

Temporal Graph Models for Forecasting Clique Tension in Evolving Communication Networks: Group Turnover and Evaluation Pitfalls

Arnav Angarkar

arnav.angarkar20@gmail.com

Indian Institute of Information Technology Dharwad

Anand P. Barangi

Indian Institute of Information Technology Dharwad

Research Article

Keywords: Organizational networks, Temporal graph neural networks, Group evolution, Clique tracking, Forecast evaluation, Enron email

Posted Date: October 27th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11508475/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Temporal Graph Models for Forecasting Clique Tension in Evolving Communication Networks: Group Turnover and Evaluation Pitfalls

Arnav Angarkar^{1*} and Anand P. Barangi²

^{1*}Department of Computer Science and Engineering, Indian Institute of Information Technology Dharwad, Dharwad, Karnataka, India.

²Department of Arts, Science and Design, Indian Institute of Information Technology Dharwad, Dharwad, Karnataka, India.

*Corresponding author(s). E-mail(s): arnav.angarkar20@gmail.com;
Contributing authors: apb4math@gmail.com;

Abstract

Clique-based conflict scores were proposed to detect silo-like structure in email networks, but only on static graphs. We ask whether next month’s clique tension can be forecast better than by carrying the current value forward (persistence). The Enron email data are split into 44 thirty-day snapshots; in each, the largest maximal cliques are linked into a conflict meta-graph and tracked by one-to-one Jaccard matching. About 60% of cliques have no match a month later, so the target is undefined for most rows: scoring dissolved cliques as zero rewards models that predict small values, while scoring only survivors uses information known after the fact. We therefore separate a deployable protocol from an oracle-conditioned one and evaluate over 24 rolling forecast origins with block-bootstrap intervals and Diebold–Mariano tests. Graph neural networks (GCN, a GraphSAGE-style encoder, a GCN with a GRU, and two-stage hurdle versions), trained with early stopping and an output that starts at persistence, end up equivalent to persistence on surviving cliques, and a plain GCN does worse. Neighbourhood information and recurrent memory add nothing measurable. With the raw target this holds for five group definitions; with a normalized target the neural models do worse, and only simple linear forecasts improve on persistence. In the deployable setting, persistence multiplied by a simple hazard-model survival probability lowers the error by about 17%. Much of the turnover reflects how groups are defined: 55% of retained cliques are exactly the recipient list of a single email.

Keywords: Organizational networks, Temporal graph neural networks, Group evolution, Clique tracking, Forecast evaluation, Enron email

1 Introduction

Email logs record who talks to whom and when. Read as a network, they give a view of how an organization actually coordinates, which is often different from the org chart (Granovetter 1973; Burt 1992; Cross and Parker 2004). One worry managers have is the *silo*: a group that talks a lot internally and very little to the rest of the company (Tsai 2002; Cross and Parker 2004). Prior work by one of us (Barangi 2025) proposed to look for silo-like structure by extracting maximal cliques from the email graph, connecting cliques that share boundary members into a conflict meta-graph, and scoring both cliques (the Inter-Clique Conflict Score, ICCS, and the tension it sums to) and people (the Silo Risk Index, SRI).

Those scores were computed on one static graph. The obvious next step is to compute them over time and try to forecast them. Temporal graph neural networks are the standard tool for this kind of problem (Skarding et al. 2021; Pareja et al. 2020; Sankar et al. 2020; Seo et al. 2018; You et al. 2022; Rossi et al. 2020; Xu et al. 2020), so our first plan was to build one on top of the clique meta-graph and show that it forecasts tension better than simple baselines.

Once cliques were followed from one 30-day window to the next, it became clear that only about two in five of them could be matched in the following window (40.3% at $\tau = 0.5$). For the remaining rows the forecasting target is not defined, because a clique with no successor has no next-month tension, and the error figures then depend heavily on whether such rows get a target of zero or are dropped from scoring. With these rows handled the same way for every model, the neural models did not improve on persistence, i.e., on carrying the current value $T^{(k)}(C)$ forward to $k+1$, even when they were trained carefully and given an output that starts at persistence and only has to learn a correction. Poursafaei et al. (2022) and the authors of the Temporal Graph Benchmark (Huang et al. 2023) report something comparable for dynamic link prediction, where memorization baselines are often competitive and changes in the evaluation protocol can reorder the methods.

Restricting the test set to the collapse months (November 2001 to March 2002) and scoring only the cliques that survive gives a different picture, in which a two-stage (hurdle) temporal GNN has a lower MSE than persistence. That number is computed on a subset defined by the very snapshot being forecast, the single-stage models it is compared with were fitted to zero-filled targets, and it comes from 35 cliques in four snapshots. Section 5.6 and Table 3 show that it does not hold once matched controls and a rolling-origin evaluation are used.

The contribution of the paper is mainly methodological. Section 3.4 formalizes the zero-target problem that arises when attributes of tracked groups are forecast, and defines a deployable protocol (Protocol A, every clique present at k) and an oracle-conditioned one (Protocol B, survivors only), each of which can mislead in its own way.

Section 5 compares neural models (an MLP, a GCN, a GraphSAGE-style encoder and a GCN with a GRU), trained with early stopping and a residual output, with the controls that studies of this kind usually leave out: a regressor fitted to survivors without a classifier, recalibrated persistence, ridge regression on the tension history, gradient-boosted trees with and without neighbour information, and persistence weighted by a survival probability from a simple hazard model. The survival forecasts are scored on their own (Table 5). All comparisons use a rolling evaluation over 24 forecast origins, 20 of them before the collapse, with block-bootstrap intervals, Diebold–Mariano tests and explicit equivalence margins, and they are repeated for five group definitions and a normalized target.

Section 5.1 measures how much of the observed turnover is produced by the construction itself, namely broadcast emails, the K' cap, the Jaccard threshold, the window length, and the use of maximal cliques rather than k -cores or Louvain communities.

Sections 5.7 and 6.2 examine what node-level SRI measures in this data. It behaves much like a count of clique memberships, and forecasts of it rank people about as well as the SRI observed in the current month.

2 Related Work

2.1 Silos and organizational networks

The idea that boundary-spanning ties matter goes back to Granovetter’s weak ties and Burt’s structural holes (Granovetter 1973; Burt 1992). Tsai (2002) showed that knowledge sharing across units depends on these informal links, and Cross and Parker (2004) turned network analysis into a practical diagnostic for organizations. The clique-conflict framework we build on (Barangi 2025) is one way of making “silo” measurable. A point we return to in Section 6 is that none of these measures has been validated against an external definition of a silo in the Enron data.

2.2 Enron data

The Enron corpus was released during the FERC investigation and prepared for research by Klimt and Yang (2004). Diesner et al. (2005) studied how its communication network changed during the crisis, and Shetty and Adibi (2004) compiled job-title information for many of the custodians. The version we use is the higher-order (simplicial) dataset of Benson et al. (2018), in which every email is a set of Enron employees. The mailboxes come from a sample of about 150 custodians, so the network is not the full company graph.

2.3 Tracking groups over time

Palla et al. (2007) tracked overlapping communities in co-authorship and phone-call networks and found that small groups persist when their membership is stable, while large groups persist when it keeps changing. Greene et al. (2010) matched communities across time steps with a Jaccard threshold, which is close to what we do. Rossetti and Cazabet (2018) review the wider field of dynamic community discovery. These papers

treat appearance, disappearance, splits and merges as basic events in group life. A related line of work predicts these events directly: [Takaffoli et al. \(2014\)](#) and [Diakidis et al. \(2015\)](#) train classifiers on community features to forecast whether a community will survive, merge, split or dissolve in the next time step. Our survival classifiers address the same question for cliques, and in our data they do only modestly better than chance.

2.4 Temporal graph learning and its evaluation

Discrete-time models combine a graph encoder with a recurrent update: GCRN ([Seo et al. 2018](#)), T-GCN ([Zhao et al. 2020](#)), EvolveGCN ([Pareja et al. 2020](#)), DySAT ([Sankar et al. 2020](#)) and ROLAND ([You et al. 2022](#)). Continuous-time models such as TGAT and TGN update node memories event by event ([Xu et al. 2020](#); [Rossi et al. 2020](#)). Hypergraph networks handle group interactions directly ([Feng et al. 2019](#)). [Poursafaei et al. \(2022\)](#) found that a memorization baseline, which predicts an edge whenever it has been seen before, is hard to beat on many dynamic link prediction datasets, and the Temporal Graph Benchmark ([Huang et al. 2023](#)) was designed partly with that result in mind; for its node-property tasks the reference baseline is a persistent forecast that repeats the last observed value, which is the closest analogue of our persistence baseline. Our models are a simple GCN ([Kipf and Welling 2017](#)), a GraphSAGE-style encoder that keeps separate weights for a node’s own features and for the mean of its neighbours ([Hamilton et al. 2017](#)), and a GCN followed by a GRU ([Cho et al. 2014](#)) in the GCRN family. We did not compare against EvolveGCN, DySAT or ROLAND, and we make no claim about the state of the art.

2.5 Zero-heavy targets

Data with many structural zeros has a long history in econometrics. Cragg’s two-part model ([Cragg 1971](#)) and Mullahy’s hurdle model ([Mullahy 1986](#)) separate whether an outcome is zero from how large it is when it is not. Lambert’s zero-inflated Poisson model does something similar for counts ([Lambert 1992](#)). The closest forecasting analogue is intermittent demand, where Croston’s method forecasts demand size and demand interval separately ([Croston 1972](#)). Our two-stage model is a hurdle model with a neural encoder. Nothing about it is new in itself. Scoring only the groups that survive (our Protocol B, Section 3.4) conditions on an outcome that is determined after the forecast is made; in the epidemiological and causal literature this is a form of selection or survivorship bias ([Hernán et al. 2004](#)), and it can make a forecast look better than it would be in use.

3 Data and Group Construction

3.1 Dataset

We use the email-Enron simplicial dataset ([Benson et al. 2018](#)). Each record is one email with a timestamp and the set of Enron employees on it (sender and recipients). There are 148 employee IDs, of which 143 appear in at least one email, and 10,883 emails. Of these, 7,940 (73%) have exactly two Enron participants, 431 have a single

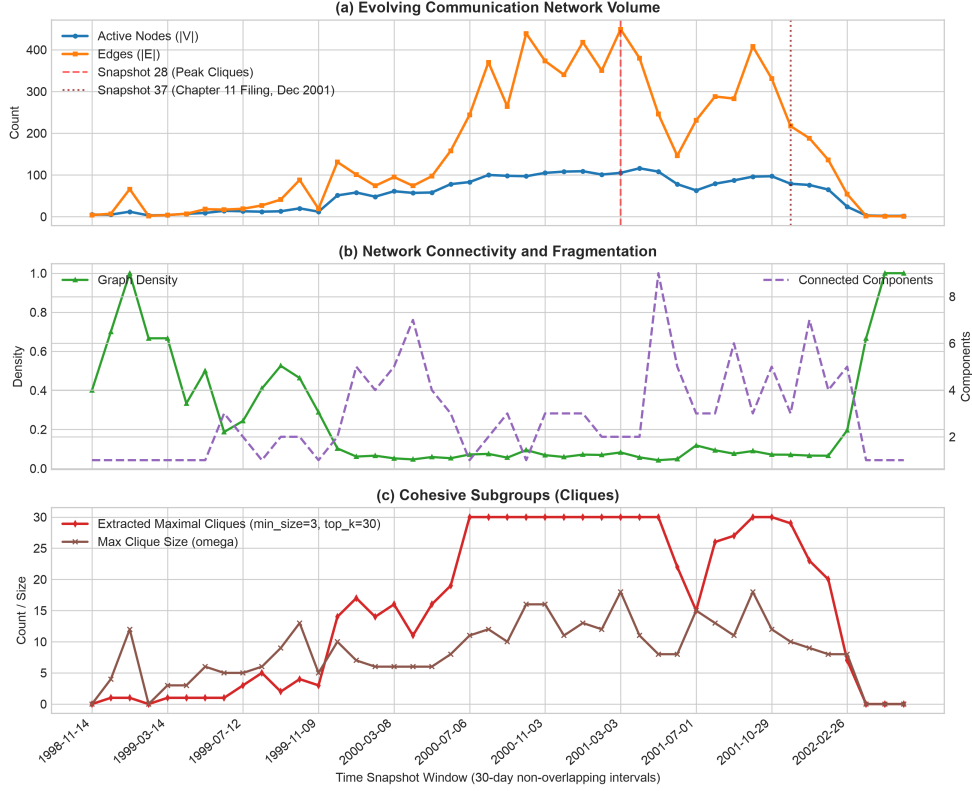


Fig. 1 The Enron email network across 44 thirty-day snapshots (November 1998 to June 2002). (a) Active employees $|V|$ and edges $|E|$. (b) Density and connected components. (c) Retained maximal cliques (capped at $K' = 30$) and the largest clique size $\omega(G)$. The cap is reached in snapshots 20–30, 35 and 36. Activity falls sharply after the December 2001 bankruptcy, and snapshots 41–43 contain no cliques.

one, and the largest has 18. Timestamps are in milliseconds from year 1 of the Gregorian calendar. We convert them to dates and sort the 115 records that are out of order. Table 1 gives summary statistics.

3.2 Snapshots and cliques

Let $\mathcal{E} = \{(S_i, t_i)\}_{i=1}^M$ be the emails, where $S_i \subseteq V$ is the set of Enron employees on email i and t_i its timestamp. We cut the time axis into T non-overlapping windows of length $\Delta_w = 30$ days, starting at the first timestamp t_0 , and build one weighted graph per window,

$$G^{(k)} = (V^{(k)}, E^{(k)}, W^{(k)}), \quad k = 0, \dots, T-1, \quad (1)$$

from the emails with $t_i \in [t_0 + k\Delta_w, t_0 + (k+1)\Delta_w)$. Each email with n participants is projected onto all $n(n-1)/2$ pairs, and the edge weight counts the emails a pair

Table 1 Summary statistics of the email-Enron simplicial data and the snapshots built from it.

Statistic	Value
Employee IDs (appearing in any email)	148 (143)
Emails	10,883
Emails with 2 / ≥ 5 Enron participants	73.0% / 6.6%
Unique employee pairs, whole period	1,800
Time span	1,307 days (Nov 1998–Jun 2002)
Non-overlapping 30-day snapshots T	44
Active employees per snapshot	55.0 ± 39.7
Edges per snapshot	163.8 ± 147.1
Maximal cliques (≥ 3), all snapshots	965
Retained cliques (top $K' = 30$ per snapshot)	689
Mean size of retained cliques	4.89
Snapshots where the K' cap binds	13

shares in the window:

$$W^{(k)}(u, v) = |\{i : t_i \in \text{window } k, \{u, v\} \subseteq S_i\}|. \quad (2)$$

In each snapshot $G^{(k)}$ we list all maximal cliques with at least three members using Bron–Kerbosch (Bron and Kerbosch 1973). If there are more than $K' = 30$, we keep the 30 largest (ties broken by member IDs). Projecting a group email onto pairs loses the information that the people were on the same message (Benson et al. 2018). We come back to this in Section 5.1, because it turns out to shape the cliques a great deal.

3.3 Conflict scores

For a clique C in snapshot k , its external boundary is the set of non-members adjacent to some member:

$$N_{\text{ext}}(C) = \{v \in V^{(k)} \setminus C \mid \exists u \in C, (u, v) \in E^{(k)}\}. \quad (3)$$

For two cliques A and B , the boundary overlap is

$$\text{ov}(A, B) = (N_{\text{ext}}(A) \cap B) \cup (N_{\text{ext}}(B) \cap A), \quad (4)$$

and the Inter-Clique Conflict Score is

$$\text{ICCS}(A, B) = \frac{|\text{ov}(A, B)|}{|A| + |B|}. \quad (5)$$

We keep the name used in (Barangi 2025), but it should not be read too literally. ICCS is high when many members of one clique are directly linked to the other clique, so it measures how tightly two groups are coupled at their boundary, not how isolated

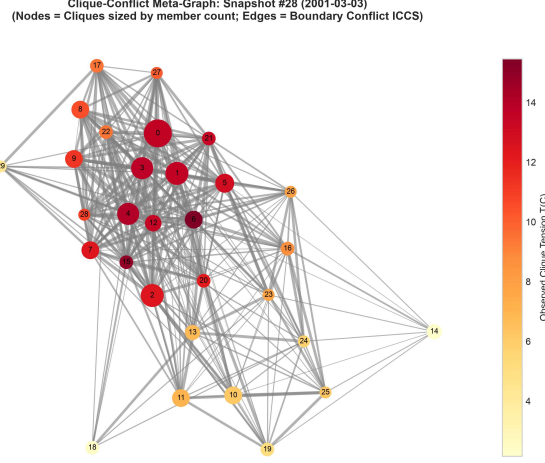


Fig. 2 Conflict meta-graph for snapshot 28 (March 2001). Nodes are retained maximal cliques, sized by membership and coloured by tension $T(C)$. Edge width is ICCS.

they are. The tension of a clique and the SRI of a person are

$$T(C) = \sum_{C' \neq C} \text{ICCS}(C, C'), \quad \text{SRI}(v) = \sum_{C \ni v} T(C). \quad (6)$$

Because $T(C)$ is a sum over all other cliques, it grows with the number of cliques in the snapshot. Across snapshots, the mean tension and the clique count have a Spearman correlation of 0.94.

The meta-graph $G_C^{(k)}$ has one node per retained clique and an edge, weighted by ICCS, between any two cliques with $\text{ICCS} > 0$. Each meta-node carries five features: clique size, conflict degree (number of meta-edges), current tension $T(C)$, mean internal edge weight, and total external edge weight. Fig. 2 shows the meta-graph at the activity peak.

3.4 Tracking and targets

Cliques have no identity across time, so we link snapshot k to $k + 1$ using the Jaccard similarity

$$J(C_i^{(k)}, C_j^{(k+1)}) = \frac{|C_i^{(k)} \cap C_j^{(k+1)}|}{|C_i^{(k)} \cup C_j^{(k+1)}|}. \quad (7)$$

Matching is one-to-one and greedy. All pairs with $J \geq \tau$ ($\tau = 0.5$) are sorted by decreasing J , and a pair is accepted if neither clique has been used yet. This gives a partial injective map π_k from cliques at k to cliques at $k + 1$. The survival indicator of clique $C_i^{(k)}$ is

$$s_i^{(k)} = \mathbf{1}[i \in \text{dom}(\pi_k)]. \quad (8)$$

where $\mathbf{1}[\cdot]$ is the indicator function (1 if the condition holds, 0 otherwise). A clique with $s_i^{(k)} = 1$ *survives*, and one with $s_i^{(k)} = 0$ *dissolves*. The same rule applied backwards (from $k - 1$ to k) gives each clique an age and a lagged tension:

$$a_i^{(k)} = \begin{cases} a_j^{(k-1)} + 1 & \text{if } \pi_{k-1}(j) = i, \\ 0 & \text{otherwise,} \end{cases} \quad (9)$$

and the lagged tension is $T^{(k-1)}(C_j^{(k-1)})$ for the matched predecessor j (set to the current tension when there is none).

Two evaluation protocols follow.

Protocol A (deployable) scores every clique present at k , with target

$$y_i^A = s_i^{(k)} \cdot T^{(k+1)}(C_{\pi_k(i)}^{(k+1)}), \quad (10)$$

so dissolved cliques get target 0. This is what a user would face, because survival is not known when the forecast is made.

Protocol B (oracle-conditioned) scores only the cliques with $s_i^{(k)} = 1$, with target $y_i^B = T^{(k+1)}(C_{\pi_k(i)}^{(k+1)})$. It answers the question “how good is the tension forecast if we already know the group will still exist?” It cannot be used in practice, because the set of survivors is defined by the future snapshot. We keep it because it isolates the regression part of the problem, and because results reported only under this protocol can look much better than they are.

4 Forecasting Setup

4.1 Evaluation design

There are 43 one-step transitions ($k \rightarrow k+1$), but snapshots 0, 3, 41, 42 and 43 contain no cliques, so 39 transitions carry targets. Our main evaluation is a **rolling origin** (Tashman 2000). For each origin $o = 16, \dots, 39$ (24 origins, 20 of them before the bankruptcy), every model is trained on the transitions before o , the last 15% of those (at least two) are held out for early stopping, and the model then forecasts transition o only. Pooled over the 24 origins this gives 614 Protocol A rows (all cliques) and 237 Protocol B rows (survivors). Tabular baselines have no early stopping and use all transitions before o .

For comparison with earlier work we also report the **chronological split** used in our first experiments: transitions 0–29 for training, 30–35 for validation and 36–42 for testing. Forecasts in this test window are made from snapshots 36–40 (29 October 2001 to 28 March 2002); transitions 41 and 42 have no cliques. Only snapshots 36–39 have survivors (35 Protocol B rows), so this window is short and covers the bankruptcy alone.

4.2 Targets and group definitions

The main target is the raw tension $T^{(k+1)}$ of Section 3.4. Since T is a sum over all other groups in the snapshot, its scale grows with the number of groups K_k , and pooled errors are dominated by busy months. We therefore also use a normalized target,

$$\tilde{T}^{(k)}(C) = \frac{T^{(k)}(C)}{\max(K_k - 1, 1)}, \quad (11)$$

which is the average ICCS of a group with the others; inputs and targets are both rescaled in this case.

Besides the default (the 30 largest maximal cliques), we repeat the comparison with four other ways of defining groups, all tracked with the same one-to-one rule: (i) maximal cliques that are not contained in the participant list of any single email, which removes cc-list cliques; (ii) maximal cliques of the graph restricted to pairs that exchanged at least two emails in the window; (iii) clique-percolation communities with $k = 3$ (Palla et al. 2007), which can overlap; and (iv) Louvain communities (Blondel et al. 2008). In each case groups have at least three members and at most 30 are kept per snapshot.

4.3 Baselines

Persistence predicts $\hat{y}_i = T^{(k)}(C_i)$. *Recalibrated persistence* predicts

$$\hat{y}_i = \max(0, \hat{a} T^{(k)}(C_i) + \hat{b}), \quad (12)$$

with $(\hat{a}, \hat{b})$ fitted by least squares on training survivors. *Ridge AR* regresses on the current tension, the lagged tension and their difference. *GBDT* is a gradient-boosted regressor (200 trees of depth 2, learning rate 0.05) on the five meta-node features and the clique age $a_i^{(k)}$. *GBDT+nbr* adds two neighbour aggregates, the ICCS-weighted mean and the maximum tension of the groups a clique is linked to, which gives the trees the same neighbourhood information a graph network sees. All regressors are fitted on survivors.

For Protocol A, survival must be forecast as well. The fair version of persistence scales the current tension by a survival probability,

$$\hat{y}_i^A = \hat{p}_i \cdot T^{(k)}(C_i), \quad (13)$$

and we try five choices of $\hat{p}_i$: the training base rate; the survival fraction observed in the previous month (a persistence forecast for survival); a logistic regression on all features; a GBDT classifier; and a simple discrete-time *hazard* model, which is a logistic regression with a separate intercept for each clique age,

$$\hat{p}_i = \sigma \left(\sum_{a=0}^3 \gamma_a \mathbf{1}[\min(a_i^{(k)}, 3) = a] + \beta^\top \mathbf{w}_i \right), \quad (14)$$

where $\mathbf{w}_i$ holds the (log) size, tension and group count, and σ is the logistic function. We also report a GBDT hurdle model (GBDT classifier times GBDT regressor) and a GBDT trained directly on the zero-heavy Protocol A target with a Poisson loss.

4.4 Neural models

Let $X^{(k)} \in \mathbb{R}^{K_k \times 5}$ hold the meta-node features of the K_k cliques at snapshot k and $A^{(k)}$ the ICCS-weighted adjacency of the meta-graph. Features are log-transformed where they are heavy-tailed and standardized with training-set means and standard deviations. Each model has an encoder that gives a hidden vector $\mathbf{h}_i^{(k)} \in \mathbb{R}^D$ for every clique, and two heads on top of it. We write layers with square brackets, $H^{[1]}, H^{[2]}$, to keep them apart from the snapshot index k .

The *GCN encoder* (Kipf and Welling 2017) uses the symmetrically normalized adjacency

$$\hat{A} = \tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2}, \quad \tilde{A} = A^{(k)} + I, \quad \tilde{D}_{ii} = \sum_j \tilde{A}_{ij}, \quad (15)$$

$$H^{[1]} = \text{Dropout}\left(\text{ReLU}\left(\hat{A}(X^{(k)}W_1 + \mathbf{b}_1)\right)\right), \quad (16)$$

$$H^{[2]} = \hat{A}(H^{[1]}W_2 + \mathbf{b}_2), \quad (17)$$

and $\mathbf{h}_i^{(k)}$ is row i of $H^{[2]}$. Our meta-graphs are dense, and two rounds of averaging can wash out a clique’s own tension, which is the strongest predictor. The *SAGE encoder* (Hamilton et al. 2017) avoids this by keeping separate weights for the clique itself and for the mean of its neighbours. With $\bar{A}$ the row-normalized adjacency (no self-loops),

$$H^{[\ell]} = \text{ReLU}\left(H^{[\ell-1]}U_\ell + \bar{A}H^{[\ell-1]}V_\ell\right), \quad H^{[0]} = X^{(k)}, \quad (18)$$

for $\ell = 1, 2$ (no ReLU after the second layer). The *MLP encoder* uses the same features with no neighbours at all.

The *GRU* adds memory over tracked cliques (Cho et al. 2014). If clique i at k is matched to clique j at $k-1$ ($\pi_{k-1}(j) = i$), its state is updated from its predecessor’s state; otherwise it starts from its encoder output:

$$\mathbf{z}_i^{(k)} = \begin{cases} \text{GRU}\left(\mathbf{h}_i^{(k)}, \mathbf{z}_j^{(k-1)}\right) & \text{if } \pi_{k-1}(j) = i, \\ \mathbf{h}_i^{(k)} & \text{otherwise,} \end{cases} \quad (19)$$

where, with $\mathbf{x} = \mathbf{h}_i^{(k)}$ and $\mathbf{z} = \mathbf{z}_j^{(k-1)}$,

$$\mathbf{r} = \sigma(W_r\mathbf{x} + U_r\mathbf{z} + \mathbf{b}_r), \quad (20)$$

$$\mathbf{u} = \sigma(W_u\mathbf{x} + U_u\mathbf{z} + \mathbf{b}_u), \quad (21)$$

$$\mathbf{n} = \tanh(W_n\mathbf{x} + \mathbf{r} \odot (U_n\mathbf{z}) + \mathbf{b}_n), \quad (22)$$

$$\text{GRU}(\mathbf{x}, \mathbf{z}) = (1 - \mathbf{u}) \odot \mathbf{n} + \mathbf{u} \odot \mathbf{z}. \quad (23)$$

Without a GRU, $\mathbf{z}_i^{(k)} = \mathbf{h}_i^{(k)}$. States are detached between snapshots, and at evaluation time the model is rolled forward from snapshot 0 so that every clique starts with its state from the past (“warm start”).

Two heads read $\mathbf{z}_i^{(k)}$. The survival head gives $\hat{p}_i = \sigma(g_p(\mathbf{z}_i^{(k)}))$, and the tension head is written as a correction to persistence,

$$\hat{T}_i = \text{Softplus}(T^{(k)}(C_i) + g_c(\mathbf{z}_i^{(k)})), \quad (24)$$

where g_p and g_c are small two-layer networks. The last layer of g_c starts at zero, so a *residual* model begins exactly at persistence and moves away from it only if that lowers the validation loss. For comparison we also train a *plain* GCN whose head is $\text{Softplus}(g_c(\mathbf{z}_i^{(k)}))$, as in the usual formulation.

Models are trained in two ways. *Cond-only* training uses the tension head alone, on survivors only:

$$\mathcal{L}_{\text{cond}} = \frac{1}{N_k} \sum_{i: s_i^{(k)}=1} (\hat{T}_i - y_i^B)^2, \quad N_k = \sum_i s_i^{(k)}. \quad (25)$$

Two-stage (hurdle) training adds the survival head with binary cross-entropy,

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{K_k} \sum_{i=1}^{K_k} \left[s_i^{(k)} \log \hat{p}_i + (1 - s_i^{(k)}) \log (1 - \hat{p}_i) \right], \quad (26)$$

$$\mathcal{L}_{\text{two}} = \mathcal{L}_{\text{BCE}} + \lambda \mathcal{L}_{\text{cond}}, \quad (27)$$

with λ equal to one over the variance of the training targets, so that the two terms have a similar size. Losses are summed over training snapshots. Under Protocol B we score $\hat{T}_i$, and under Protocol A a two-stage model outputs $\hat{y}_i^A = \hat{p}_i \cdot \hat{T}_i$.

Training. We use Adam with weight decay 10^{-4} , dropout 0.1 and up to 300 epochs, and stop when the validation loss has not improved for 30 epochs, keeping the best epoch. The hidden size (16 or 32) and learning rate (10^{-3} , 3×10^{-3} or 10^{-2}) were chosen separately for each model by validation loss on the chronological split (Appendix A). Results are averaged over five random seeds (three for the alternative group definitions).

4.5 How forecasts are compared

Our main summary is the ratio of a model’s pooled MSE to that of persistence (below 1 means better than persistence). We also give the mean and median absolute error. To judge whether a difference is real, we use two tools that respect the fact that neighbouring months are not independent. The first is an interval for the MSE ratio from a moving-block bootstrap (Künsch 1989), which resamples runs of three consecutive origins rather than single months. The second is the Diebold–Mariano test (Diebold and Mariano 1995), a standard test of whether two forecasts have the same expected error; it works on the per-origin difference in MSE, with a variance that allows for

correlation between nearby origins (Newey and West 1987) and the small-sample correction of Harvey et al. (1997). Because many models are compared with the same baseline, p -values are adjusted with the Holm procedure (Holm 1979).

A non-significant difference does not show that two models are equally good. When we claim that two forecasts are *equivalent*, we mean that the 90% bootstrap interval of their MSE ratio lies inside $[0.9, 1.1]$, i.e., that neither is more than about 10% better than the other (Lakens 2017). Survival classifiers are scored by AUROC (with a bootstrap interval over origins), the Brier score and the expected calibration error, which is the average gap between predicted and observed survival rates within probability bins.

Use of AI tools.

An AI assistant (Claude, Anthropic) was used to help write and debug the analysis code and to edit the manuscript text. The study design, all experiments and results, and the conclusions were checked by the authors, who take full responsibility for the content.

5 Results

5.1 How much of the turnover is measurement?

Of the 689 retained cliques, 278 (40.3%) have a one-to-one successor at $\tau = 0.5$ and 411 (59.7%) do not. Checking the 411 unmatched cliques against the full, untruncated clique list at $k + 1$ separates four cases: 61 (15%) had a successor with $J \geq 0.5$ that the greedy one-to-one assignment gave to a different clique; 42 (10%) had a successor that was removed by the $K' = 30$ limit; 158 (38%) had a best match with $0.3 \leq J < 0.5$, which we read as a split or a partial change of membership; and for the other 150 (37%) no clique at $k + 1$ overlapped them to that degree. Part of this comes from the threshold, which is harsh for small cliques, since a four-member clique that swaps two members has $J = 2/6 \approx 0.33$ with its successor and is counted as dissolved.

We did not anticipate how much of the clique structure comes from single emails. In 55.3% of the retained cliques the member set coincides with the Enron participant list of one email sent in that window, so the “clique” may be no more than a cc list, and it disappears from the next window unless a similar message is sent again. Capping emails at five participants removes only 11 cliques (678 remain), since most cliques are small anyway.

Table 2 and Fig. 3 vary the construction one parameter at a time. Survival drops from 56% at $\tau = 0.3$ to 22% at $\tau = 0.7$, whereas varying K' between 10 and no cap moves it only between 38% and 45%, and the window length (14, 30 or 60 days) between 36% and 42%. With overlapping windows (30/15 and 60/30 days) it is around 60%, as expected when adjacent windows share half of their emails. The other group definitions of Section 4.2 do not remove the turnover either: survival is 30.5% for cliques not contained in one email, 43.8% for cliques on repeated pairs, 32.8% for clique-percolation communities and 36.4% for Louvain communities. Clique fragility, broadcast emails and truncation thus account for part of the turnover, but none of the

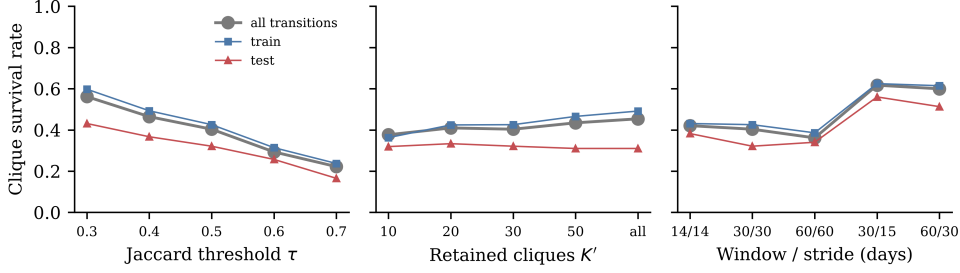


Fig. 3 Clique survival rate under different construction choices: Jaccard threshold τ (left), number of retained cliques K' (middle), and window length and stride (right). The default is $\tau = 0.5$, $K' = 30$, 30/30 days. Survival in the test period (red) is lower than in training (blue) in every setting.

Table 2 Sensitivity of turnover and of Protocol B forecasts. “Trunc.” is the share of unmatched cliques whose match was removed by the K' cut-off; “Bcast” is the share of cliques equal to a single email’s participant list. The last three columns are rolling-origin MSE (origins in the last two thirds of the series) for persistence, recalibrated persistence and ridge AR. MSE values are only comparable within a row group, since the target scale changes with the construction.

Setting		Surv.	Trunc.	Bcast	Pers.	Recal.	AR
τ	0.3	0.56	0.17	0.55	11.68	11.37	11.67
	0.5	0.40	0.10	0.55	10.75	10.48	10.33
	0.7	0.22	0.03	0.55	9.41	8.16	7.93
K'	10	0.38	0.21	0.67	2.29	2.00	1.81
	30	0.40	0.10	0.55	10.75	10.48	10.33
	all	0.45	0.00	0.52	37.25	39.68	40.80
Window/stride (days)	14/14	0.42	0.02	0.65	11.09	9.85	9.32
	60/60	0.36	0.14	0.46	18.25	18.07	19.28
	30/15	0.62	0.17	0.54	7.31	7.14	7.24
Max. participants	3	0.46	0.10	0.44	9.33	8.10	8.55
	5	0.44	0.13	0.56	7.90	6.62	6.83

definitions we tried brings survival much above one half, and in every setting survival in the test months is below that in training.

5.2 Did the neural models train properly?

With standardized inputs and early stopping, the residual models stopped after 31–106 epochs (Appendix A). On the training data of the chronological split their error on survivors is close to that of persistence (MSE between 11.9 and 12.6, against 12.9 for persistence), so they fit the data, but there is little systematic signal left to learn once the current tension is known. For the GCN and SAGE residual models the best validation loss was reached at the very first epoch, which means that every step away from persistence made validation worse. The plain GCN, which has no direct access

Table 3 Protocol B (survivors only), rolling origin over 24 origins, 237 surviving cliques, default construction, raw tension. Ratio: pooled MSE divided by that of persistence, with a 95% moving-block bootstrap interval. p : Diebold–Mariano test against persistence, Holm-adjusted. Neural models are averaged over five seeds; all are two-stage unless marked.

Model	MSE	Ratio [95% CI]	MAE	p (Holm)
Persistence	11.56	1 (reference)	2.52	–
Recalibrated persistence	11.29	0.98 [0.92, 1.03]	2.57	1.00
Ridge AR	11.00	0.95 [0.87, 1.07]	2.55	1.00
GBDT	12.92	1.12 [0.99, 1.30]	2.75	1.00
GBDT + neighbour aggregates	13.54	1.17 [1.00, 1.39]	2.83	1.00
Plain GCN	13.35	1.15 [0.96, 1.33]	2.76	1.00
Residual MLP	11.16	0.96 [0.93, 1.00]	2.49	0.29
Residual MLP, cond-only	11.29	0.98 [0.95, 1.02]	2.51	1.00
Residual GCN	11.57	1.00 [0.97, 1.03]	2.52	1.00
Residual SAGE	11.53	1.00 [0.96, 1.03]	2.54	1.00
Residual GCN+GRU	11.74	1.02 [0.98, 1.04]	2.55	1.00

to the current tension, cannot even match persistence on its own training data (MSE 14.6), which explains why models of this kind lost to persistence in the past.

5.3 Surviving cliques (Protocol B)

Table 3 gives the main comparison. Persistence has an MSE of 11.56 on survivors. The residual neural models all end up within a few percent of it: the MLP is 4% better, the GCN and SAGE models are level with it, and the GCN with a GRU is 2% worse. After the Holm adjustment none of these differences is significant, and the MLP’s interval [0.93, 1.00] only just reaches 1. Recalibrated persistence and ridge AR are also within 5% of persistence. The models that do not have the current tension as a direct input do worse: the plain GCN is 15% worse, GBDT 12% worse, and GBDT with neighbour aggregates 17% worse.

The equivalence checks of Section 4.5 make these statements more precise. The residual MLP and persistence are equivalent (ratio 0.96, 90% interval [0.94, 0.99]). The graph encoders are equivalent to the MLP (GCN 1.04 [1.02, 1.06]; SAGE 1.03 [1.02, 1.05]) and slightly worse, so neighbourhood information did not help, whether it entered through GCN averaging, through separate SAGE weights or through features for the trees. Adding a GRU did not help either (GCN+GRU against GCN, 1.01 [0.99, 1.03]). Finally, two-stage and cond-only training are equivalent (MLP 0.99 [0.97, 1.00]; GCN 1.00 [0.99, 1.01]), so the survival head of the hurdle model has no effect on the tension forecast for survivors.

5.4 Other group definitions and a normalized target

Fig. 4 and Table 4 repeat the comparison for the five group definitions, each with the raw and the normalized target. With the raw target, the residual neural models stay close to persistence for every definition: the MLP and GCN ratios lie between 0.92 and 1.09, and none of the models is significantly better than persistence after the Holm adjustment. The intervals are wide for clique percolation and Louvain communities,

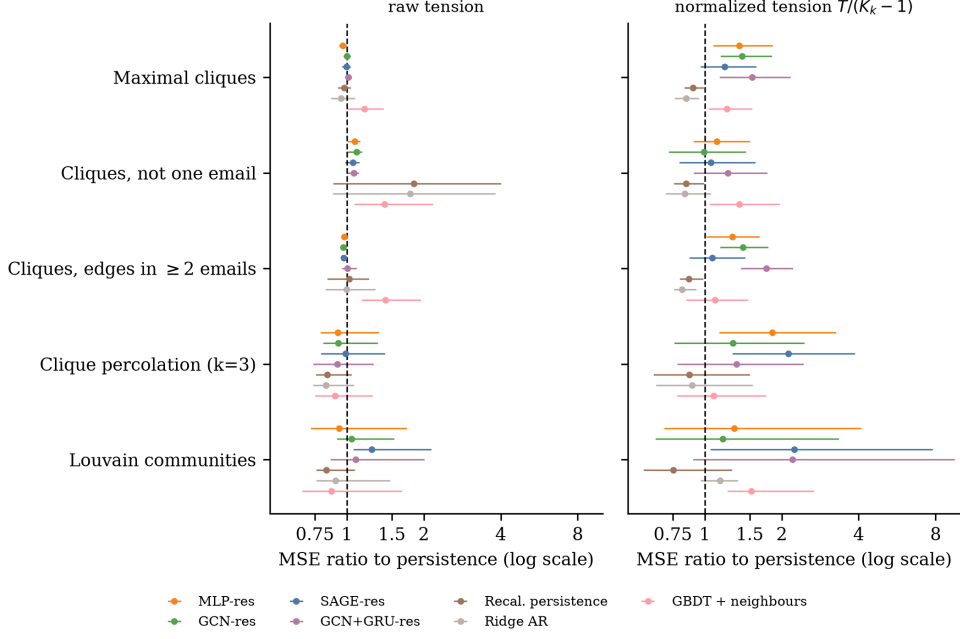


Fig. 4 Protocol B forecasts for five group definitions, with the raw target (left) and the normalized target $\tilde{T}$ (right). Each point is a model’s pooled MSE divided by that of persistence over the rolling origin, with a 95% moving-block bootstrap interval; values left of the dashed line are better than persistence.

where only 51 and 69 survivors are scored, and on Louvain communities the SAGE model is 25% worse than persistence.

The normalized target behaves differently. Here the neural models are worse than persistence in almost every case (MLP ratios 1.12 to 1.84; the one exception is the GCN on cliques not contained in one email, at 0.99), while the two simplest regressions do better: for the three clique-based definitions, recalibrated persistence and ridge AR lower the error by 10 to 18%, and their intervals mostly lie below 1, although none of these differences survives the Holm adjustment. A likely reason is that average coupling, unlike raw tension, tends to drift back towards typical values from one month to the next, which a linear fit picks up easily, while the residual networks stay near persistence because early stopping ends their training before they learn it. This is the only setting in our study where any forecast beats persistence by more than a few percent, and the forecast that does so is a two-parameter linear model.

In the deployable setting, persistence times the hazard probability beats persistence times the base rate for four of the five definitions (raw-target ratios between 0.65 and 0.92), but not for Louvain communities (1.03), where the hazard model separates survivors from dissolved groups less well (AUROC 0.61).

Table 4 MSE ratio to persistence for each group definition and target. MLP, GCN, SAGE: residual two-stage models; Recal.: recalibrated persistence; AR: ridge AR. These five columns are Protocol B (survivors). Haz. (A): Protocol A, persistence times the hazard probability relative to persistence times the base rate. n is the number of surviving cliques scored over the rolling origin.

Groups	Target	n	MLP	GCN	SAGE	Recal.	AR	Haz. (A)
Maximal cliques	raw	237	0.96	1.00	1.00	0.98	0.95	0.83
	norm.	237	1.36	1.40	1.20	0.90	0.84	0.88
Cliques, not one email	raw	126	1.07	1.09	1.06	1.83	1.77	0.92
	norm.	126	1.12	0.99	1.06	0.85	0.83	0.95
Cliques, repeated pairs	raw	201	0.98	0.97	0.97	1.02	1.00	0.88
	norm.	201	1.28	1.41	1.07	0.87	0.82	0.92
Clique percolation	raw	51	0.92	0.93	0.99	0.84	0.83	0.65
	norm.	51	1.84	1.29	2.13	0.87	0.89	0.76
Louvain	raw	69	0.93	1.04	1.25	0.83	0.90	1.03
	norm.	69	1.30	1.18	2.24	0.75	1.15	1.07

Table 5 Survival classification over the rolling origin (614 cliques, 38.6% survive). AUROC with a 95% bootstrap interval over origins; ECE is the expected calibration error. Neural rows are the survival heads of the two-stage models.

Survival forecast	AUROC [95% CI]	Brier	ECE
Training base rate	0.53 [0.49, 0.58]	0.243	0.085
Last month's survival rate	0.53 [0.49, 0.57]	0.248	0.060
Logistic regression	0.66 [0.62, 0.71]	0.227	0.067
Hazard model (age bins)	0.65 [0.60, 0.70]	0.225	0.050
GBDT classifier	0.67 [0.62, 0.72]	0.241	0.130
Residual MLP head	0.62 [0.58, 0.66]	0.238	0.112
Residual GCN head	0.55 [0.51, 0.60]	0.243	0.091
Residual SAGE head	0.57 [0.53, 0.62]	0.243	0.099
Residual GCN+GRU head	0.61 [0.57, 0.65]	0.239	0.092
Plain GCN head	0.57 [0.54, 0.61]	0.239	0.064

5.5 Survival and deployable forecasts (Protocol A)

Table 5 shows that survival is only weakly predictable. Last month's survival rate is no better than the training base rate (AUROC 0.53). The logistic, hazard and GBDT classifiers reach 0.65–0.67, with intervals well above 0.5, and the hazard model is the best calibrated (ECE 0.05). The survival heads of the neural models are weaker (0.55–0.62).

For the deployable forecasts (Table 6), persistence on its own is not a fair reference, because it predicts tension for groups that will disappear; we compare against persistence times the training base rate (MSE 19.42). Multiplying persistence by the hazard probability lowers the error by 17% (ratio 0.83, interval [0.76, 0.94]), and the logistic probability gives a similar result. Both intervals exclude 1, but after the Holm adjustment the Diebold–Mariano p -values are 0.54 and 0.56, so the evidence is suggestive rather than conclusive. The residual two-stage models gain less (ratios 0.93–1.00).

Table 6 Protocol A (all cliques, dissolved ones scored as 0), rolling origin, 614 cliques. Ratio: pooled MSE divided by that of persistence times the training base rate, with a 95% moving-block bootstrap interval. p : Diebold–Mariano test against that baseline, Holm-adjusted.

Forecast	MSE	Ratio [95% CI]	MAE	p (Holm)
Persistence $\times$ base rate	19.42	1 (reference)	3.57	–
All zeros	22.35	1.15 [0.99, 1.32]	2.42	0.46
Persistence $\times$ last month’s rate	18.47	0.95 [0.89, 1.04]	3.40	1.00
Persistence $\times$ logistic	16.48	0.85 [0.78, 0.95]	3.13	0.56
Persistence $\times$ hazard	16.20	0.83 [0.76, 0.94]	3.13	0.54
Persistence $\times$ GBDT classifier	16.84	0.87 [0.75, 1.05]	2.97	1.00
GBDT hurdle	16.86	0.87 [0.77, 1.05]	3.06	1.00
Poisson GBDT	18.55	0.96 [0.82, 1.19]	2.99	1.00
Plain GCN (two-stage)	16.68	0.86 [0.79, 0.94]	3.19	0.53
Residual MLP (two-stage)	18.12	0.93 [0.87, 1.00]	3.47	1.00
Residual GCN (two-stage)	18.86	0.97 [0.93, 1.01]	3.54	1.00
Residual SAGE (two-stage)	19.34	1.00 [0.95, 1.04]	3.57	1.00
Residual GCN+GRU (two-stage)	18.47	0.95 [0.92, 1.01]	3.45	1.00

The plain GCN does better here (0.86), but only because its forecasts are low across the board, which helps on dissolved cliques (error part 4.7) and hurts on survivors (11.9). Splitting the error into the part from survivors and the part from dissolved cliques shows where the gain comes from: for persistence times the hazard probability, the dissolved part falls from 10.8 to 5.8, while the survivor part rises only from 8.6 to 10.4. Better survival forecasts, not better tension forecasts, drive the improvement.

5.6 The collapse window

The chronological split of Section 4.1 shows how a short test window can mislead. On its 35 surviving test cliques, all from the bankruptcy months, persistence has an MSE of 5.04. The plain GCN with two-stage training reaches 2.62 ± 0.67 over five seeds, which would look like a large improvement. Over the 24 rolling origins the same model is 15% worse than persistence (Table 3). The reason is a shift in level: in the collapse months, surviving cliques carried much less tension than in earlier years (mean 4.23 against 6.17 in the training period), so any model that forecasts low does well on this window and badly elsewhere. The residual models, whose validation loss never improved much on persistence, stay close to it here as well (MSE 4.91 to 5.29), and the plain GCN trained on zero-filled targets reaches 6.69. A result reported only on this window, and only for survivors, would have pointed in the wrong direction.

5.7 Node-level SRI

Node-level risk does not depend on matching cliques across time, so it is a useful check. Each person’s predicted risk is the sum of the expected tension $\hat{p}_i \hat{T}_i$ over the cliques they belong to at k , and we compare it with the SRI actually observed at $k+1$, using Spearman correlation over people who are in a retained clique at k or $k+1$. Averaged over the 24 rolling origins, this month’s observed SRI gives $\rho = 0.30$ (95% interval [0.20, 0.41]). Persistence times the hazard probability gives 0.32 [0.22, 0.41],

and the neural models 0.29–0.30. Simpler rankings do slightly worse: the number of cliques a person belongs to gives 0.26 and degree 0.23. The intervals overlap, so no forecast ranks people clearly better than this month’s SRI. We do not compute the correlation over all 148 employees, because about 100 of them are tied at zero in any month, which inflates ρ without saying anything about risk.

6 Discussion

6.1 Summary of findings

In this dataset, carrying the current clique tension forward is as good a forecast of next month’s tension as anything we tried. Once the neural models were trained with standardized inputs, early stopping and a residual head that starts at persistence, they converged to persistence rather than beating it, and the graph and recurrent parts did not change that. Without the residual head, a GCN did worse than persistence, because two rounds of averaging over a dense meta-graph blur the clique’s own tension. These conclusions hold for a small dataset: 689 cliques in 39 usable months, with early origins trained on a few dozen survivors. A model with thousands of parameters competing against a baseline with none is at a disadvantage in this regime, and a larger network might give a different answer.

The hurdle formulation is a reasonable way to write down the problem, but its benefit on survivors comes from training on the right rows, and its survival head adds nothing to the tension forecast. What helps in practice is a decent survival probability. A simple hazard model with clique age, size and tension reaches an AUROC of 0.65 and, multiplied by persistence, gives the best deployable forecast. Intermittent-demand forecasting (Croston 1972) and dynamic-graph benchmarks (Poursafaei et al. 2022; Huang et al. 2023) point the same way: naive baselines are hard to beat and should be reported first.

6.2 Construct validity of ICCS and SRI

The present study cannot establish that these scores measure silos. By its definition in Section 3.3, ICCS increases when members of one clique are directly connected to the other clique, which is a measure of coupling at the boundary and in that sense nearly the opposite of isolation. SRI, in turn, behaves largely like a membership count: within a snapshot its Spearman correlation with the number of retained cliques a person belongs to is 0.82, and with degree 0.67. We used no external labels, although they exist both for Enron (job titles for many custodians, compiled by Shetty and Adibi (2004)) and for the email-Eu-core network (department membership; Yin et al. 2017). A check against such labels, or against an explicit operational definition of a silo, is needed before either score is used to predict silos; in this paper they serve only as descriptive measures of the boundary structure between dense groups.

6.3 Limitations

The study covers a single company, a single 30-day horizon and a single period that ends with the firm’s collapse. As the mailboxes were collected from custodians, most

Enron employees and every external contact are missing from the graph. With about 50 active nodes and at most 72 maximal cliques per snapshot, clique enumeration is cheap, and nothing here shows that the approach scales to large organizations. We tuned only two hyperparameters on one split and did not compare with EvolveGCN, DySAT, ROLAND or hypergraph networks; a more expressive model trained on more data might find signal that ours did not. Projecting emails onto pairs throws away the group structure that a hypergraph model could use, and 55% of our cliques are single emails, so tracking hyperedges directly is a natural next step. A second dataset, ideally one with department labels such as email-Eu-core, is needed to see whether these results carry over.

7 Conclusion

We set out to forecast clique tension in an evolving email network with temporal graph neural networks. Groups turn over quickly, about 60% within a month, and how dissolved groups are scored can make almost any model look good or bad. With a deployable protocol, a rolling-origin evaluation and properly trained models, the neural forecasts end up equivalent to persistence, and the graph and memory components add nothing measurable. The best usable forecast was persistence scaled by a simple survival probability. With the raw target this held for five different group definitions. With a normalized target the neural models did worse, and the only forecasts that beat persistence there were simple linear ones. Part of the turnover is itself an artifact of how groups are defined, especially single broadcast emails. Whether ICCS and SRI measure silos remains open until they are checked against labels such as departments or job titles. We hope the protocols and controls described here are useful to others who forecast properties of groups that may not exist next month.

Appendix A Training details

Table A1 lists the hyperparameters chosen for each model on the validation transitions of the chronological split, the number of epochs run before early stopping, and the training-set MSE on survivors (persistence: 12.85). Fig. A1 shows the training and validation losses.

Table A1 Selected hyperparameters and training behaviour on the chronological split (seed 42). Best epoch 0 means the validation loss never improved on the starting point, which for residual models is persistence.

Model	Hidden	Learning rate	Epochs run	Best epoch	Train MSE
Residual MLP	32	0.001	97	66	11.86
Residual GCN	16	0.001	31	0	12.55
Residual SAGE	32	0.001	31	0	12.55
Residual GCN+GRU	32	0.001	106	75	12.11
Plain GCN	16	0.01	57	26	14.57

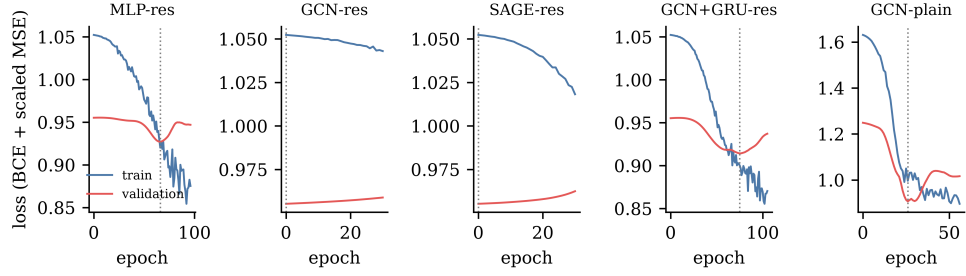


Fig. A1 Training (blue) and validation (red) loss per epoch on the chronological split for each neural model. The dotted line marks the epoch kept by early stopping.

Acknowledgements. Not applicable.

Declarations

- **Funding.** No funding was received for conducting this study.
- **Competing interests.** The authors have no relevant financial or non-financial interests to disclose.
- **Ethics approval and consent to participate.** Not applicable. The study uses only a publicly released, de-identified research dataset of email metadata (Benson et al. 2018); no human participants were recruited and no message content was used.
- **Consent for publication.** Not applicable.
- **Data availability.** This study analyses a publicly available dataset. The email-Enron simplicial dataset (email-Enron) was released by Benson et al. (2018) (<https://doi.org/10.1073/pnas.1800683115>) and is built from the Enron email corpus distributed by Carnegie Mellon University (<https://www.cs.cmu.edu/~enron/>). A copy of the dataset files used here is included in the code repository listed below (folder *Dataset/email-Enron*), together with all derived results reported in this article (files *outputs/revision/revision_results.json* and *outputs/revision2/results.json*).
- **Code availability.** All code used for data processing, experiments and figures is available at <https://github.com/ArnavBallinCode/temporal-gnn-silo-prediction>. The turnover analysis (Table 2, Fig. 3) is produced by *src/training/revision_experiments.py*; all forecasting results (Tables 3–A1, Figs. 4 and A1) by *src/training/revision2_experiments.py* and *src/training/revision2_figures.py*. Package versions are listed in *requirements.txt*, and unit tests for the matching rule, target construction and test statistics are in *tests/*.
- **Author contribution.** Arnav Angarkar: conceptualization of the temporal extension, methodology, software, formal analysis, investigation, visualization, writing (original draft). Anand P. Barangi: original clique-conflict framework (ICCS, SRI), conceptualization, supervision, writing (review and editing). Both authors read and approved the final manuscript.

References

- Barangi AP (2025) Clique-based conflict metrics for predicting organizational silos in email communication networks. Preprint
- Benson AR, Abebe R, Schaub MT, et al (2018) Simplicial closure and higher-order link prediction. *Proceedings of the National Academy of Sciences* 115(48):E11221–E11230. <https://doi.org/10.1073/pnas.1800683115>
- Blondel VD, Guillaume JL, Lambiotte R, et al (2008) Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* 2008(10):P10008. <https://doi.org/10.1088/1742-5468/2008/10/P10008>
- Bron C, Kerbosch J (1973) Algorithm 457: Finding all cliques of an undirected graph. *Communications of the ACM* 16(9):575–577. <https://doi.org/10.1145/362342.362367>
- Burt RS (1992) *Structural Holes: The Social Structure of Competition*. Harvard University Press, Cambridge, MA, <https://doi.org/10.4159/9780674029095>
- Cho K, van Merriënboer B, Gulcehre C, et al (2014) Learning phrase representations using RNN encoder–decoder for statistical machine translation. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, pp 1724–1734, <https://doi.org/10.3115/v1/D14-1179>
- Cragg JG (1971) Some statistical models for limited dependent variables with application to the demand for durable goods. *Econometrica* 39(5):829–844. <https://doi.org/10.2307/1909582>
- Cross R, Parker A (2004) *The Hidden Power of Social Networks: Understanding How Work Really Gets Done in Organizations*. Harvard Business School Press, Boston, MA
- Croston JD (1972) Forecasting and stock control for intermittent demands. *Operational Research Quarterly* 23(3):289–303. <https://doi.org/10.2307/3007885>
- Diakidis G, Karna D, Fasarakis-Hilliard D, et al (2015) Predicting the evolution of communities in social networks. In: *Proceedings of the 5th International Conference on Web Intelligence, Mining and Semantics (WIMS '15)*. ACM, pp 1–6, <https://doi.org/10.1145/2797115.2797119>
- Diebold FX, Mariano RS (1995) Comparing predictive accuracy. *Journal of Business & Economic Statistics* 13(3):253–263. <https://doi.org/10.1080/07350015.1995.10524599>
- Diesner J, Frantz TL, Carley KM (2005) Communication networks from the Enron email corpus “It’s always about the people. Enron is no different”. *Computational*

- and Mathematical Organization Theory 11(3):201–228. <https://doi.org/10.1007/s10588-005-5377-0>
- Feng Y, You H, Zhang Z, et al (2019) Hypergraph neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 3558–3565, <https://doi.org/10.1609/aaai.v33i01.33013558>
- Granovetter MS (1973) The strength of weak ties. American Journal of Sociology 78(6):1360–1380. <https://doi.org/10.1086/225469>
- Greene D, Doyle D, Cunningham P (2010) Tracking the evolution of communities in dynamic social networks. In: 2010 International Conference on Advances in Social Networks Analysis and Mining (ASONAM). IEEE, pp 176–183, <https://doi.org/10.1109/ASONAM.2010.17>
- Hamilton WL, Ying R, Leskovec J (2017) Inductive representation learning on large graphs. In: Advances in Neural Information Processing Systems 30 (NeurIPS 2017), <https://doi.org/10.48550/arXiv.1706.02216>
- Harvey D, Leybourne S, Newbold P (1997) Testing the equality of prediction mean squared errors. International Journal of Forecasting 13(2):281–291. [https://doi.org/10.1016/S0169-2070\(96\)00719-4](https://doi.org/10.1016/S0169-2070(96)00719-4)
- Hernán MA, Hernández-Díaz S, Robins JM (2004) A structural approach to selection bias. Epidemiology 15(5):615–625. <https://doi.org/10.1097/01.ede.0000135174.63482.43>
- Holm S (1979) A simple sequentially rejective multiple test procedure. Scandinavian Journal of Statistics 6(2):65–70
- Huang S, Poursafaei F, Danovitch J, et al (2023) Temporal graph benchmark for machine learning on temporal graphs. In: Advances in Neural Information Processing Systems 36 (NeurIPS 2023), Datasets and Benchmarks Track, <https://doi.org/10.48550/arXiv.2307.01026>
- Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (ICLR), <https://doi.org/10.48550/arXiv.1609.02907>
- Klimt B, Yang Y (2004) The Enron corpus: A new dataset for email classification research. In: Machine Learning: ECML 2004, Lecture Notes in Computer Science, vol 3201. Springer, Berlin, Heidelberg, pp 217–226, https://doi.org/10.1007/978-3-540-30115-8_22
- Künsch HR (1989) The jackknife and the bootstrap for general stationary observations. The Annals of Statistics 17(3):1217–1241. <https://doi.org/10.1214/aos/1176347265>

- Lakens D (2017) Equivalence tests: A practical primer for t tests, correlations, and meta-analyses. *Social Psychological and Personality Science* 8(4):355–362. <https://doi.org/10.1177/1948550617697177>
- Lambert D (1992) Zero-inflated Poisson regression, with an application to defects in manufacturing. *Technometrics* 34(1):1–14. <https://doi.org/10.2307/1269547>
- Mullahy J (1986) Specification and testing of some modified count data models. *Journal of Econometrics* 33(3):341–365. [https://doi.org/10.1016/0304-4076\(86\)90002-3](https://doi.org/10.1016/0304-4076(86)90002-3)
- Newey WK, West KD (1987) A simple, positive semi-definite, heteroskedasticity and autocorrelation consistent covariance matrix. *Econometrica* 55(3):703–708. <https://doi.org/10.2307/1913610>
- Palla G, Barabási AL, Vicsek T (2007) Quantifying social group evolution. *Nature* 446:664–667. <https://doi.org/10.1038/nature05670>
- Pareja A, Domeniconi G, Chen J, et al (2020) EvolveGCN: Evolving graph convolutional networks for dynamic graphs. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, pp 5363–5370, <https://doi.org/10.1609/aaai.v34i04.5984>
- Poursafaei F, Huang S, Pelrine K, et al (2022) Towards better evaluation for dynamic link prediction. In: *Advances in Neural Information Processing Systems* 35 (NeurIPS 2022), Datasets and Benchmarks Track, <https://doi.org/10.48550/arXiv.2207.10128>
- Rossetti G, Cazabet R (2018) Community discovery in dynamic networks: A survey. *ACM Computing Surveys* 51(2):35:1–35:37. <https://doi.org/10.1145/3172867>
- Rossi E, Chamberlain B, Frasca F, et al (2020) Temporal graph networks for deep learning on dynamic graphs. <https://doi.org/10.48550/arXiv.2006.10637>, iCML 2020 Workshop on Graph Representation Learning and Beyond. arXiv:2006.10637
- Sankar A, Wu Y, Gou L, et al (2020) DySAT: Deep neural representation learning on dynamic graphs via self-attention networks. In: *Proceedings of the 13th International Conference on Web Search and Data Mining (WSDM)*. ACM, pp 519–527, <https://doi.org/10.1145/3336191.3371845>
- Seo Y, Defferrard M, Vandergheynst P, et al (2018) Structured sequence modeling with graph convolutional recurrent networks. In: *Neural Information Processing (ICONIP 2018)*, Lecture Notes in Computer Science, vol 11301. Springer, Cham, pp 362–373, https://doi.org/10.1007/978-3-030-04167-0_33
- Shetty J, Adibi J (2004) The Enron email dataset: Database schema and brief statistical report. Tech. rep., Information Sciences Institute, University of Southern California, Marina del Rey, CA

- Skarding J, Gabrys B, Musial K (2021) Foundations and modeling of dynamic networks using dynamic graph neural networks: A survey. *IEEE Access* 9:79143–79168. <https://doi.org/10.1109/ACCESS.2021.3082932>
- Takaffoli M, Rabbany R, Zaïane OR (2014) Community evolution prediction in dynamic social networks. In: 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014). IEEE, pp 9–16, <https://doi.org/10.1109/ASONAM.2014.6921553>
- Tashman LJ (2000) Out-of-sample tests of forecasting accuracy: An analysis and review. *International Journal of Forecasting* 16(4):437–450. [https://doi.org/10.1016/S0169-2070\(00\)00065-0](https://doi.org/10.1016/S0169-2070(00)00065-0)
- Tsai W (2002) Social structure of “coopetition” within a multiunit organization: Coordination, competition, and intraorganizational knowledge sharing. *Organization Science* 13(2):179–190. <https://doi.org/10.1287/orsc.13.2.179.536>
- Xu D, Ruan C, Korpeoglu E, et al (2020) Inductive representation learning on temporal graphs. In: International Conference on Learning Representations (ICLR), <https://doi.org/10.48550/arXiv.2002.07962>
- Yin H, Benson AR, Leskovec J, et al (2017) Local higher-order graph clustering. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp 555–564, <https://doi.org/10.1145/3097983.3098069>
- You J, Du T, Leskovec J (2022) ROLAND: Graph learning framework for dynamic graphs. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. ACM, pp 2358–2366, <https://doi.org/10.1145/3534678.3539300>
- Zhao L, Song Y, Zhang C, et al (2020) T-GCN: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems* 21(9):3848–3858. <https://doi.org/10.1109/TITS.2019.2935152>