VALID CONFORMAL PREDICTION FOR DYNAMIC GNNS

Ed Davis[1] **Ian Gallagher**[2] **Daniel John Lawson**[1] **Patrick Rubin-Delanchy**[3]
[1]University of Bristol, U.K. [2]The University of Melbourne, Australia
[3]School of Mathematics and Maxwell Institute for Mathematical Sciences,
University of Edinburgh, U.K.
{edward.davis, dan.lawson}@bristol.ac.uk
ian.gallagher@unimelb.edu.au
prd@ed.ac.uk

ABSTRACT

Dynamic graphs provide a flexible data abstraction for modelling many sorts of real-world systems, such as transport, trade, and social networks. Graph neural networks (GNNs) are powerful tools allowing for different kinds of prediction and inference on these systems, but getting a handle on uncertainty, especially in dynamic settings, is a challenging problem.

In this work we propose to use a dynamic graph representation known in the tensor literature as the unfolding, to achieve valid prediction sets via conformal prediction. This representation, a simple graph, can be input to any standard GNN and does not require any modification to existing GNN architectures or conformal prediction routines.

One of our key contributions is a careful mathematical consideration of the different inference scenarios which can arise in a dynamic graph modelling context. For a range of practically relevant cases, we obtain valid prediction sets with almost no assumptions, even dispensing with exchangeability. In a more challenging scenario, which we call the semi-inductive regime, we achieve valid prediction under stronger assumptions, akin to stationarity.

We provide real data examples demonstrating validity, showing improved accuracy over baselines, and sign-posting different failure modes which can occur when those assumptions are violated.

1 INTRODUCTION

Graph neural networks (GNNs) have seen success in a wide variety of application domains including prediction of protein structure (Jumper et al., 2021) and molecular properties (Duvenaud et al., 2015; Gilmer et al., 2017), drug discovery (Stokes et al., 2020), computer vision (Sarlin et al., 2020), natural language processing (Peng et al., 2018; Wu et al., 2023), recommendation systems (Berg et al., 2017; Wu et al., 2022; Borisyuk et al., 2024), estimating time of arrival (ETA) in services like Google Maps (Derrow-Pinion et al., 2021), and advancing mathematics (Davies et al., 2021). They often top leaderboards in benchmarks relating to machine-learning on graphs (Ope), for a range of tasks such as node, edge, or graph property prediction. Useful resources for getting started with GNNs include the introductions (Hamilton, 2020; Sanchez-Lengeling et al., 2021) and the Pytorch Geometric library (Pyt).

Conformal prediction (CP), advanced by Vovk and co-authors (Vovk et al., 2005), and later expanded on by various researchers in the Statistics community (Shafer and Vovk, 2008; Lei et al., 2013; Lei and Wasserman, 2014; Lei et al., 2018; Foygel Barber et al., 2021; Tibshirani et al., 2019; Barber et al., 2023; Gibbs et al., 2023; Romano et al., 2020), 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 (Huang et al., 2024b; Clarkson, 2023; Zargarbashi and Bojchevski, 2023).

There are many applications of significant societal importance, e.g. cyber-security (Bilot et al., 2023; He et al., 2022; Bowman and Huang, 2021), human-trafficking prevention (Szekely et al., 2015; Nair et al., 2024), social media & misinformation (Phan et al., 2023), contact-tracing (Tan et al., 2022), patient care (Liu et al., 2020; Boll et al., 2024) 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. The unsolved problem we address, in a nutshell, is guaranteeing that two nodes behaving in the same way at distinct points in time are embedded in an exchangeable way, regardless of global network dynamics.

Inspired by a concurrent line of statistical research on spectral embedding and tensor decomposition (Zhang and Xia, 2018; Cape et al., 2019; Abbe et al., 2020; Rubin-Delanchy et al., 2022; Jones and Rubin-Delanchy, 2020; Gallagher et al., 2021; Agterberg and Zhang, 2022), we propose to use one of the *unfoldings* (De Lathauwer et al., 2000) of the tensor representation of the dynamic graph as input to a standard GNN, and demonstrate validity in both transductive and semi-inductive regimes.

Related work. The use of (tensor) unfoldings in GNNs is new. At present, to our knowledge, the only paper studying conformal inference on dynamic graphs is (Zargarbashi and Bojchevski, 2023), but the paper considers a strictly growing graph, which is very different from our meaning, in which a 'dynamic graph' has edges which appear and disappear in complex and random ways, as in all of our examples and in the applications alluded to above. The motivation for our work derives from earlier contributions studying CP on static graphs using GNNs (Huang et al., 2024b; Clarkson, 2023; Zargarbashi and Bojchevski, 2023) but our theory is distinct from those contributions, with arguments closer to (Barber et al., 2023). In the dynamic graph literature, unfoldings have exclusively been proposed for unsupervised settings (Jones and Rubin-Delanchy, 2020; Gallagher et al., 2021; Agterberg and Zhang, 2022; Davis et al., 2023; Wang et al., 2023), particularly spectral embedding, and have never been applied to conformal inference.

Our contribution should be viewed as a novel *interface* between CP and GNNs, rather than an improvement of either. This interface allows for a modular system design in dynamic graph inference in which both the GNN and CP modules can be chosen and optimised independently. In view of the rapid development of both fields, we have opted against fine-tuning to allow more transparent, stable, and reproducible performance assessments using recognised and standard baselines for GNNs and CP: Graph Convolutional Networks (GCN) (Kipf and Welling, 2016) and Graph Attention Networks (GAT) (Veličković et al., 2017) for GNNs and APS (Romano et al., 2020) for CP. Our theory applies to the *representation* of the dynamic graph, not the choice of GNN or CP procedures.

2 Theory & Methods

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

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 node/time 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 not be observed, in some discrete set of cardinality d;
2. a set of attributes, which is observed, and represented by a feature vector $X_i^{(t)} \in \mathbb{R}^c$;
3. a corresponding column in $\mathbf{A}^{(t)}$, denoted $\mathbf{A}_i^{(t)}$, with at least one non-zero entry.

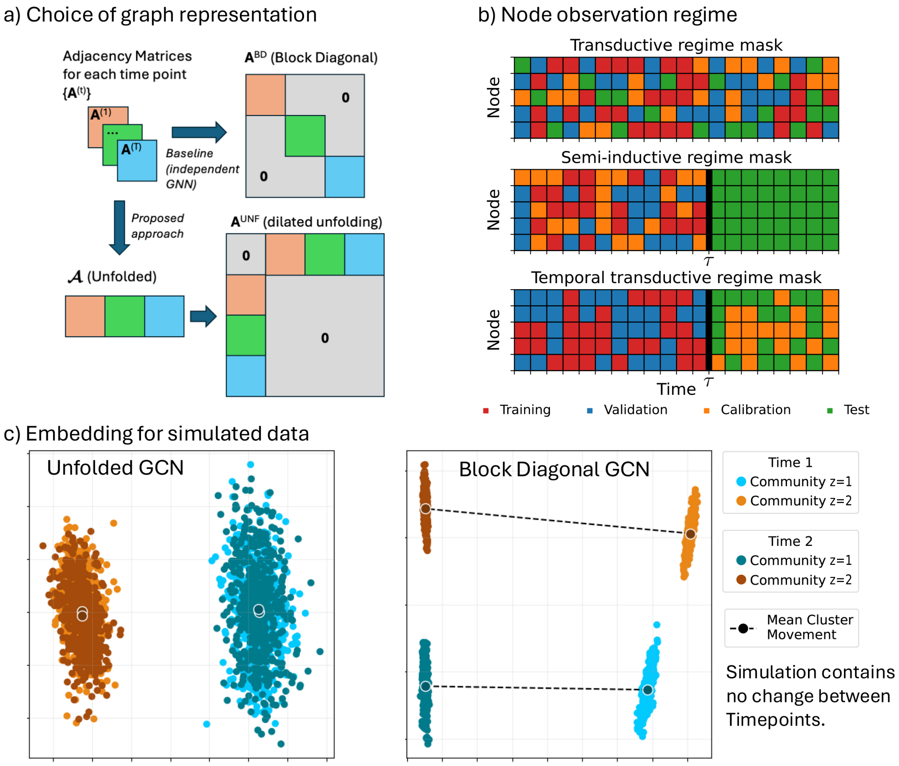


Figure 1: Contribution overview. a) This paper is about the representation of the collection of adjacency matrix snapshots. The baseline (current practice) approach treats these as independent and can be viewed as padding a 'block-diagonal' matrix with zeroes. Unfolding instead column concatenates which links nodes to themselves over time. Dilation results in a square symmetric matrix. b) Which data are available at training time affects performance; we report results for transductive (all time-points are exchangeable in terms of test/train split), semi-inductive (a future period is reserved for testing), or temporal transductive (a future period is reserved for testing and calibration). c) Simulation of an i.i.d. stochastic block model showing the embedding 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.

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

The data are split into four sets: $\nu^{(\text{training})}$ to learn the GNN parameters; $\nu^{(\text{validation})}$ for out-of-sample performance evaluation of the GNN; $\nu^{(\text{calibration})}$ to train the conformal model, and $\nu^{(\text{test})}$ is the target for conformal inference.

Clearly, the theory and methodology of this paper are applicable to multiple test nodes, but for simplicity in this section we assume $\nu^{(\text{test})}$ comprises only one node.

From these objects we define three versions of the inference problem.

Transductive regime. The test label is missing completely at random allowing training, validation, and calibration at any time (Figure 1b, top).

Temporal transductive regime. Training and validation are undertaken up to some time τ. The test label is missing completely at random after time τ, allowing calibration from this interval (Figure 1b, bottom).

Semi-inductive regime. Training and validation is undertaken up to some time τ. The test label occurs after time τ, corresponding to a fixed node i, preventing calibration from this interval (Figure 1b, middle).

In either of the transductive regimes, a valid confidence set can be constructed with no further assumptions (Algorithm 1). We believe this could be exciting because we view the temporal transductive regime, in particular, as an extremely common scenario. The semi-inductive regime is more challenging, and reflects 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.

Formally, for some $m < |\nu| - 1$, we reorder the dataset ν so that the first m node/time pairs correspond to the calibration and test sets, while the remaining pairs correspond to the training and validation sets. For both transductive regimes, the missing label is missing at random among the first m node/time pairs of ν; for the semi-inductive regime, the missing label corresponds to a fixed node/time pair in this set. Let $\nu^{(\text{test})} := \{w_{\text{test}}\}$ denote the missing node/time pair, $\nu^{(\text{calibration})} := \{w_\ell, \ell \leq m\} \setminus \nu^{(\text{test})}$, and $\nu^{(\text{training})} \cup \nu^{(\text{validation})} := \{w_\ell, \ell > m\}$.

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

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

To represent the adjacency matrix we apply dilated unfolding (Davis et al., 2023) representing the dynamic graph G in the form

$$\mathbf{A}^{\text{UNF}} := \begin{bmatrix} \mathbf{0} & \boldsymbol{\mathcal{A}} \\ \boldsymbol{\mathcal{A}}^\top & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{(p+nT)\times(p+nT)}$$

where $\boldsymbol{\mathcal{A}} = \left(\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(T)}\right) \in \mathbb{R}^{p\times nT}$ (column-concatenation) containing p global (i.e. not t dependent) nodes and nT temporal nodes (each a node/time pair).

The GNN takes as input a matrix $X^{\text{UNF}} \in \mathbb{R}^{(p+nT)\times c}$ containing c-dimensional attributes for all nodes in the dilated unfolding. For networks without attributes this matrix is the $(p+nT)\times(p+nT)$ identity matrix. The GNN also requires training and validation labels $Y_w, w \in \nu^{(\text{training})} \cup \nu^{(\text{validation})}$, where global nodes are not assigned a label. The resulting output splits into embeddings

$$\begin{bmatrix} \hat{\mathbf{U}}^{\text{UNF}} \\ \hat{\mathbf{V}}^{\text{UNF}} \end{bmatrix} := \boldsymbol{\mathcal{G}}\left(\mathbf{A}^{\text{UNF}}, X^{\text{UNF}}, \{Y_w : w \in \nu^{(\text{training})} \cup \nu^{(\text{validation})}\}\right),$$

where $\hat{\mathbf{U}}^{\text{UNF}} \in \mathbb{R}^{p\times d}$ contains global representations and the embedding $\hat{\mathbf{V}}^{\text{UNF}} \in \mathbb{R}^{nT\times d}$ contains a representation of each node/time pair, of principal interest here.

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 Romano et al. (2020).

With these ingredients in place, we are in a position to obtain a prediction set $\hat{C}$ for w_{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 $(Y_w, w \in \nu \setminus \nu^{(\text{test})})$.

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,*

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

For the semi-inductive regime, we make the following assumptions:

A1. *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 $\pi : [m] \to [m]$,*

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

Algorithm 1 Split conformal inference

```
Input: Dynamic graph $G$, node/time pairs $\nu = \nu^{(\text{training})} \cup \nu^{(\text{validation})} \cup \nu^{(\text{calibration})} \cup \nu^{(\text{test})}$, attributes $(X_w, w \in \nu)$, labels $(Y_w, w \in \nu \setminus \nu^{(\text{test})})$, confidence level $\alpha$, non-conformity function $r$
Learn $\hat{\mathbf{V}}^{\text{UNF}}$ based on $\nu^{(\text{training})}$ and $\nu^{(\text{validation})}$
Let the initial confidence set be $\hat{C} = \mathcal{Y}$
Let $R_w = r(\hat{\mathbf{V}}^{\text{UNF}}_w, Y_w)$ for $w \in \nu^{(\text{calibration})}$
Let
    $\hat{q} = \lfloor \alpha(m) \rfloor$ largest value in $R_w, w \in \nu^{(\text{calibration})}$
for $y \in \mathcal{Y}$ do
    Let $R_{\text{test}} = r(\hat{\mathbf{V}}^{\text{UNF}}_{w_{\text{test}}}, y)$
    Remove $y$ from $\hat{C}$ if $R_{\text{test}} \geq \hat{q}$
end for
Output: Prediction set $\hat{C}$
```

The symbol $\triangleq$ means "equal in distribution". Informally, we can swap columns of $\mathcal{A} = \left(\mathbf{A}^{(1)}, \dots, \mathbf{A}^{(T)}\right)$ corresponding to test or calibration pairs, along with the attributes and labels, without changing the likelihood of the data.

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

A2. *The GNN $\mathcal{G}$ is permutation equivariant, that is, for any permutation $\pi : [N] \to [N]$ with associated permutation matrix Π,*

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

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,*

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

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 $\nu^{(\text{calibration})} \cup \nu^{(\text{test})}$ 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 $\nu^{(\text{calibration})} \cup \nu^{(\text{test})}$ 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) (Yang et al., 2011; Xu and Hero, 2014) 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 $z \in \{1, 2\}^n$ be a community allocation vector. We then draw each symmetric adjacency matrix point as $\mathbf{A}_{ij}^{(1)} \overset{\text{ind}}{\sim} \text{Bernoulli}(\mathbf{B}_{z_i,z_j}^{(1)})$ and $\mathbf{A}_{ij}^{(2)} \overset{\text{ind}}{\sim} \text{Bernoulli}(\mathbf{B}_{z_i,z_j}^{(2)})$, 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 (Fey and Lenssen, 2019),

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

from which a dynamic graph embedding is obtained through

$$\hat{\mathbf{V}}^{\text{BD}} = \mathcal{G}\left(\mathbf{A}^{\text{BD}}, X^{\text{BD}}, \{Y_w : w \in \nu^{(\text{training})} \cup \nu^{(\text{validation})}\}\right),$$

where $X^{\text{BD}} \in \mathbb{R}^{nT \times c}$ is the matrix of c-dimensional attributes for all node/time pairs. (For networks without attributes, X^{BD} is set to be the $nT \times nT$ identity matrix). From this representation, it is clear that $\mathbf{A}^{\text{BD}}$ is not exchangeable over time.

Figure 1 plots the representations, $\hat{\mathbf{V}}_{\text{BD}}$ and $\hat{\mathbf{V}}_{\text{UNF}}$ of $\mathbf{A}^{(1)}, \mathbf{A}^{(2)}$ using a GCN (Kipf and Welling, 2016) 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.

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.

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 $8 = 2^3$ possible combinations, drawing each $\mathbf{A}^{(t)}$ from each unique $\mathbf{B}^{(t)}$ as detailed in Section 2.1. The ordering of these time points is random, leading to dynamic total edge count (Figure 2). 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 (Stehlé et al., 2011). Each of the 232 pupils wore a radio identification device such that each interaction, with its

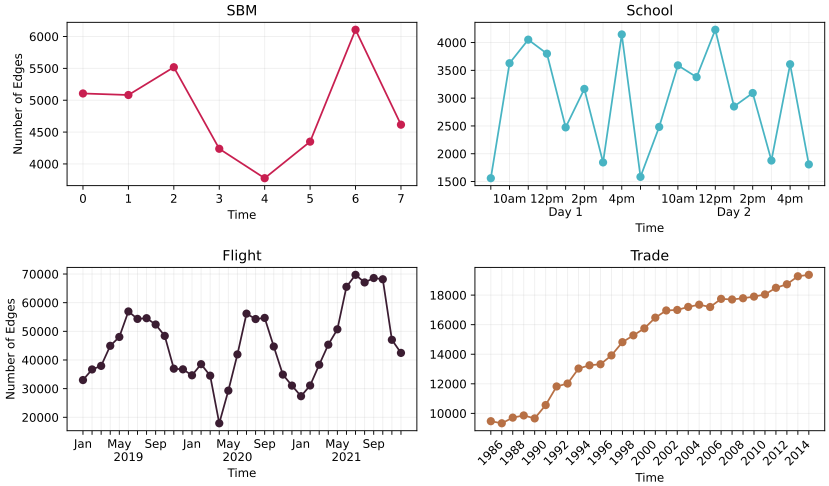


Figure 2: Numbers of edges over time. The School and Flight data show rough periodic/seasonal structure, which the Trade data features drift with the number of edges growing smoothly with time.

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 (Olive et al., 2022). 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 (MacDonald et al., 2015), which features in the Temporal Graph Benchmark (Huang et al., 2024b). 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).

3.1 Experimental Setup

On each dataset, we apply GCN (Kipf and Welling, 2016) and GAT (Veličković et al., 2017) 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 (Huang et al., 2024a) 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 (Romano et al., 2020) 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. Code to reproduce all 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.

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	**0.980 ± 0.004**	**0.985 ± 0.003**	**0.924 ± 0.009**	**0.915 ± 0.013**
Block GAT	0.916 ± 0.028	0.346 ± 0.024	0.807 ± 0.016	0.107 ± 0.024
UGAT	**0.947 ± 0.032**	**0.969 ± 0.017**	**0.896 ± 0.016**	**0.868 ± 0.017**

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

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	**0.901 ± 0.014**	0.659 ± 0.045	**0.901 ± 0.012**	0.812 ± 0.033
UGCN	**0.901 ± 0.015**	**0.918 ± 0.025**	**0.901 ± 0.012**	**0.924 ± 0.013**
Block GAT	**0.901 ± 0.015**	0.450 ± 0.154	**0.901 ± 0.012**	0.662 ± 0.084
UGAT	**0.901 ± 0.014**	**0.914 ± 0.022**	**0.901 ± 0.012**	**0.909 ± 0.021**

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

Table 3: Coverage (targeted 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 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. The problem of drift is difficult to handle for UGNN alone, but we hypothesise that downstream methods could be applied to improve coverage in these cases, for example (Barber et al., 2023) and (Clarkson, 2023). 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.

In the appendix, we include many further experiments which include a demonstration of validity in the temporal transductive regime (Appendix E), consideration of other score functions for obtaining prediction sets (Appendix G), time-conditional coverage (Appendix D), various other train/valid/calibration/test splits (Appendix F), and use of other GNN architectures (Appendix H). In each case, these further experiments bolster the claims made here; UGNN is superior in both predictive performance and in conformal metrics.

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*

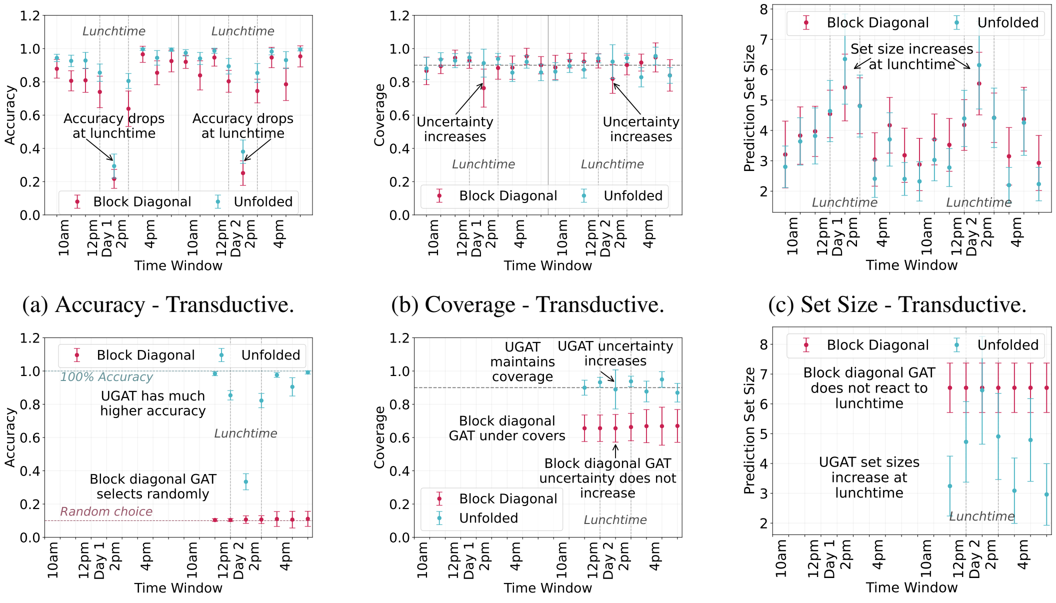
Table 4: Set sizes (lower is better) for 2 GNNs (GCN or GAT) under 2 representations (block adjacency or unfolding) for 4 datasets. Values in bold indicate a smaller set size for a given GNN. Values in italics indicate invalid sets.

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.

Figures 3a and 3d compare the 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 3a 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 (Figure 3d), 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 3c. UGAT also displays this increase in the semi-inductive regime (Figure 3f), 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. Figure 3b confirms that both methods maintain coverage in the transductive case (increasing uncertainty at lunchtimes), and UGAT maintains coverage in the semi-inductive case (Figure 3e). The story is similar for UGCN (Appendix C).



(a) Accuracy - Transductive.

(b) Coverage - Transductive.

(c) Set Size - Transductive.

(d) Accuracy - Semi-inductive.

(e) Coverage - Semi-inductive.

(f) Set Size - Semi-inductive.

Figure 3: Performance metrics 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. Prediction set sizes increase at lunchtime, with only UGAT set sizes reacting in the semi-inductive case. 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.

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.

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.

The raw computational challenge of unfolding is modest (a factor of 2) but typical tricks for mega-scale analyses, for example loading time-points into memory independently, cannot be applied. However, the embedding of the behaviour of a node over time (‘rows’ of $\mathcal{A}$) is in principle amenable to acceleration. Specifically, the behaviour of a node at only a few representative times may be needed to anchor behaviour at a target time. Therefore acceleration of unfolded GNNs to mega-scale is plausible for future work (e.g. Gao et al. (2024a;b)).

There are also opportunities to improve the downstream treatment of embeddings, such as deploying the CF-GNN algorithm (Huang et al., 2024a) to reduce prediction set sizes, or weighting calibration points according to their exchangeability with the test point (Barber et al., 2023; Clarkson, 2023).

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.

Acknowledgements

Ed Davis gratefully acknowledges support by LV= Insurance (Allianz Group) and the Centre for Doctoral Training in Computational Statistics and Data Science (Compass, EPSRC Grant number EP/S023569/1). Compass is funded by United Kingdom Research and Innovation (UKRI) through the Engineering and Physical Sciences Research Council (EPSRC), https://www.ukri.org/councils/epsrc. This work was carried out using the computational facilities of the Advanced Computing Research Centre, University of Bristol - http://www.bris.ac.uk/acrc/. PR-D gratefully acknowledges support from the NeST programme grant, EPSRC grant number EP/X002195/1.

References

Open graph benchmark. https://ogb.stanford.edu/. Accessed: 2024-05-22.

PyG documentation. https://pytorch-geometric.readthedocs.io/en/latest/. Accessed: 2024-05-22.

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.

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

Anastasios Angelopoulos, Stephen Bates, Jitendra Malik, and Michael I Jordan. Uncertainty sets for image classifiers using conformal prediction. *arXiv preprint arXiv:2009.14193*, 2020.

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

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.

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

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

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.

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.

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

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. doi: 10.1214/18-AOS1752. URL https://doi.org/10.1214/18-AOS1752.

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

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.

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.

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.

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.

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.

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

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.

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. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/5446f217e9504bc593ad9dcf2ec88dda-Paper.pdf.

Shihong Gao, Yiming Li, Yanyan Shen, Yingxia Shao, and Lei Chen. ETC: Efficient training of temporal graph neural networks over large-scale dynamic graphs. *Proc. VLDB Endow.*, 17(5): 1060–1072, May 2024a. ISSN 2150-8097. doi: 10.14778/3641204.3641215. URL https://doi.org/10.14778/3641204.3641215.

Shihong Gao, Yiming Li, Xin Zhang, Yanyan Shen, Yingxia Shao, and Lei Chen. Simple: Efficient temporal graph neural network training at scale with dynamic data placement. *Proc. ACM Manag. Data*, 2(3), May 2024b. doi: 10.1145/3654977. URL https://doi.org/10.1145/3654977.

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

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.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

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

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.

Jianguo Huang, Huajun Xi, Linjun Zhang, Huaxiu Yao, Yue Qiu, and Hongxin Wei. Conformal prediction for deep classifier via label ranking. *arXiv preprint arXiv:2310.06430*, 2023.

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, 2024a.

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, 2024b.

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

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.

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

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.

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

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. doi: 10.1080/01621459.2017.1307116.

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.

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.

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.

Xavier Olive, Martin Strohmeier, and Jannis Lübbe. Crowdsourced air traffic data from The OpenSky Network 2020 [CC-BY], January 2022. URL `https://doi.org/10.5281/zenodo.5815448`.

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.

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.

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

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.

Benjamin Sanchez-Lengeling, Emily Reif, Adam Pearce, and Alexander B. Wiltschko. A gentle introduction to graph neural networks. *Distill*, 2021. doi: 10.23915/distill.00033. https://distill.pub/2021/gnn-intro.

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.

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

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.

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.

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.

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.

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

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

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

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.

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.

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.

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.

Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In *International conference on machine learning*, pages 5453–5462. PMLR, 2018.

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.

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

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

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 (Barber et al., 2023), 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 $\{1, \ldots, m\}$.

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 $r = (r_1, \ldots, r_m)$ of real values, which will act as non-conformity scores.

Algorithm 2 Full conformal inference

```
Input: Dynamic graph $G$, index set $\nu$, attributes $(X_w, w \in \nu)$, labels $(Y_w, w \in \nu \setminus \nu_K)$, confidence level $\alpha$, algorithm $\mathcal{A}$, query point $K$
Let $\hat{C} = \mathcal{Y}$
for $y \in \mathcal{Y}$ do
    Let
        $X = (X_{\nu_1}, \ldots, X_{\nu_{|\nu|}}); \quad Y^+ = (Y_{\nu_1}, \ldots, Y_{\nu_{K-1}}, y, Y_{\nu_{K+1}}, \ldots, Y_{\nu_{|\nu|}}).$
    Compute
        $(R_1, \ldots, R_m) = \mathcal{A}(G, X, Y^+).$
    Remove $y$ from $\hat{C}$ if
        $R_K$ among $\lfloor \alpha(m) \rfloor$ largest of $R_1, \ldots, R_m$.
end for
Output: Prediction set $\hat{C}$
```

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

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

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

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).

Algorithm 3 Split conformal inference (abstract version)

```
Input: Dynamic graph $G$, index set $\nu$, attributes $(X_w, w \in \nu)$, labels $(Y_w, w \in \nu \setminus \nu_K)$, confidence level $\alpha$, prediction function $\hat{f}$, non-conformity function $r$, query point $K$
Let $\hat{C} = \mathcal{Y}$
Let $R_i = r(\hat{f}(\nu_i), Y_{\nu_i})$ for $i \in [m] \setminus K$
Let
    $\hat{q} = \lfloor \alpha(m) \rfloor$ largest value in $R_i, i \in [m] \setminus K$
for $y \in \mathcal{Y}$ do
    Let $R_K = r(\hat{f}(\nu_K), y)$
    Remove $y$ from $\hat{C}$ if $R_K \geq \hat{q}$
end for
Output: Prediction set $\hat{C}$
```

B.1.2 Split conformal prediction

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 $Y_{\nu_\ell}, \ell > m$, 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

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

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 $Y_{\nu_\ell}, \ell > m$ are being used for training, and $Y_{\nu_\ell}, \ell \in [m] \setminus K$ 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) $K = 1$.

A3. *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 1 and 3, the prediction set output by Algorithm 2 (and so Algorithm 3) is valid even if $K = 1$ deterministically.*

Proof. The combination of assumptions 1 and 3 ensures that $R_1, \ldots, R_m$ are exchangeable, and therefore the event

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

occurs with probability

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

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

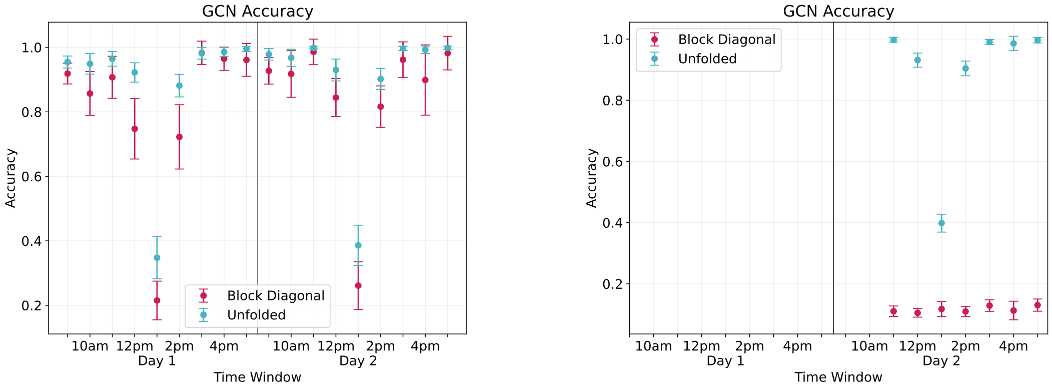
$$\mathbb{P}(Y_{\nu_1} \in \hat{C}) = 1 - \mathbb{P}(Y_{\nu_K} \notin \hat{C}) = 1 - \mathbb{P}(E) \geq 1 - \alpha.$$

□

To map Theorem 2 to Theorem 1, we observe that Assumptions 1 and 1 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 3.

C Temporal Analysis using GCN

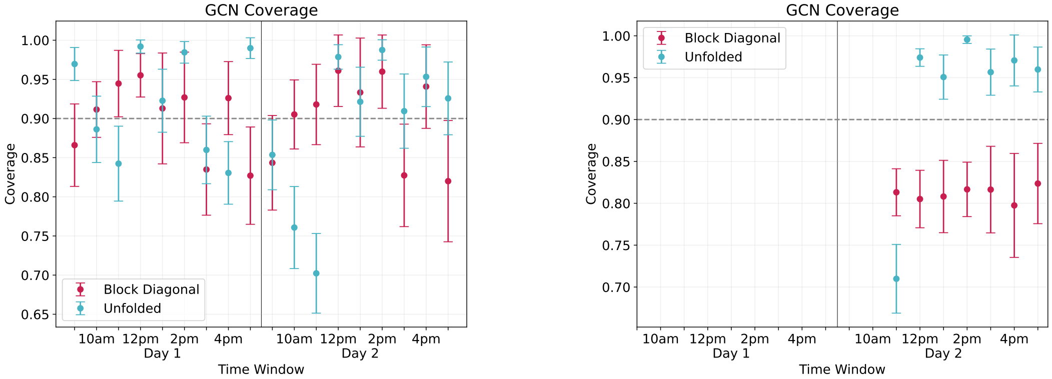
In this section, we present a temporal analysis of the school data example using GCN instead of GAT as the GNN model. We see similar conclusions to those stated in Section 3. The accuracy of UGCN is slightly higher at every time point in the transductive case and much higher in the semi-inductive case. 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.

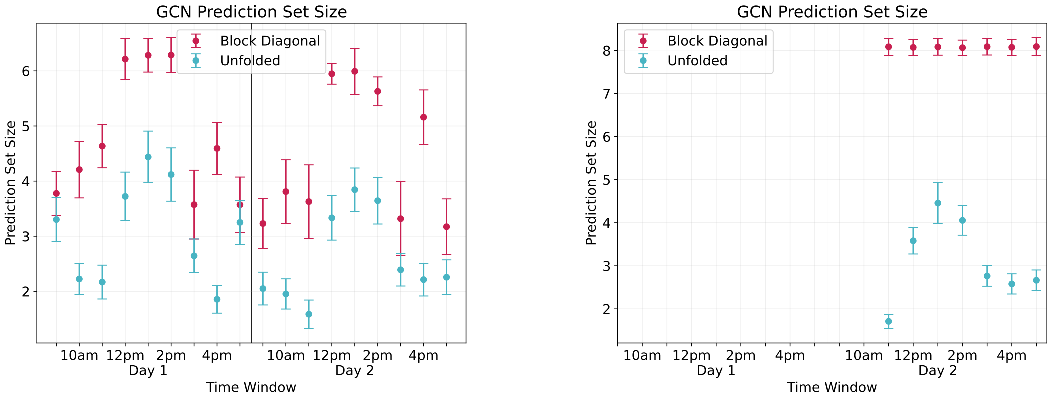
Figure 4: Prediction accuracy for each time window of the school dataset for unfolded GCN and block diagonal GCN. The prediction task gets more difficult at lunchtime, as shown by the drop in accuracy of both methods in the transductive case. UGCN has marginally better performance in the transductive case and significantly better performance in the semi-inductive case.



(a) Transductive.

(b) Semi-inductive.

Figure 5: 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 6: 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.

D Time-conditional Coverage

When evaluating a conformal procedure, it is common to consider conditional coverage (Angelopoulos and Bates, 2021; Huang et al., 2024a; Lei and Wasserman, 2014). A procedure with conditional coverage would be very adaptable, as it would return prediction sets with coverage $1 - \alpha$ conditional on the test point. While this property is desirable, it is known to be impossible to achieve exact distribution-free conditional coverage (Lei and Wasserman, 2014). Despite this, it is common to evaluate approximate conditional coverage by looking at the worst-case coverage across certain features or groups (Angelopoulos and Bates, 2021). For the purposes of this paper, we evaluate the *time-conditional* coverage, in other words, the worst-case coverage for a single time point out of all time points in the series.

Table 5 displays the time-conditional coverage for each method. While UGAT is often valid, it is not generally true that either block GNN or UGNN achieves validity here. The reason for this is again rooted in data 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.

Despite this, let us consider an example with no drift between time points. Consider another three-community DSBM with inter-community edge probability matrix

$$\mathbf{B} = \begin{bmatrix} 0.16 & 0.02 & 0.02 \\ 0.02 & 0.08 & 0.02 \\ 0.02 & 0.02 & 0.16 \end{bmatrix},$$

and draw each $\mathbf{A}^{(t)}$ from $\mathbf{B}$. This means that the series is simply a list of i.i.d. draws and so features no data drift. Table 6 displays the result of running our experiments on this data. As expected, each method achieved value time-conditional coverage in the transductive and temporal transductive regimes as drift between time points is no longer a problem. However, even with such a basic data example, it is only the UGNNs that are able to achieve valid temporal-conditional coverage here.

Methods	SBM		School	
	Trans.	Semi-ind.	Trans.	Semi-ind.
Block GCN	0.783 ± 0.091	0.645 ± 0.055	0.820 ± 0.077	0.797 ± 0.062
UGCN	0.806 ± 0.044	0.869 ± 0.030	0.702 ± 0.051	0.710 ± 0.041
Block GAT	**0.856 ± 0.068**	0.450 ± 0.154	0.763 ± 0.114	0.656 ± 0.083
UGAT	**0.877 ± 0.040**	**0.907 ± 0.031**	0.828 ± 0.058	**0.870 ± 0.056**

Methods	Flight		Trade	
	Trans.	Semi-ind.	Trans.	Semi-ind.
Block GCN	0.881 ± 0.011	0.852 ± 0.013	0.801 ± 0.064	0.767 ± 0.044
UGCN	0.877 ± 0.011	0.894 ± 0.004	0.800 ± 0.094	0.774 ± 0.056
Block GAT	0.885 ± 0.011	0.861 ± 0.014	0.822 ± 0.061	0.776 ± 0.046
UGAT	**0.889 ± 0.009**	**0.896 ± 0.004**	0.784 ± 0.076	0.792 ± 0.065

Table 5: Time-conditional coverage over time with target ≥ 0.90. Values in bold indicate valid coverage.

Methods	SBM (i.i.d.)		
	Trans.	Temp. Trans.	Semi-Ind.
Block GCN	**0.875 ± 0.063**	**0.982 ± 0.018**	0.653 ± 0.072
UGCN	**0.880 ± 0.031**	**0.881 ± 0.048**	**0.891 ± 0.010**
Block GAT	**0.862 ± 0.079**	**0.976 ± 0.022**	0.449 ± 0.158
UGAT	**0.883 ± 0.031**	**0.887 ± 0.045**	**0.885 ± 0.016**

Table 6: Time-conditional coverage on an i.i.d. version of the SBM experiment. Bold values indicate validity with target ≥ 0.90.

E Temporal Transductive

As mentioned in Section 2, and visualised in Figure 1b (bottom), the temporal transductive regime represents the case where the train and validation sets are selected randomly from data at times $t < T$, and calibration and test sets are selected randomly from $t \geq T$. This regime exists as a more practical version of the standard transductive regime, despite, theoretically, requiring no extra assumptions for valid conformal sets. However, this regime presents more of a challenge in comparison to the standard transductive case as we require a model that is trained on $t < T$ to generalise to $t \geq T$.

Table 7 displays the results of running our experiments in this regime. As predicted by our theory, and similarly to the transductive experiments, every method (including the block diagonal methods) achieves valid coverage here. However, as only the unfolded methods are exchangeable over time, the UGNN methods have a significantly higher accuracy on the datasets with out drift. Due to this, the average prediction set size is also reduced for the UGNNs.

Methods	SBM	School	Flight	Trade
Block GCN	0.318 ± 0.036	0.118 ± 0.025	0.103 ± 0.063	**0.052 ± 0.014**
UGCN	**0.984 ± 0.010**	**0.925 ± 0.020**	**0.464 ± 0.075**	0.043 ± 0.021
Block GAT	0.355 ± 0.046	0.104 ± 0.032	0.106 ± 0.064	0.049 ± 0.015
UGAT	**0.980 ± 0.012**	**0.903 ± 0.028**	**0.420 ± 0.064**	**0.051 ± 0.016**

(a) Accuracy. Bold values indicate higher accuracy.

Methods	SBM	School	Flight	Trade
Block GCN	**0.985 ± 0.011**	**0.911 ± 0.026**	**0.900 ± 0.005**	**0.901 ± 0.014**
UGCN	**0.904 ± 0.028**	**0.901 ± 0.027**	**0.900 ± 0.005**	**0.901 ± 0.015**
Block GAT	**0.977 ± 0.014**	**0.931 ± 0.033**	**0.900 ± 0.005**	**0.901 ± 0.015**
UGAT	**0.905 ± 0.027**	**0.902 ± 0.026**	**0.900 ± 0.005**	**0.901 ± 0.015**

(b) Coverage. Bold values indicate validity with target ≥ 0.90.

Methods	SBM	School	Flight	Trade
Block GCN	2.952 ± 0.018	9.069 ± 0.186	30.075 ± 2.783	**111.315 ± 10.401**
UGCN	**1.170 ± 0.199**	**3.718 ± 0.259**	**21.466 ± 2.226**	112.427 ± 8.300
Block GAT	2.933 ± 0.031	9.153 ± 0.322	30.842 ± 2.519	115.805 ± 8.289
UGAT	**0.933 ± 0.036**	**4.488 ± 0.698**	**23.626 ± 2.399**	**111.892 ± 7.711**

(c) Average set size. Bold values indicate smaller set sizes.

Methods	SBM	School	Flight	Trade
Block GCN	**0.985 ± 0.017**	**0.898 ± 0.072**	**0.899 ± 0.017**	**0.858 ± 0.055**
UGCN	**0.858 ± 0.080**	0.654 ± 0.104	**0.884 ± 0.017**	0.842 ± 0.049
Block GAT	**0.975 ± 0.024**	**0.914 ± 0.083**	**0.898 ± 0.017**	**0.850 ± 0.055**
UGAT	**0.870 ± 0.058**	**0.832 ± 0.084**	**0.891 ± 0.017**	**0.844 ± 0.068**

(d) Time-conditional coverage. Bold values indicate validity with target ≥ 0.90.

Table 7: Experiments run in the temporal transductive regime.

F Repeat Experiments with Different Data Splits

Throughout this work, we have used a standard 20/10/35/35 train/validation/calibration/testing data split. To explore the sensitivity of our results to data split selection, we carry out a repeat of our experiments using six different data split ratios. For the sake of computational efficiency, we repeat these experiments using only the school dataset.

Across the different data splits, we almost always achieve the same results as those in the main text. While there is one case of UGCN just missing out on valid coverage, and a few cases of block GCN achieving valid coverage in the semi-inductive regime, we never see block GNN achieving valid coverage when UGNN does not. Therefore, UGNN appears to be the favoured method, regardless of data split choice.

Embedding	School		
	Trans.	Semi-ind.	Temp. Trans.
Block GCN	**0.901 ± 0.014**	0.833 ± 0.046	**0.909 ± 0.019**
UGCN	**0.901 ± 0.014**	**0.943 ± 0.018**	**0.903 ± 0.014**
Block GAT	**0.902 ± 0.014**	0.681 ± 0.102	**0.916 ± 0.026**
UGAT	**0.901 ± 0.014**	**0.912 ± 0.024**	**0.901 ± 0.014**

(a) Coverage for the School experiment with data split 25/25/25/25.

Embedding	School		
	Trans.	Semi-ind.	Temp. Trans.
Block GCN	**0.903 ± 0.015**	**0.826 ± 0.075**	**0.913 ± 0.020**
UGCN	**0.902 ± 0.016**	**0.939 ± 0.020**	**0.902 ± 0.016**
Block GAT	**0.904 ± 0.015**	0.692 ± 0.109	**0.926 ± 0.029**
UGAT	**0.903 ± 0.016**	**0.901 ± 0.035**	**0.902 ± 0.015**

(b) Coverage for the School experiment with data split 50/10/20/20.

Embedding	School		
	Trans.	Semi-ind.	Temp. Trans.
Block GCN	**0.901 ± 0.011**	0.822 ± 0.055	**0.905 ± 0.016**
UGCN	**0.900 ± 0.011**	0.867 ± 0.022	**0.899 ± 0.011**
Block GAT	**0.901 ± 0.011**	0.628 ± 0.099	**0.909 ± 0.017**
UGAT	**0.901 ± 0.011**	**0.909 ± 0.021**	**0.901 ± 0.011**

(c) Coverage for the School experiment with data split 10/10/40/40.

Embedding	School		
	Trans.	Semi-ind.	Temp. Trans.
Block GCN	**0.901 ± 0.012**	0.823 ± 0.049	**0.906 ± 0.017**
UGCN	**0.901 ± 0.012**	**0.889 ± 0.020**	**0.903 ± 0.012**
Block GAT	**0.901 ± 0.012**	0.654 ± 0.093	**0.914 ± 0.022**
UGAT	**0.901 ± 0.012**	**0.907 ± 0.024**	**0.901 ± 0.012**

(d) Coverage for the School experiment with data split 20/10/35/35.

Embedding	School		
	Trans.	Semi-ind.	Temp. Trans.
Block GCN	**0.906 ± 0.022**	**0.845 ± 0.070**	**0.918 ± 0.031**
UGCN	**0.905 ± 0.022**	**0.903 ± 0.030**	**0.906 ± 0.026**
Block GAT	**0.906 ± 0.022**	0.705 ± 0.115	**0.946 ± 0.046**
UGAT	**0.906 ± 0.022**	**0.876 ± 0.042**	**0.907 ± 0.026**

(e) Coverage for the School experiment with data split 50/30/10/10.

Embedding	School		
	Trans.	Semi-ind.	Temp. Trans.
Block GCN	**0.901 ± 0.013**	**0.870 ± 0.039**	**0.910 ± 0.017**
UGCN	**0.901 ± 0.013**	**0.897 ± 0.020**	**0.902 ± 0.014**
Block GAT	**0.901 ± 0.013**	0.786 ± 0.069	**0.908 ± 0.019**
UGAT	**0.902 ± 0.013**	**0.908 ± 0.023**	**0.901 ± 0.013**

(f) Coverage for the School experiment with data split 5/35/30/30.

Table 8: Coverage for repeated runs of the School experiment with various data splits. Bold values indicate valid coverage with target ≥ 0.9.

Embedding	School		
	Trans.	Semi-ind.	Temp. Trans.
Block GCN	0.888 ± 0.013	0.109 ± 0.025	0.112 ± 0.010
UGCN	**0.935 ± 0.007**	**0.896 ± 0.010**	**0.938 ± 0.006**
Block GAT	0.834 ± 0.021	0.108 ± 0.028	0.106 ± 0.018
UGAT	**0.908 ± 0.019**	**0.883 ± 0.014**	**0.904 ± 0.024**

(a) Accuracy for the School experiment with data split 25/25/25/25.

Embedding	School		
	Trans.	Semi-ind.	Temp. Trans.
Block GCN	0.911 ± 0.010	0.108 ± 0.035	0.118 ± 0.015
UGCN	**0.931 ± 0.010**	**0.971 ± 0.008**	**0.923 ± 0.009**
Block GAT	0.880 ± 0.016	0.111 ± 0.034	0.095 ± 0.032
UGAT	**0.912 ± 0.017**	**0.962 ± 0.010**	**0.908 ± 0.014**

(b) Accuracy for the School experiment with data split 50/10/20/20.

Embedding	School		
	Trans.	Semi-ind.	Temp. Trans.
Block GCN	0.787 ± 0.020	0.105 ± 0.019	0.110 ± 0.010
UGCN	**0.928 ± 0.009**	**0.913 ± 0.014**	**0.924 ± 0.006**
Block GAT	0.697 ± 0.031	0.102 ± 0.023	0.108 ± 0.015
UGAT	**0.880 ± 0.019**	**0.903 ± 0.016**	**0.876 ± 0.014**

(c) Accuracy for the School experiment with data split 10/10/40/40.

Embedding	School		
	Trans.	Semi-ind.	Temp. Trans.
Block GCN	0.871 ± 0.010	0.105 ± 0.024	0.113 ± 0.009
UGCN	**0.933 ± 0.006**	**0.913 ± 0.009**	**0.953 ± 0.006**
Block GAT	0.808 ± 0.018	0.107 ± 0.025	0.109 ± 0.010
UGAT	**0.901 ± 0.021**	**0.899 ± 0.012**	**0.910 ± 0.014**

(d) Accuracy for the School experiment with data split 20/10/35/35.

Embedding	School		
	Trans.	Semi-ind.	Temp. Trans.
Block GCN	0.910 ± 0.016	0.113 ± 0.038	0.135 ± 0.026
UGCN	**0.934 ± 0.014**	**0.996 ± 0.006**	**0.998 ± 0.002**
Block GAT	0.881 ± 0.017	0.116 ± 0.035	0.114 ± 0.041
UGAT	**0.913 ± 0.018**	**0.991 ± 0.007**	**0.979 ± 0.013**

(e) Accuracy for the School experiment with data split 50/30/10/10.

Embedding	School		
	Trans.	Semi-ind.	Temp. Trans.
Block GCN	0.583 ± 0.031	0.105 ± 0.022	0.105 ± 0.011
UGCN	**0.918 ± 0.026**	**0.897 ± 0.022**	**0.939 ± 0.009**
Block GAT	0.543 ± 0.030	0.106 ± 0.025	0.111 ± 0.010
UGAT	**0.838 ± 0.031**	**0.874 ± 0.022**	**0.877 ± 0.032**

(f) Accuracy for the School experiment with data split 5/35/30/30.

Table 9: Accuracy values (higher is better) for repeated runs of the School experiment with various data splits. Bold values indicate the highest accuracy for a given GNN/representation pair.

G Repeat Experiments using Different Score Functions

Up to now, we have used adaptive prediction sets (APS) (Romano et al., 2020) to get our prediction sets. However, our theoretical results are not limited to a single choice of conformal procedure. To highlight this, we include experiments where we use the APS, RAPS (Angelopoulos et al., 2020) and SAPS (Huang et al., 2023) methods to compute prediction sets.

As both RAPS and SAPS require the choice of a hyperparameter, we follow Huang et al. (2023) in holding out 20% of the calibration set to select the best hyperparameter for these methods. As the school dataset is the smallest real-world dataset considered in this work, we run these extra experiments on this dataset alone.

In almost all cases, both block GNN and UGNN achieve valid coverage in the two transductive regimes, with UGNN also mostly being valid in the semi-inductive regime.

Embedding	School		
	Trans.	Semi-ind.	Temp. Trans.
Block GCN	**0.901 ± 0.012**	0.821 ± 0.049	**0.913 ± 0.024**
UGCN	**0.901 ± 0.012**	**0.889 ± 0.020**	**0.902 ± 0.016**
Block GAT	**0.901 ± 0.012**	0.665 ± 0.097	**0.922 ± 0.027**
UGAT	**0.901 ± 0.012**	**0.907 ± 0.023**	**0.903 ± 0.016**

(a) Coverage for the School experiment using APS.

Embedding	School		
	Trans.	Semi-ind.	Temp. Trans.
Block GCN	**0.902 ± 0.013**	0.821 ± 0.049	**0.915 ± 0.027**
UGCN	**0.901 ± 0.012**	**0.890 ± 0.020**	**0.903 ± 0.017**
Block GAT	**0.902 ± 0.012**	0.665 ± 0.096	**0.921 ± 0.029**
UGAT	**0.901 ± 0.013**	**0.907 ± 0.024**	**0.904 ± 0.017**

(b) Coverage for the School experiment using RAPS.

Embedding	School		
	Trans.	Semi-ind.	Temp. Trans.
Block GCN	**0.891 ± 0.014**	0.817 ± 0.050	0.226 ± 0.018
UGCN	**0.897 ± 0.013**	0.875 ± 0.023	**0.901 ± 0.017**
Block GAT	0.855 ± 0.022	0.651 ± 0.098	0.169 ± 0.097
UGAT	**0.895 ± 0.017**	**0.900 ± 0.025**	**0.898 ± 0.020**

(c) Coverage for the School experiment using SAPS.

Table 10: Coverage values for of the School experiment using different conformal score functions. Values in bold indicate valid coverage with target ≥ 0.90.

Embedding	School		
	Trans.	Semi-ind.	Temp. Trans.
Block GCN	5.195 ± 0.156	*8.174 ± 0.386*	9.121 ± 0.254
UGCN	**3.541 ± 0.314**	**2.714 ± 0.326**	**3.719 ± 0.239**
Block GAT	**4.367 ± 0.798**	*6.567 ± 0.891*	9.124 ± 0.310
UGAT	4.480 ± 0.882	**3.024 ± 0.508**	**4.503 ± 0.683**

(a) Average set sizes for the School experiment using APS.

Embedding	School		
	Trans.	Semi-ind.	Temp. Trans.
Block GCN	5.205 ± 0.169	*8.166 ± 0.393*	9.135 ± 0.273
UGCN	**3.549 ± 0.318**	**2.728 ± 0.336**	**3.735 ± 0.250**
Block GAT	**4.377 ± 0.802**	*6.565 ± 0.887*	9.117 ± 0.322
UGAT	4.486 ± 0.891	**3.029 ± 0.501**	**4.516 ± 0.696**

(b) Average set sizes for the School experiment using RAPS.

Embedding	School		
	Trans.	Semi-ind.	Temp. Trans.
Block GCN	5.044 ± 0.162	*8.130 ± 0.400*	*1.972 ± 0.030*
UGCN	**3.489 ± 0.329**	*2.593 ± 0.355*	**3.719 ± 0.254**
Block GAT	*3.755 ± 0.770*	*6.419 ± 0.936*	*1.570 ± 0.886*
UGAT	**4.451 ± 0.954**	**2.982 ± 0.519**	**4.483 ± 0.740**

(c) Average set sizes for the School experiment using SAPS.

Table 11: The average prediction set size for each method using APS, RAPS and SAPS on the school dataset. Values in bold indicate the smallest valid prediction set size for a given GNN architecture. Values in italics are the set sizes for invalid prediction sets.

H Repeat Experiments using Different GNNs

In section 3 we presented experiments using two of the most popular GNN architectures: GCN (Kipf and Welling, 2016) and GAT (Veličković et al., 2017). However, our theoretical results are not limited by the choice of GNN, so long as that GNN meets assumption A2 (which is light). Below we repeat the school experiment using two different choices of GNN: GraphSAGE (Hamilton et al., 2017), and JKNet (Xu et al., 2018).

Similar to the results in the main text, we see that coverage remains valid under these GNNs. Further, we see that the unfolded embeddings out-perform the block embeddings, both in terms of accuracy and prediction set size, in both the transductive and semi-inductive regimes.

Embedding	School	
	Trans.	Semi-ind.
Block GraphSAGE	0.856 ± 0.014	0.102 ± 0.025
Unfolded GraphSAGE	**0.932 ± 0.006**	**0.917 ± 0.009**
Block JKNet	0.885 ± 0.009	0.102 ± 0.027
Unfolded JKNet	**0.930 ± 0.007**	**0.909 ± 0.009**

(a) Accuracy. Bold values indicate higher accuracy.

Embedding	School	
	Trans.	Semi-ind.
Block GraphSAGE	**0.900 ± 0.012**	**0.869 ± 0.072**
Unfolded GraphSAGE	**0.902 ± 0.011**	**0.922 ± 0.027**
Block JKNet	**0.901 ± 0.012**	**0.897 ± 0.028**
Unfolded JKNet	**0.901 ± 0.012**	**0.907 ± 0.027**

(b) Coverage. Bold values indicate validity with target ≥ 0.90.

Embedding	School	
	Trans.	Semi-ind.
Block GraphSAGE	5.350 ± 0.230	8.632 ± 0.683
Unfolded GraphSAGE	**3.879 ± 0.978**	**3.194 ± 0.659**
Block JKNet	4.724 ± 0.332	8.946 ± 0.189
Unfolded JKNet	**2.566 ± 0.292**	**2.472 ± 0.477**

(c) Average set size. Bold values indicate smaller set sizes.

Table 12: Repeated runs on the school dataset using GraphSAGE and JKNet as the choice of GNN.