. The semi-inductive regime is more challenging, and could reflect a scenario where we have only historical labels. Here, we will need to invoke further assumptions such as symmetry and exchangeability, familiar in the conformal prediction literature.

Proposed approach. We will repurpose a standard GNN, designed for static graphs, into producing dynamic graph embeddings with desirable exchangeability properties. Let $\mathcal{G}$ denote a GNN taking as input an adjacency matrix $\mathbf{A}$ corresponding to an undirected graph on N nodes, along with node attributes and labels (some missing), and returning an *embedding* $\hat{\mathbf{Y}} \in \mathbb{R}^{N \times d}$. Typically, the ith row of $\hat{\mathbf{Y}}$ predicts the label of node i through

$$\hat{\mathbb{P}}(\text{node } i \text{ has label } k) = \left[\text{softmax}(\hat{\mathbf{Y}}_{i1}, \ldots, \hat{\mathbf{Y}}_{id})\right]_k .$$

In place of this generic input $\mathbf{A}$, we will enter the dilated unfolding [47] corresponding to the dynamic graph G,

$$\mathbf{A}^{\text{UNF}} \coloneqq \begin{bmatrix} \mathbf{0} & \boldsymbol{\mathcal{A}} \\ \boldsymbol{\mathcal{A}}^\top & \mathbf{0} \end{bmatrix},$$

where $\boldsymbol{\mathcal{A}} = \left(\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(T)}\right)$ (column-concatenation), along with the attributes $X_w, w \in \nu$ and training labels $Y_w, w \in \nu^{(\text{training})}$.

The resulting output from the GNN splits into embeddings

$$\begin{bmatrix} \hat{\mathbf{X}}^{\text{UNF}} \\ \hat{\mathbf{Y}}^{\text{UNF}} \end{bmatrix} \coloneqq \boldsymbol{\mathcal{G}}\left(\mathbf{A}^{\text{UNF}}\right),$$

where $\hat{\mathbf{X}}^{\text{UNF}} \in \mathbb{R}^{n_r \times d}$ contains global representations. The embedding $\hat{\mathbf{Y}}^{\text{UNF}}$ contains representations of node/time pairs, of principal interest here, and we let $\hat{\mathbf{Y}}_w^{\text{UNF}}$ denote the row representing node/pair $w \in \nu$.

Finally, we shall require some non-conformity score function $r : \mathbb{R}^d \times \mathcal{Y} \to \mathbb{R}$, with the convention that $r(\hat{y}, y)$ is large when $\hat{y}$ is a poor prediction of y. In practice, we make use of the adaptive non-conformity measures detailed by [26].

With these ingredients in place, we are in a position to obtain a prediction set $\hat{C}$ for ν_{test}, the computation of which is detailed in Algorithm 1. The notation leaves implicit the dependence of the set $\hat{C}$ on the observed data; explicitly, the set $\hat{C}$ is a deterministic function of the dynamic graph G, the attributes $(X_w, w \in \nu)$, and the labels $(Y_w, w \in \nu \setminus \{\nu_{\text{test}}\})$.

Algorithm 1 Split conformal inference

Input: Dynamic graph G, node/time pairs $\nu = \nu^{(\text{training})} \cup \nu^{(\text{calibration})} \cup \{\nu_{\text{test}}\}$, attributes $(X_w, w \in \nu)$, labels $(Y_w, w \in \nu \setminus \{\nu_{\text{test}}\})$, confidence level α, non-conformity function r

1: Learn $\hat{\mathbf{Y}}^{\text{UNF}}$ based on $\nu^{(\text{training})}$
2: Let $\hat{C} = \mathcal{Y}$
3: Let $R_w = r(\hat{\mathbf{Y}}_w^{\text{UNF}}, Y_w)$ for $w \in \nu^{(\text{calibration})}$
4: Let
$$\hat{q} = \lfloor \alpha(m+1) \rfloor \text{ largest value in } R_w, w \in \nu^{(\text{calibration})}$$
5: **for** $y \in \mathcal{Y}$ **do**
6: Let $R_{\text{test}} = r(\hat{\mathbf{Y}}_{\nu_{\text{test}}}^{\text{UNF}}, y)$
7: Remove y from $\hat{C}$ if $R_{\text{test}} \geq \hat{q}$
8: **end for**

Output: Prediction set $\hat{C}$

Theory. As mentioned earlier, the transductive regime requires no further assumptions.

Lemma 1. *In the transductive regime, the prediction set output by Algorithm 1 is valid, that is,*

$$\mathbb{P}(Y_{\nu_{\text{test}}} \in \hat{C}) \geq 1 - \alpha.$$

Let $\nu^+_{\text{calibration}} = (\nu_\ell, \ell \in [m+1])$, comprising $\nu_{\text{calibration}}$ and ν_{test}. For the semi-inductive regime, we make the following assumptions:

A1. *The columns $\mathbf{A}_w$, attributes X_w, and labels Y_w corresponding to $w \in \nu^+_{\text{calibration}}$ are jointly exchangeable.*

We make the definition of exchangeability precise in the supplementary material. Informally, we can swap columns of $\mathcal{A} = \left(\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(T)}\right)$ corresponding to test or calibration pairs, along with the attributes and labels, without changing the likelihood of the data.

A2. *The GNN $\mathcal{G}$ is label equivariant.*

This definition is again made precise in the supplementary material. Informally, if we re-order the nodes, attributes and labels, we re-order the resulting embedding.

Theorem 1. *Under assumptions 1 and 2, the prediction set output by Algorithm 1 is valid in the semi-inductive regime, that is,*

$$\mathbb{P}(Y_{\nu_{\text{test}}} \in \hat{C}) \geq 1 - \alpha.$$

Discussion. Assumption 2 is relatively mild and roughly satisfied by standard GNN architectures. Assumption 1 is more challenging, and can guide us towards curating a calibration set. For example, if we hypothesise that the $(\mathbf{A}_i^{(t)}, t \in [T])$ are i.i.d. stationary stochastic processes (over time), then we can satisfy Assumption 1 by ensuring $\nu^+_{\text{calibration}}$ contains only distinct nodes. The independence hypothesis will typically break in undirected graphs because of the necessary symmetry of each $\mathbf{A}^{(t)}$, and in this case we may also wish to ensure that $\nu^+_{\text{calibration}}$ contains only distinct time points. In general, for predicting the label of nodes based on historical information, Assumption 1 should make us wary of global drift, a poor distributional overlap between the test and calibration nodes, and autocorrelation. The semi-inductive regime is where our theory actively requires a dilated unfolding, and where alternative approaches can fail completely; however, even in the transductive regime where our theory covers alternative approaches, the prediction sets from unfolded GNNs (UGNNs) are often smaller.

2.1 A Visual Motivation: i.i.d. Draws of a Stochastic Block Model

Consider a simple two-community dynamic stochastic block model (DSBM) [49, 50] for undirected graphs. Let

$$\mathbf{B}^{(1)} = \mathbf{B}^{(2)} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.9 \end{bmatrix}, \tag{1}$$

be matrices of inter-community edge probabilities, and let $\tau \in \{1, 2\}^n$ be a community allocation vector. We then draw each symmetric adjacency matrix point as $\mathbf{A}^{(1)}_{ij} \overset{\text{ind}}{\sim} \text{Bernoulli}(\mathbf{B}^{(1)}_{\tau_i,\tau_j})$ and $\mathbf{A}^{(2)}_{ij} \overset{\text{ind}}{\sim} \text{Bernoulli}(\mathbf{B}^{(2)}_{\tau_i,\tau_j})$, for $i \leq j$.

A typical way of applying a GNN to a series of graphs is to first stack the adjacency matrices as blocks on the diagonal [51],

$$\mathbf{A}^{\text{BD}} = \begin{bmatrix} \mathbf{A}^{(1)} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{A}^{(T)} \end{bmatrix},$$

from which a dynamic graph embedding is obtained through

$$\hat{\mathbf{Y}}_{\text{BD}} = \mathcal{G}\left(\mathbf{A}^{\text{BD}}\right).$$

Figure 1 plots the representations, $\hat{\mathbf{Y}}_{\text{BD}}$ and $\hat{\mathbf{Y}}_{\text{UNF}}$ of $\mathbf{A}^{(1)}, \mathbf{A}^{(2)}$ using a GCN [52] as $\mathcal{G}$. It is clear that, despite the fact that $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$ come from the same distribution, the output of the block diagonal GCN appears to encode a change that is not present. We refer to this issue as *temporal shift*. In contrast, the output of UGCN is exchangeable over time and so features no temporal shift.

This example highlights two advantages of unfolded GNN over the standard GNN case. First, as there is no temporal shift, we would expect it to show improved predictive power. Second, exchangeability over time allows for the application of conformal inference across time.

3 Experiments

We evaluate the performance of UGNNs using four examples, comprising simulated and real data, summarised in Table 1. We will then delve deeper into a particular dataset to show how variation in prediction sets can tell us something about the underlying network dynamics. Figure 2 displays the number of edges in each of the considered datasets over time. The SBM, school and flight data each feature abrupt changes in structure, while the trade data is relatively smooth.

SBM. A three-community DSBM with inter-community edge probability matrix

$$\mathbf{B}^{(t)} = \begin{bmatrix} s_1 & 0.02 & 0.02 \\ 0.02 & s_2 & 0.02 \\ 0.02 & 0.02 & s_3 \end{bmatrix},$$

where s_1, s_2, and s_3 represent within-community connection states. Each s can be one of two values: 0.08 or 0.16. We simulate a dynamic network over $T = 8$ time points, corresponding to the $8 = 2^3$

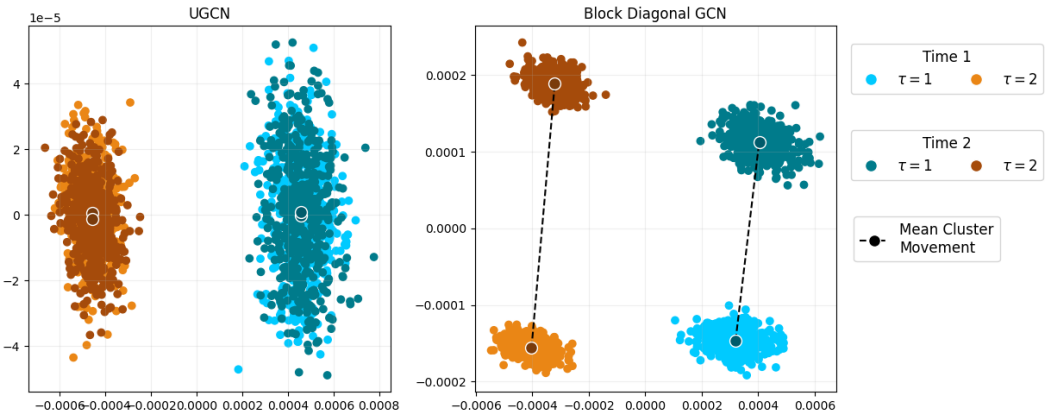


Figure 1: UGCN and block diagonal GCN representations of an i.i.d. stochastic block model after applying PCA. The models were trained with transductive masks. Block diagonal GCN appears to encode a significant change over time despite there being none. The embedding from UGCN is exchangeable over time, as would be expected.

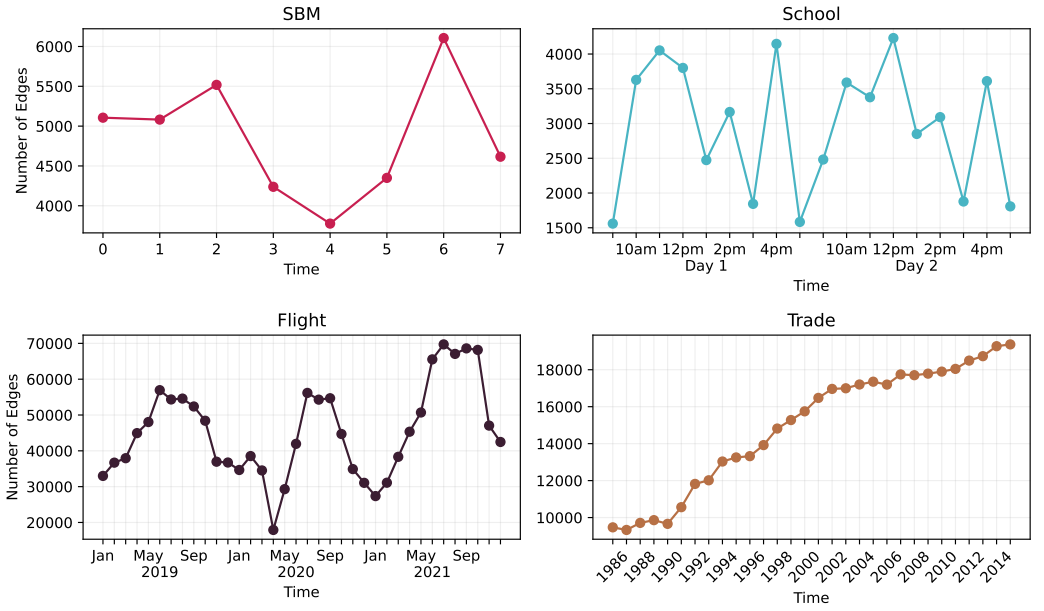


Figure 2: Numbers of edges over time. The School and Flight data show rough periodic/seasonal structure, which the Trade data features drift with the number of edges growing smoothly with time.

possible combinations, drawing each $\mathbf{A}^{(t)}$ from each unique $\mathbf{B}^{(t)}$ as detailed in Section 2.1. The task is to predict the community label of each node.

School. A dynamic social network between pupils at a primary school in Lyon, France [53]. Each of the 232 pupils wore a radio identification device such that each interaction, with its timestamp, could be recorded, forming a dynamic network. An interaction was defined by close proximity for 20 seconds. The task here is to predict the classroom allocation of each pupil. This dataset has temporal structure particularly distinguishing class time, where pupils cluster together based on their class (easier), and lunchtime, where the cluster structure breaks down (harder). The data covers two full school days, making it roughly repeating.

Flight. The OpenSky dataset tracks the number of flights (edges) between airports (nodes) over each month from the start of 2019 to the end of 2021 [54]. The task is to predict the country of a given

(European only) airport. The network exhibits seasonal and periodic patterns, and features a change in structure when the COVID-19 pandemic hit Europe around March 2020.

Trade. An agricultural trade network between members of the United Nations tracked yearly between 1986 and 2016 [55], which features in the Temporal Graph Benchmark [27]. The network is directed and edges are weighted by total trade value. Unlike the other examples, this network exhibits important drift over time. This behaviour is visualised in Figure 2. We consider the goal of predicting the top trade partner for each nation for the next year, a slight deviation from the benchmark where performance is quantified by the Normalized Discounted Cumulative Gain of the top 10 ranked trade partners (less naturally converted into a conformal prediction problem).

Dataset	Nodes	Edges	Time points	Classes	Weighted	Directed	Drift
SBM	300	38,350	8	3	No	No	No
School	232	53,172	18	10	No	No	No
Flight	2,646	1,635,070	36	46	Yes	No	No
Trade	255	468,245	32	163	Yes	Yes	Yes

Table 1: A summary of considered datasets.

3.1 Experimental Setup

On each dataset, we apply GCN [52] and GAT [56] to the block diagonal and unfolded matrix structures (referred to as UGCN and UGAT respectively) in both the transductive and the semi-inductive settings. The datasets considered do not have attributes. In the transductive regime, we randomly assign nodes to train, validation, calibration and test sets with ratios 20/10/35/35, regardless of their time point label. Due to the high computational cost of fitting multiple GNNs to quantify randomisation error, we follow [57] by fitting the GNN 10 times, then constructing 100 different permutations of calibration and test sets to allow 1,000 conformal inference instances per model and dataset, and apply Algorithm 2.

For the semi-inductive regime, we apply a similar approach, except that we reserve the *last* 35% of the observation period as the test set, rounded to use an integer number of time points for testing to ensure every node/time pair in the test set is unlabelled. The training, validation and calibration sets are then picked at random, regardless of time point label, from the remaining data with ratios 20/10/35. As the test set is fixed, efficiency saving is not possible and we instead run Algorithm 1 (split conformal) on 50 random data splits.

Prediction sets are computed using the Adaptive Prediction Sets (APS) algorithm [26] in both regimes. For each experiment, we return the mean accuracy, coverage and prediction set size across all conformal runs to evaluate the predictive power of each GNN, as well as its conformal performance. To quantify error, we quote the standard deviation of each metric. We additionally consider an empirical evaluation of conditional coverage, which is desirable but known to be theoretically impossible to guarantee [58]. We empirically estimate this property by returning the worst-case coverage across time for each of our examples [59].

Code to reproduce experiments can be found in Appendix A.

3.2 Results

UGNN has higher accuracy in every semi-inductive example and most transductive examples. In the transductive regime such advantages are often minor, however in the semi-inductive regime UGNN displays a more significant advantage (Table 2). Block GNN is close to random guessing in the semi-inductive SBM and School examples (having 3 and 10 classes respectively). In contrast, UGNN maintains a strong accuracy in both regimes.

Conformal prediction on UGNN and block GNN is empirically valid for every transductive example. This confirms (Lemma 1) that exchangeability/label equivariance are not necessary for valid conformal prediction in the transductive regime, as even block GNN (which has no exchangeability properties) is valid here (Table 3). However, UGNN achieves valid coverage with smaller prediction sets in most cases, making it the preferable method (Table 4).

Methods	SBM		School	
	Trans.	Semi-ind.	Trans.	Semi-ind.
Block GCN	0.964 ± 0.011	0.334 ± 0.024	0.856 ± 0.011	0.116 ± 0.011
UGCN	$\mathbf{0.980 \pm 0.004}$	$\mathbf{0.985 \pm 0.003}$	$\mathbf{0.924 \pm 0.009}$	$\mathbf{0.915 \pm 0.013}$
Block GAT	0.916 ± 0.028	0.346 ± 0.024	0.807 ± 0.016	0.107 ± 0.024
UGAT	$\mathbf{0.947 \pm 0.032}$	$\mathbf{0.969 \pm 0.017}$	$\mathbf{0.896 \pm 0.016}$	$\mathbf{0.868 \pm 0.017}$

Methods	Flight		Trade	
	Trans.	Semi-ind.	Trans.	Semi-ind.
Block GCN	0.405 ± 0.075	0.121 ± 0.059	$\mathbf{0.095 \pm 0.017}$	0.049 ± 0.018
UGCN	$\mathbf{0.441 \pm 0.069}$	$\mathbf{0.477 \pm 0.061}$	0.082 ± 0.031	$\mathbf{0.050 \pm 0.017}$
Block GAT	$\mathbf{0.417 \pm 0.031}$	0.114 ± 0.061	$\mathbf{0.125 \pm 0.017}$	0.046 ± 0.015
UGAT	0.408 ± 0.060	$\mathbf{0.427 \pm 0.061}$	0.111 ± 0.010	$\mathbf{0.048 \pm 0.015}$

Table 2: Accuracy (higher is better) for 2 GNNs (GCN or GAT) under 2 representations (block diagonal adjacency or unfolding) for 4 datasets. Bold values indicate the highest accuracy for a given GNN/representation pair.

Methods	SBM		School	
	Trans.	Semi-ind.	Trans.	Semi-ind.
Block GCN	$\mathbf{0.901 \pm 0.014}$	0.659 ± 0.045	$\mathbf{0.901 \pm 0.012}$	0.812 ± 0.033
UGCN	$\mathbf{0.901 \pm 0.015}$	$\mathbf{0.918 \pm 0.025}$	$\mathbf{0.901 \pm 0.012}$	$\mathbf{0.924 \pm 0.013}$
Block GAT	$\mathbf{0.901 \pm 0.015}$	0.450 ± 0.154	$\mathbf{0.901 \pm 0.012}$	0.662 ± 0.084
UGAT	$\mathbf{0.901 \pm 0.014}$	$\mathbf{0.914 \pm 0.022}$	$\mathbf{0.901 \pm 0.012}$	$\mathbf{0.909 \pm 0.021}$

Methods	Flight		Trade	
	Trans.	Semi-ind.	Trans.	Semi-ind.
Block GCN	$\mathbf{0.900 \pm 0.002}$	0.853 ± 0.013	$\mathbf{0.900 \pm 0.009}$	0.842 ± 0.015
UGCN	$\mathbf{0.900 \pm 0.002}$	$\mathbf{0.910 \pm 0.003}$	$\mathbf{0.900 \pm 0.009}$	0.847 ± 0.017
Block GAT	$\mathbf{0.900 \pm 0.002}$	0.862 ± 0.013	$\mathbf{0.900 \pm 0.009}$	0.840 ± 0.021
UGAT	$\mathbf{0.900 \pm 0.002}$	$\mathbf{0.906 \pm 0.002}$	$\mathbf{0.901 \pm 0.009}$	0.854 ± 0.023

Table 3: Coverage (targetted to ≥ 0.9) for 2 GNNs (GCN or GAT) under 2 representations (block diagonal adjacency or unfolding) for 4 datasets. Bold values indicate valid coverage with target ≥ 0.9.

Conformal prediction on UGNN is empirically valid for every semi-inductive example without drift, while block GNN is valid for none. In all examples, except for the trade dataset, UGNN produces valid conformal (Table 3) with similar prediction set sizes to the transductive case (Table 4).

We include the trade data as an example of where UGNN fails to achieve valid coverage in the semi-inductive regime. This failure is due to the drift present in the trade network (Figure 2), which grows over time (approximately doubling in edges over the whole period). Therefore, the network at the start of this series is not approximately exchangeable with the network at the end of the series.

Neither method achieves conditional coverage. In Appendix C we show that (as anticipated), the worst-case conditional coverage across all time points for both methods is less than the target coverage, both for block and unfolded matrices and both for GCN and GATs. In most semi-inductive regimes, UGNN is closer to this target, but they perform similarly in the transductive regime here.

The problems of drift and conditional coverage are difficult to handle inside an unfolded GNN, but we hypothesise that downstream methods could be applied to improve coverage in these cases, for example [24] and [28]. We leave this investigation for future work.

Due to the unfolded matrix having double the entries of the block diagonal matrix, the computation times for UGNN were roughly double that of block GNN. The maximum time to train an individual model was around a minute on an AMD Ryzen 5 3600 CPU processor. A full run of experiments on the largest dataset took around 4.5 hours.

3.3 Temporal Analysis

The goal of conformal prediction is to provide a notion of uncertainty to an otherwise black-box model. As a problem becomes more difficult, this should be reflected in a larger prediction set size to maintain target coverage. We analyse this behaviour by focusing on the school data example.

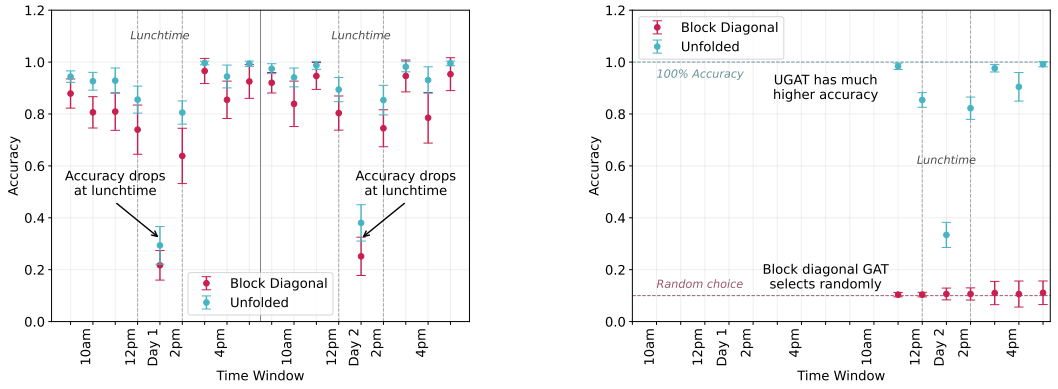
Methods	SBM		School	
	Trans.	Semi-ind.	Trans.	Semi-ind.
Block GCN	**1.258 ± 0.053**	*1.977 ± 0.138*	4.542 ± 0.167	*8.079 ± 0.188*
UGCN	1.263 ± 0.206	**1.097 ± 0.171**	**2.763 ± 0.311**	**3.037 ± 0.251**
Block GAT	1.063 ± 0.180	*1.320 ± 0.466*	3.863 ± 0.813	*6.540 ± 0.830*
UGAT	**1.053 ± 0.249**	**1.042 ± 0.201**	**3.552 ± 0.756**	**4.185 ± 1.228**

Methods	Flight		Trade	
	Trans.	Semi-ind.	Trans.	Semi-ind.
Block GCN	24.116 ± 2.711	*23.283 ± 2.285*	**82.556 ± 4.966**	***80.652 ± 6.709***
UGCN	**22.369 ± 1.953**	**23.030 ± 2.115**	86.319 ± 7.520	*85.006 ± 9.516*
Block GAT	25.173 ± 1.631	*24.956 ± 1.977*	**87.603 ± 6.945**	***84.708 ± 9.690***
UGAT	**24.453 ± 2.368**	**24.451 ± 2.307**	92.200 ± 7.364	*90.021 ± 11.592*

Table 4: Set sizes (lower is better) for 2 GNNs (GCN or GAT) under 2 representations (block adjacency or Unfolding) for 4 datasets. Values in bold indicate a smaller set size for a given GNN. Values in italics indicate invalid sets.

Figure 3 compares accuracy of both UGAT and block GAT for each time window of the school dataset. During lunchtime, pupils are no longer clustered in their classrooms, making prediction of their class more difficult. Figure 3 confirms a significant drop in accuracy for both models at lunchtime in the transductive case. UGAT also displays this behaviour in the semi-inductive case, while block GAT's performance is no better than random selection.

At these more difficult lunchtime windows, the prediction set sizes increase for both methods in the transductive case as shown in Figure 4. UGAT also displays this increase in the semi-inductive regime, while block GAT does not adapt. This demonstrates that conformal prediction can quantify classification difficulty over time when using a UGNN. Note that while computing accuracy requires the knowledge of test labels to quantify difficulty, conformal prediction of set size does not. In this particular example, both methods maintain coverage in the transductive case (increasing uncertainty at lunchtimes), and UGAT maintains coverage in the semi-inductive case. The story is similar for UGCN (Appendix D).



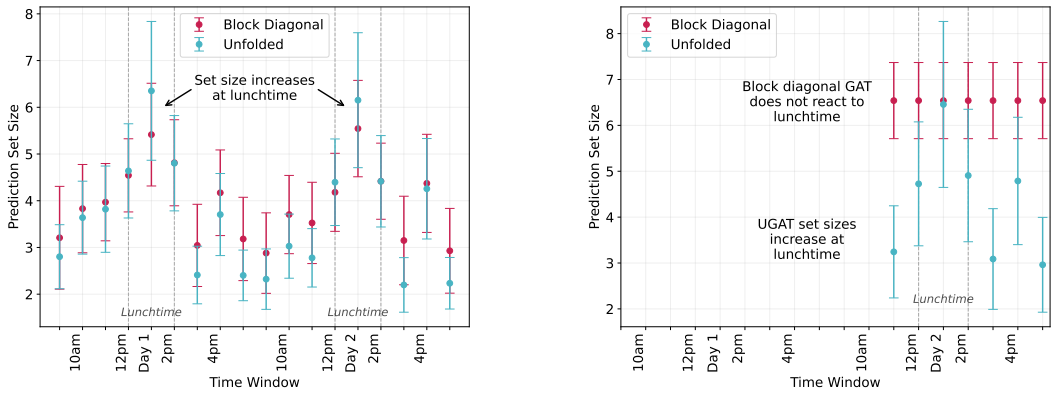
(a) Transductive.

(b) Semi-inductive.

Figure 3: Prediction accuracy for each time window of the school dataset for unfolded GAT and block diagonal GAT. The prediction task gets more difficult at lunchtime, as shown by the drop in accuracy of both methods in the transductive case. UGAT has marginally better performance in the transductive case and significantly better performance in the semi-inductive case.

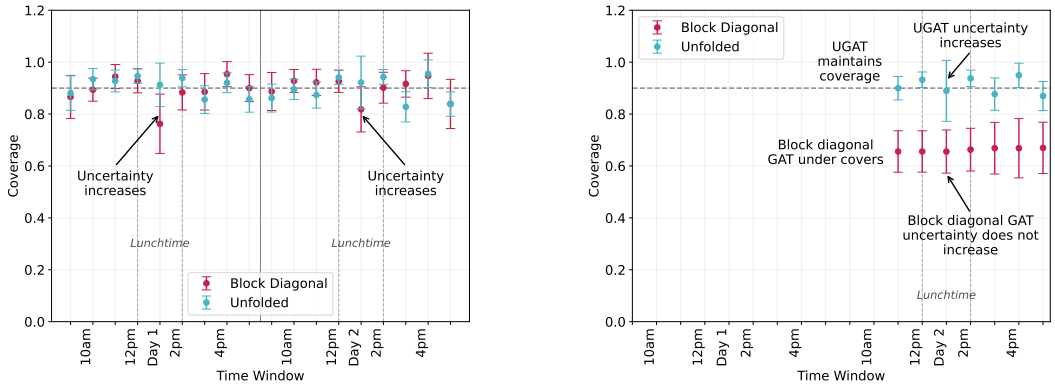
4 Discussion

This paper proposes unfolded GNNs, which exhibit exchangeability properties allowing conformal inference in both transductive and semi-inductive regimes. We demonstrate improved predictive performance, coverage, and prediction set size across a range of simulated and real data examples.



(a) Transductive. (b) Semi-inductive.

Figure 4: Prediction set sizes for each time window of the school dataset for unfolded GAT and block diagonal GAT. As the prediction task gets more difficult at lunchtime the prediction set sizes increase. Only the UGAT set sizes react to lunchtime in the semi-inductive case.



(a) Transductive. (b) Semi-inductive.

Figure 5: Coverage at each time window of the school dataset for unfolded GAT and block diagonal GAT. Both methods maintain target coverage in the transductive case, with uncertainty increasing at the more difficult lunchtime window. UGAT also maintains target coverage in the semi-inductive case, while block GAT under-covers.

As the supplied code will demonstrate, we have intentionally avoided any type of fine-tuning, because the goal here is less prediction accuracy than uncertainty quantification. There are significant opportunities to improve the prediction performance of unfolded GNNs, such as employing architectures better suited to bipartite graphs.

There are also opportunities to improve the downstream treatment of embeddings, such as deploying the CF-GNN algorithm [57] to reduce prediction set sizes, or weighting calibration points according to their exchangeability with the test point [24, 28].

Our use of unfolding originates from a body of statistical literature on spectral embedding and tensor decomposition, where it may be useful to observe that inductive embedding is straightforward: the singular vectors provide a projection operator which can be applied to new nodes. Being able to somehow emulate this operation in unfolded GNNs could provide a path towards fully inductive inference.

References

[1] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al.

Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[2] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. *Advances in neural information processing systems*, 28, 2015.

[3] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

[4] Jonathan M Stokes, Kevin Yang, Kyle Swanson, Wengong Jin, Andres Cubillos-Ruiz, Nina M Donghia, Craig R MacNair, Shawn French, Lindsey A Carfrae, Zohar Bloom-Ackermann, et al. A deep learning approach to antibiotic discovery. *Cell*, 180(4):688–702, 2020.

[5] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020.

[6] Hao Peng, Jianxin Li, Yu He, Yaopeng Liu, Mengjiao Bao, Lihong Wang, Yangqiu Song, and Qiang Yang. Large-scale hierarchical text classification with recursively regularized deep graph-cnn. In *Proceedings of the 2018 world wide web conference*, pages 1063–1072, 2018.

[7] Lingfei Wu, Yu Chen, Kai Shen, Xiaojie Guo, Hanning Gao, Shucheng Li, Jian Pei, Bo Long, et al. Graph neural networks for natural language processing: A survey. *Foundations and Trends® in Machine Learning*, 16(2):119–328, 2023.

[8] Rianne van den Berg, Thomas N Kipf, and Max Welling. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263*, 2017.

[9] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. Graph neural networks in recommender systems: a survey. *ACM Computing Surveys*, 55(5):1–37, 2022.

[10] Fedor Borisyuk, Shihai He, Yunbo Ouyang, Morteza Ramezani, Peng Du, Xiaochen Hou, Chengming Jiang, Nitin Pasumarthy, Priya Bannur, Birjodh Tiwana, et al. Lignn: Graph neural networks at linkedin. *arXiv preprint arXiv:2402.11139*, 2024.

[11] Austin Derrow-Pinion, Jennifer She, David Wong, Oliver Lange, Todd Hester, Luis Perez, Marc Nunkesser, Seongjae Lee, Xueying Guo, Brett Wiltshire, et al. Eta prediction with graph neural networks in google maps. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3767–3776, 2021.

[12] Alex Davies, Petar Veličković, Lars Buesing, Sam Blackwell, Daniel Zheng, Nenad Tomašev, Richard Tanburn, Peter Battaglia, Charles Blundell, András Juhász, et al. Advancing mathematics by guiding human intuition with ai. *Nature*, 600(7887):70–74, 2021.

[13] Open graph benchmark. `https://ogb.stanford.edu/`. Accessed: 2024-05-22.

[14] William L Hamilton. *Graph representation learning*. Morgan & Claypool Publishers, 2020.

[15] Benjamin Sanchez-Lengeling, Emily Reif, Adam Pearce, and Alexander B. Wiltschko. A gentle introduction to graph neural networks. *Distill*, 2021. https://distill.pub/2021/gnn-intro.

[16] Pyg documentation. https://pytorch-geometric.readthedocs.io/en/latest/. Accessed: 2024-05-22.

[17] Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*, volume 29. Springer, 2005.

[18] Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3), 2008.

[19] Jing Lei, James Robins, and Larry Wasserman. Distribution-free prediction sets. *Journal of the American Statistical Association*, 108(501):278–287, 2013.

[20] Jing Lei and Larry Wasserman. Distribution-free prediction bands for non-parametric regression. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 76(1):71–96, 2014.

[21] Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111, 2018.

[22] Rina Foygel Barber, Emmanuel J Candes, Aaditya Ramdas, and Ryan J Tibshirani. The limits of distribution-free conditional predictive inference. *Information and Inference: A Journal of the IMA*, 10(2):455–482, 2021.

[23] Ryan J Tibshirani, Rina Foygel Barber, Emmanuel Candes, and Aaditya Ramdas. Conformal prediction under covariate shift. *Advances in neural information processing systems*, 32, 2019.

[24] Rina Foygel Barber, Emmanuel J Candes, Aaditya Ramdas, and Ryan J Tibshirani. Conformal prediction beyond exchangeability. *The Annals of Statistics*, 51(2):816–845, 2023.

[25] Isaac Gibbs, John J Cherian, and Emmanuel J Candès. Conformal prediction with conditional guarantees. *arXiv preprint arXiv:2305.12616*, 2023.

[26] Yaniv Romano, Matteo Sesia, and Emmanuel Candes. Classification with valid and adaptive coverage. *Advances in Neural Information Processing Systems*, 33:3581–3591, 2020.

[27] Shenyang Huang, Farimah Poursafaei, Jacob Danovitch, Matthias Fey, Weihua Hu, Emanuele Rossi, Jure Leskovec, Michael Bronstein, Guillaume Rabusseau, and Reihaneh Rabbany. Temporal graph benchmark for machine learning on temporal graphs. *Advances in Neural Information Processing Systems*, 36, 2024.

[28] Jase Clarkson. Distribution free prediction sets for node classification. In *International Conference on Machine Learning*, pages 6268–6278. PMLR, 2023.

[29] Soroush H Zargarbashi and Aleksandar Bojchevski. Conformal inductive graph neural networks. In *The Twelfth International Conference on Learning Representations*, 2023.

[30] Tristan Bilot, Nour El Madhoun, Khaldoun Al Agha, and Anis Zouaoui. Graph neural networks for intrusion detection: A survey. *IEEE Access*, 2023.

[31] Haoyu He, Yuede Ji, and H Howie Huang. Illuminati: Towards explaining graph neural networks for cybersecurity analysis. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, pages 74–89. IEEE, 2022.

[32] Benjamin Bowman and H Howie Huang. Towards next-generation cybersecurity with graph ai. *ACM SIGOPS Operating Systems Review*, 55(1):61–67, 2021.

[33] Pedro Szekely, Craig A Knoblock, Jason Slepicka, Andrew Philpot, Amandeep Singh, Chengye Yin, Dipsy Kapoor, Prem Natarajan, Daniel Marcu, Kevin Knight, et al. Building and using a knowledge graph to combat human trafficking. In *The Semantic Web-ISWC 2015: 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part II 14*, pages 205–221. Springer, 2015.

[34] Pratheeksha Nair, Javin Liu, Catalina Vajiac, Andreas Olligschlaeger, Duen Horng Chau, Mirela Cazzolato, Cara Jones, Christos Faloutsos, and Reihaneh Rabbany. T-net: Weakly supervised graph learning for combatting human trafficking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38(20), pages 22276–22284, 2024.

[35] Huyen Trang Phan, Ngoc Thanh Nguyen, and Dosam Hwang. Fake news detection: A survey of graph neural network methods. *Applied Soft Computing*, page 110235, 2023.

[36] Chee Wei Tan, Pei-Duo Yu, Siya Chen, and H Vincent Poor. Deeptrace: Learning to optimize contact tracing in epidemic networks with graph neural networks. *arXiv preprint arXiv:2211.00880*, 2022.

[37] Zheng Liu, Xiaohan Li, Hao Peng, Lifang He, and S Yu Philip. Heterogeneous similarity graph neural network on electronic health records. In *2020 IEEE international conference on big data (big data)*, pages 1196–1205. IEEE, 2020.

[38] Heloísa Oss Boll, Ali Amirahmadi, Mirfarid Musavian Ghazani, Wagner Ourique de Morais, Edison Pignaton de Freitas, Amira Soliman, Kobra Etminani, Stefan Byttner, and Mariana Recamonde-Mendoza. Graph neural networks for clinical risk prediction based on electronic health records: A survey. *Journal of Biomedical Informatics*, page 104616, 2024.

[39] Anru Zhang and Dong Xia. Tensor svd: Statistical and computational limits. *IEEE Transactions on Information Theory*, 64(11):7311–7338, 2018.

[40] Joshua Cape, Minh Tang, and Carey E. Priebe. The two-to-infinity norm and singular subspace geometry with applications to high-dimensional statistics. *The Annals of Statistics*, 47(5):2405 – 2439, 2019.

[41] Emmanuel Abbe, Jianqing Fan, Kaizheng Wang, and Yiqiao Zhong. Entrywise eigenvector analysis of random matrices with low expected rank. *Annals of statistics*, 48(3):1452, 2020.

[42] Patrick Rubin-Delanchy, Joshua Cape, Minh Tang, and Carey E Priebe. A statistical interpretation of spectral embedding: the generalised random dot product graph. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(4):1446–1473, 2022.

[43] Andrew Jones and Patrick Rubin-Delanchy. The multilayer random dot product graph. *arXiv preprint arXiv:2007.10455*, 2020.

[44] Ian Gallagher, Andrew Jones, and Patrick Rubin-Delanchy. Spectral embedding for dynamic networks with stability guarantees. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 10158–10170. Curran Associates, Inc., 2021.

[45] Joshua Agterberg and Anru Zhang. Estimating higher-order mixed memberships via the $\ell_{2,\infty}$ tensor perturbation bound. *arXiv preprint arXiv:2212.08642*, 2022.

[46] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[47] Ed Davis, Ian Gallagher, Daniel John Lawson, and Patrick Rubin-Delanchy. A simple and powerful framework for stable dynamic network embedding. *arXiv preprint arXiv:2311.09251*, 2023.

[48] Fan Wang, Wanshan Li, Oscar Hernan Madrid Padilla, Yi Yu, and Alessandro Rinaldo. Multilayer random dot product graphs: Estimation and online change point detection. *arXiv preprint arXiv:2306.15286*, 2023.

[49] Tianbao Yang, Yun Chi, Shenghuo Zhu, Yihong Gong, and Rong Jin. Detecting communities and their evolutions in dynamic social networks—a bayesian approach. *Machine learning*, 82:157–189, 2011.

[50] Kevin S Xu and Alfred O Hero. Dynamic stochastic blockmodels for time-evolving social networks. *IEEE Journal of Selected Topics in Signal Processing*, 8(4):552–562, 2014.

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

[52] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[53] Juliette Stehlé, Nicolas Voirin, Alain Barrat, Ciro Cattuto, Lorenzo Isella, Jean-François Pinton, Marco Quaggiotto, Wouter Van den Broeck, Corinne Régis, Bruno Lina, et al. High-resolution measurements of face-to-face contact patterns in a primary school. *PloS one*, 6(8):e23176, 2011.

[54] Xavier Olive, Martin Strohmeier, and Jannis Lübbe. Crowdsourced air traffic data from The OpenSky Network 2020 [CC-BY], January 2022.

[55] Graham K MacDonald, Kate A Brauman, Shipeng Sun, Kimberly M Carlson, Emily S Cassidy, James S Gerber, and Paul C West. Rethinking agricultural trade relationships in an era of globalization. *BioScience*, 65(3):275–289, 2015.

[56] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[57] Kexin Huang, Ying Jin, Emmanuel Candes, and Jure Leskovec. Uncertainty quantification over graph with conformalized graph neural networks. *Advances in Neural Information Processing Systems*, 36, 2024.

[58] Vladimir Vovk. Conditional validity of inductive conformal predictors. In *Asian conference on machine learning*, pages 475–490. PMLR, 2012.

[59] Anastasios N Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2021.

A Code availability

Python code for reproducing experiments is available at `https://github.com/edwarddavis1/valid_conformal_for_dynamic_gnn`.

B Supporting theory for Section 2

B.1 Transductive theory

Our arguments are similar in style to [24], showing validity of full conformal inference, and then split conformal as a special case of full conformal.

We observe Y_w for all $w \in \nu$ apart from Y_{ν_K}, where the query point K is chosen independently and uniformly at random among $\{1, \ldots, m+1\}$.

B.1.1 Full conformal prediction

Suppose we have access to an algorithm $\mathcal{A}(g, x, y)$ which takes as input:

1. g, a dynamic graph on n nodes and T time points;
2. x, a $|\nu|$-long sequence of feature vectors;
3. y, a $|\nu|$-long sequence of labels;

and returns a sequence $r = (r_1, \ldots, r_{m+1})$ of real values, which will act as non-conformity scores.

Algorithm 2 Full conformal inference

Input: Dynamic graph G, index set ν, attributes $(X_w, w \in \nu)$, labels $(Y_w, w \in \nu \setminus \nu_K)$, confidence level α, algorithm $\mathcal{A}$, query point K

1: Let $\hat{C} = \mathcal{Y}$
2: **for** $y \in \mathcal{Y}$ **do**
3: Let

$$X = (X_{\nu_1}, \ldots, X_{\nu_{|\nu|}}); \quad Y^+ = (Y_{\nu_1}, \ldots, Y_{\nu_{K-1}}, y, Y_{\nu_{K+1}}, \ldots, Y_{\nu_{|\nu|}}).$$

4: Compute

$$(R_1, \ldots, R_{m+1}) = \mathcal{A}(G, X, Y^+).$$

5: Remove y from $\hat{C}$ if

$$R_K \text{ among } \lfloor \alpha(m+1) \rfloor \text{ largest of } R_1, \ldots, R_{m+1}.$$

6: **end for**

Output: Prediction set $\hat{C}$

Lemma 2. *The prediction set output by Algorithm 2 is valid.*

Proof. Because $K \overset{ind}{\sim} \text{uniform}([m+1])$, the event

$$E = \text{“}R_K \text{ among } \lfloor \alpha(m+1) \rfloor \text{ largest of } R_1, \ldots, R_{m+1}\text{”},$$

occurs with probability

$$\mathbb{P}(E) \leq \alpha.$$

But E occurs if and only if $Y_{\nu_K} \notin \hat{C}$. Therefore,

$$\mathbb{P}(Y_{\nu_K} \in \hat{C}) = 1 - \mathbb{P}(Y_{\nu_K} \notin \hat{C}) = 1 - \mathbb{P}(E) \geq 1 - \alpha.$$

□

What this formalism makes clear is that the validity of full conformal inference in a transductive setting has nothing to do with exchangeability or any form of symmetry in $\mathcal{A}$ (neither of which were assumed).

B.1.2 Split conformal prediction

Algorithm 3 Split conformal inference (abstract version)

Input: Dynamic graph G, index set ν, attributes $(X_w, w \in \nu)$, labels $(Y_w, w \in \nu \setminus \nu_K)$, confidence level α, prediction function $\hat{f}$, non-conformity function r, query point K

1: Let $\hat{C} = \mathcal{Y}$
2: Let $R_i = r(\hat{f}(\nu_i), Y_{\nu_i})$ for $i \in [m+1] \setminus K$
3: Let

$$\hat{q} = \lfloor \alpha(m+1) \rfloor \text{ largest value in } R_i, i \in [m+1] \setminus K$$

4: **for** $y \in \mathcal{Y}$ **do**
5: Let $R_K = r(\hat{f}(\nu_K), y)$
6: Remove y from $\hat{C}$ if $R_K \geq \hat{q}$
7: **end for**

Output: Prediction set $\hat{C}$

Suppose we construct $\mathcal{A}$ according to a split training/calibration routine, as follows. First, using g, ν, the feature vectors $(X_w, w \in \nu)$, and the labels $Y_{\nu_\ell}, \ell > m+1$, learn a prediction function $\hat{f} : \nu \to \mathbb{R}^d$.

The vector $\hat{f}(\nu_\ell)$ might, for example, relate to predicted class probabilities via

$$\hat{\mathbb{P}}(Y_{\nu_\ell} = k) = \Big[\text{softmax}(\hat{f}(\nu_\ell)_1, \ldots, \hat{f}(\nu_\ell)_d)\Big]_k,$$

in which case $\hat{f}(\nu_\ell)$ is often known as an embedding.

Second, for a non-conformity function $r : \mathbb{R}^d \times \mathcal{Y} \to \mathbb{R}$, a dynamic graph g, and x, y of length $|\nu|$, define

$$[\mathcal{A}(g, x, y)]_\ell = r(\hat{f}(\nu_\ell), y_\ell), \quad \ell \in [m+1].$$

Then Algorithm 2 can potentially be made much more efficient, as shown in Algorithm 1. This is nothing but the usual split conformal inference algorithm, where the labels $Y_{\nu_\ell}, \ell > m+1$ are being used for training, and $Y_{\nu_\ell}, \ell \in [m+1] \setminus K$ for calibration.

Thus split conformal inference is a special case of full conformal inference, in which special structure is imposed on $\mathcal{A}$, and must therefore inherit its general validity (Lemma 1).

B.2 Semi-inductive theory

Otherwise keeping the same setup, suppose K is not uniformly chosen but is instead fixed, to (say) $K = 1$.

Given a permutation $\pi : [m+1] \to [m+1]$, we will write πG to denote a dynamic graph in which the columns of G corresponding to $\nu_1, \ldots, \nu_{m+1}$ are permuted according to π. More precisely,

πG is the dynamic graph corresponding to the sequence of adjacency matrices $\mathbf{A}^{(1)'}, \ldots, \mathbf{A}^{(T)'}$ obtained as follows. First, copy $\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(T)}$ to $\mathbf{A}^{(1)'}, \ldots, \mathbf{A}^{(T)'}$. Next, overwrite any column $\nu_\ell \in \nu_1, \ldots, \nu_{m+1}$ to $\mathbf{A}'_{\nu_\ell} = \mathbf{A}_{\nu_{\pi^{-1}(\ell)}}$. If x is a sequence of length at least $m+1$, we write πx to denote the same sequence with its first $m+1$ elements permuted according to π, that is, $(\pi x)_\ell = x_{\pi^{-1}(\ell)}$ if $\ell \leq m+1$ and $(\pi x)_\ell = x_\ell$ otherwise.

A more fully stated version of Assumption 1 is

A3. *The columns $\mathbf{A}_w, w \in \nu$, attributes $X_w, w \in \nu$, and labels $Y_w, w \in \nu$ are jointly exchangeable, that is, for any permutation $\pi : [m+1] \to [m+1]$,*

$$\{\pi G, \pi(X_w, w \in \nu), \pi(Y_w, w \in \nu)\} \triangleq \{G, (X_w, w \in \nu), (Y_w, w \in \nu)\}.$$

The symbol $\triangleq$ means "equal in distribution".

A4. *The algorithm $\mathcal{A}(g, x, y)$ is symmetric in its input. For any permutation $\pi : \nu \to \nu$,*

$$\mathcal{A}(\pi g, \pi x, \pi y) = \pi \mathcal{A}(g, x, y)$$

Theorem 2. *Under assumptions 3 and 4, the prediction set output by Algorithm 2 (and so Algorithm 3) is valid even if $K = 1$ deterministically.*

Proof. The combination of assumptions 3 and 4 ensures that $R_1, \ldots, R_{m+1}$ are exchangeable, and therefore the event

$$E = \text{“}R_1 \text{ among } \lfloor \alpha(m+1) \rfloor \text{ largest of } R_1, \ldots, R_{m+1}\text{”},$$

occurs with probability

$$\mathbb{P}(E) \leq \alpha.$$

But E occurs if and only if $Y_{\nu_1} \notin \hat{C}$. Therefore,

$$\mathbb{P}(Y_{\nu_1} \in \hat{C}) = 1 - \mathbb{P}(Y_{\nu_K} \notin \hat{C}) = 1 - \mathbb{P}(E) \geq 1 - \alpha.$$

□

For a permutation $\pi : [N] \to [N]$ with associated permutation matrix Π, we say that the GNN is label equivariant if

$$\mathcal{G}(\Pi \mathbf{A} \Pi^\top) = \Pi \mathcal{G}(\mathbf{A}).$$

To map Theorem 2 to Theorem 1, we observe that Assumptions 1 and 3 are the same (the latter a detailed version of the former); and that inputting $\mathbf{A}^{\text{UNF}}$ to a GNN satisfying Assumption 2, along with the attributes $X_w, w \in \nu$ and training labels $Y_w, w \in \nu^{(\text{training})}$, results in an $\mathcal{A}$ satisfying Assumption 4.

C Conditional Coverage

Table 5 displays the worst-case coverage across all time points for both methods, which as expected is not always conditionally valid. The problem here is similar to the problem mentioned with drift. As no two time points are exactly exchangeable, no two embedded time points will be exactly exchangeable. This then translates to variation in coverage between time points. In the marginal coverage case, variation is fine as long as the average coverage is on target. However, under this metric, any variation causes a decrease.

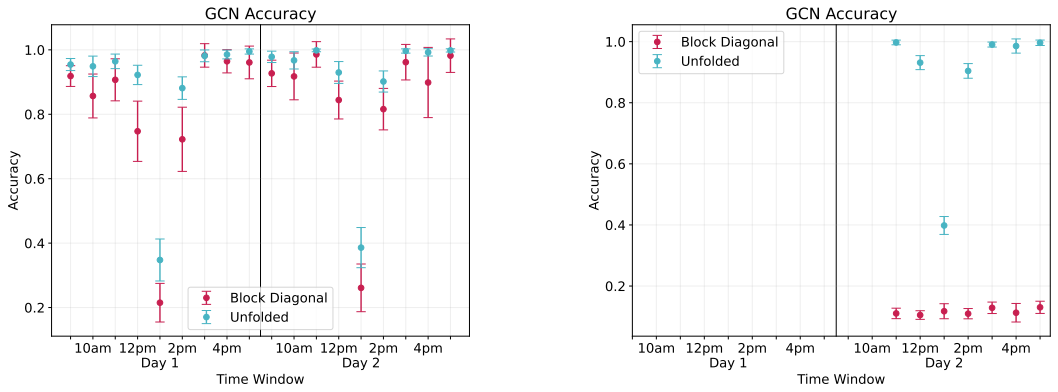
Methods	SBM		School	
	Trans.	Semi-ind.	Trans.	Semi-ind.
Block GCN	0.760 ± 0.067	0.619 ± 0.063	$\mathbf{0.746 \pm 0.045}$	$\mathbf{0.767 \pm 0.040}$
UGCN	$\mathbf{0.790 \pm 0.035}$	$\mathbf{0.857 \pm 0.026}$	0.693 ± 0.045	0.710 ± 0.041
Block GAT	0.774 ± 0.054	0.450 ± 0.154	0.702 ± 0.061	0.616 ± 0.085
UGAT	$\mathbf{0.836 \pm 0.030}$	$\mathbf{0.879 \pm 0.031}$	$\mathbf{0.763 \pm 0.048}$	$\mathbf{0.810 \pm 0.085}$

Methods	Flight		Trade	
	Trans.	Semi-ind.	Trans.	Semi-ind.
Block GCN	$\mathbf{0.872 \pm 0.008}$	0.849 ± 0.014	$\mathbf{0.745 \pm 0.043}$	0.749 ± 0.029
UGCN	0.871 ± 0.007	$\mathbf{0.893 \pm 0.004}$	0.738 ± 0.061	$\mathbf{0.755 \pm 0.037}$
Block GAT	0.874 ± 0.006	0.858 ± 0.014	$\mathbf{0.763 \pm 0.037}$	0.755 ± 0.032
UGAT	$\mathbf{0.878 \pm 0.006}$	$\mathbf{0.895 \pm 0.004}$	0.729 ± 0.049	$\mathbf{0.765 \pm 0.041}$

Table 5: Worst-case coverage over time with target ≥ 0.90. Values in bold indicate closest to the target for a given GNN.

D Temporal Analysis using GCN

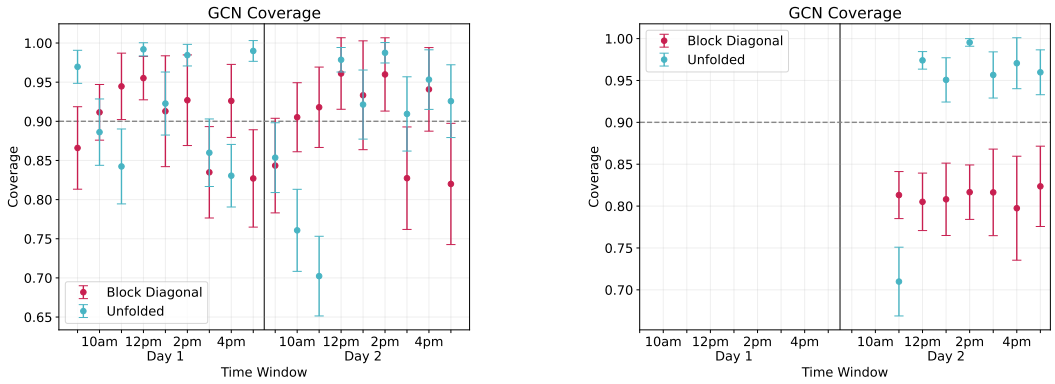
In this section, we present a temporal analysis of the school data example using GCN instead of GAT as the GNN model. We see similar conclusions to those stated in Section 3. The accuracy of UGCN is slightly higher at every time point in the transductive case and much higher in the semi-inductive case. Block GCN is essentially guessing randomly in the semi-inductive case. The prediction sets for UGCN are always smaller than those of block GCN, with UGCN's set sizes also reacting to the difficulty problem in the semi-inductive regime. However, neither method achieves valid coverage of every time point in the transductive regime, in contrast to GAT. UGCN also under-covers the first test point in the semi-inductive case (unlike UGAT), while block GCN continually under-covers here.



(a) Transductive.

(b) Semi-inductive.

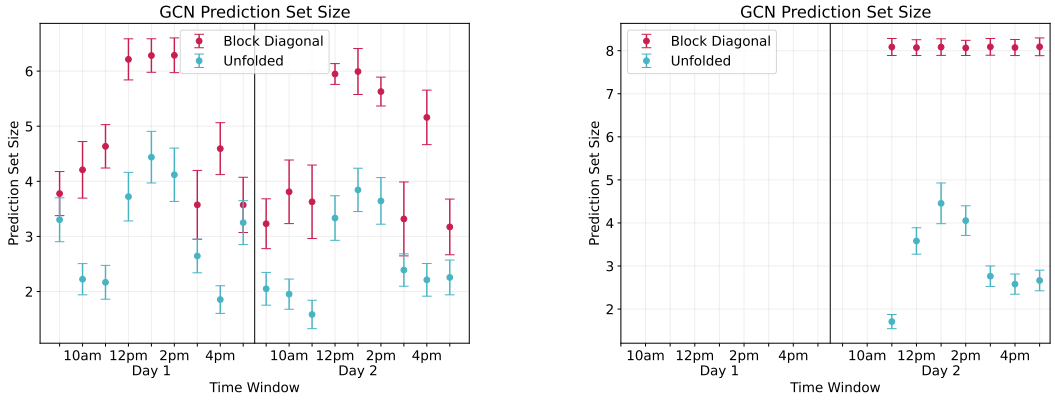
Figure 6: Prediction accuracy for each time window of the school dataset for unfolded GCN and block diagonal GCN. The prediction task gets more difficult at lunchtime, as shown by the drop in accuracy of both methods in the transductive case. UGCN has marginally better performance in the transductive case and significantly better performance in the semi-inductive case.



(a) Transductive.

(b) Semi-inductive.

Figure 7: Coverage for each time window of the school dataset for unfolded GCN and block diagonal GCN. Both methods maintain target coverage on average in the transductive case, but not at every point in time. In the semi-inductive case, block GCN under-covers continuously and UGCN under-covers for the first test point.



(a) Transductive.

(b) Semi-inductive.

Figure 8: Prediction set sizes for each time window of the school dataset for unfolded GCN and block diagonal GCN. As the prediction task gets more difficult at lunchtime the prediction set sizes increase. Only the UGCN set sizes react to lunchtime in the semi-inductive case.