Does the Explanation Drift First? Early Warning of Graph Neural Network Degradation

Anonymous authors
Paper under double-blind review

Abstract

Machine-learning models quietly lose accuracy when data drift occurs, and the labels that would reveal it often arrive late or never. A label-free clue is the model's own explanation: when its reasoning changes, something has shifted. Prior work detects this *explanation shift* in tabular data but does not analyse whether it occurs *before* accuracy drops. This paper focuses on graphs. Lead-time between explanation drift and the model's accuracy is measured on fully controlled, synthetic graph streams. The explanation drifts first, by a median of 9 steps, with no false alarms, and a controlled sweep confirms this reflects a real mechanism, not an artefact. The lead is over accuracy, not over every detector. The effect also holds, albeit more weakly, on a realistic IT infrastructure graph (RCAEval). Explanation drift can therefore serve as a simple, label-free early warning that a graph model should be revalidated and updated to new conditions.

1 Introduction

The relevance of AI models' training may decay over time as the world changes. If unattended, accuracy often falls silently because the labels needed to detect the drop are delayed, costly, or simply absent in deployment (Demšar & Bosnić, 2018). A practical monitor must therefore raise the alarm *before* accuracy has already fallen.

Feature attributions are a natural candidate for such a monitor: they can be computed at any time, with no labels (Yang & Kim, 2019; Al Bared & Nassar, 2021). Recent work tracks their drift as a warning sign (Mougan et al., 2022; 2025; Haug et al., 2022; Duckworth et al., 2021).

Two gaps remain. First, these studies ask whether attributions have shifted, not whether that shift *precedes* the loss of accuracy, the very property that would make it an early warning. Second, they are tabular. In tabular attribution, features are fixed, independent coordinates; a graph model's prediction instead flows through message passing over a topology that is itself the object of drift. The relevant importance lies in edge and node interactions, not in a static feature vector, and drift may be structural: edges and nodes can appear, vanish, or rewire with no counterpart in the tabular setting. A drift signal for GNNs must therefore track importance over the graph's structure, which is what the focused attribution-drift metric below does. On graphs, explanation methods such as PGExplainer (Luo et al., 2020) are studied for stability and faithfulness (Agarwal et al., 2023), but never inverted into a timed signal tied to accuracy. This paper addresses both gaps and studies, on graph streams, whether explanation drift arrives before accuracy degradation, and by how much — a quantity called the *lead-time*.

This question is not merely academic; it is highly useful for IT infrastructure diagnosis. A modern system can be perceived as a graph of services, and graph models are increasingly used for automated root-cause analysis (RCA) (Pham et al., 2025). Yet infrastructure evolves constantly, services are deployed, scaled, and re-wired, so a trained model slowly goes stale, becoming less useful for its operators (Szandała, 2025b). Because faults propagate along dependency edges, and resilient systems mask a failing path with redundancy until it saturates, the model's explanation should shift before its diagnoses fail. Explanation drift then becomes an operational trigger: revalidate the RCA model before it silently misdiagnoses.

The contributions are as follows.

- The *precedence* question is stated for GNNs equipped with parametrised edge explainers, and lead-time is defined as the gap between the onset of explanation drift and the onset of accuracy degradation.
- Using synthetic graph streams with a known drift onset, a positive and significant lead-time is demonstrated at a zero-false-alarm operating point, and its connection to a staged-redundancy mechanism is established through a falsifiable dose–response sweep.
- The effect is further affirmed on a realistic IT infrastructure substrate built on the RCAEval service graph (Pham et al., 2025), where precedence persists, albeit to a smaller extent, and the reasons for this difference are explained.
- Comparisons with label-free drift-detection baselines are provided and reported in full, rather than being limited to the most favourable baselines.

2 Related Works

Attributions drift when distributions move. Mougan et al. (2022; 2025) formalise this as an *explanation shift*, detecting it from SHAP distributions. Yet they state plainly that their method detects a shift but does not predict performance loss. Also, the study focuses solely on tabular data. The wider monitoring literature (Duckworth et al., 2021; Shanbhag et al., 2021; Haug et al., 2022; LOPATECKI et al., 2024) likewise treats attributions as a label-free alarm. None poses our temporal question: does the alarm sound before accuracy falls, and if so, by how much? And none ventures onto graphs.

A second strand focuses on explaining the drift rather than just anticipating it. Hinder et al. (2023) explain drift after the fact, classifying pre- against post-drift segments. This is useful, but it works retrospectively; we should ask whether the explanation moves first.

Label-free drift detection itself is mature: the classical change-point tests DDM (Gama et al., 2004), ADWIN (Bifet & Gavaldà, 2007), and Page–Hinkley (Page, 1954), and the model-based STUDD (Cerqueira et al., 2023), all raise alarms from inputs, outputs, or a student–teacher discrepancy, without labels. This machinery is reused here (the Page–Hinkley detector of Section 3), but aimed at a new signal — graph attributions — and asked a new question: whether that signal leads the very accuracy it is meant to guard.

On graphs, the focus has been on explanation quality. PGExplainer (Luo et al., 2020) is inductive: train it once, then track masks cheaply over time. Agarwal et al. (2023) formalise stability and find PGExplainer the steadiest of the field, while others show that GNN explanations buckle under shift. Counterfactual explainers such as CF-GNNExplainer (Lucic et al., 2022) offer an alternative per-edge signal (the minimal edge deletions that flip a prediction); the monitoring scheme below requires only a trackable per-edge importance, so it is not tied to PGExplainer. Throughout this literature, however, stability is treated as a static virtue to be maximised, never inverted into a timed warning tied to accuracy.

The gap is consistent: attribution monitoring is tabular and detection-only, while graph explanation is studied for stability rather than precedence. The goal of this paper is to test whether explanation drift precedes accuracy degradation with sufficient lead-time.

3 Methodology

3.1 Overview and hypothesis

The goal is to determine whether, when the data drifts, the model's explanation drifts before its accuracy falls (as illustrated in Figure 1). This temporal gap is called the *lead-time*. The hypothesis (H1) is that attribution drift precedes accuracy degradation, and that this lead-time is measurable, robust, and mechanistically explained. H1 is validated on synthetic graph streams with a known, controlled onset of drift, so that both the true cause and the timing are available as ground truth.

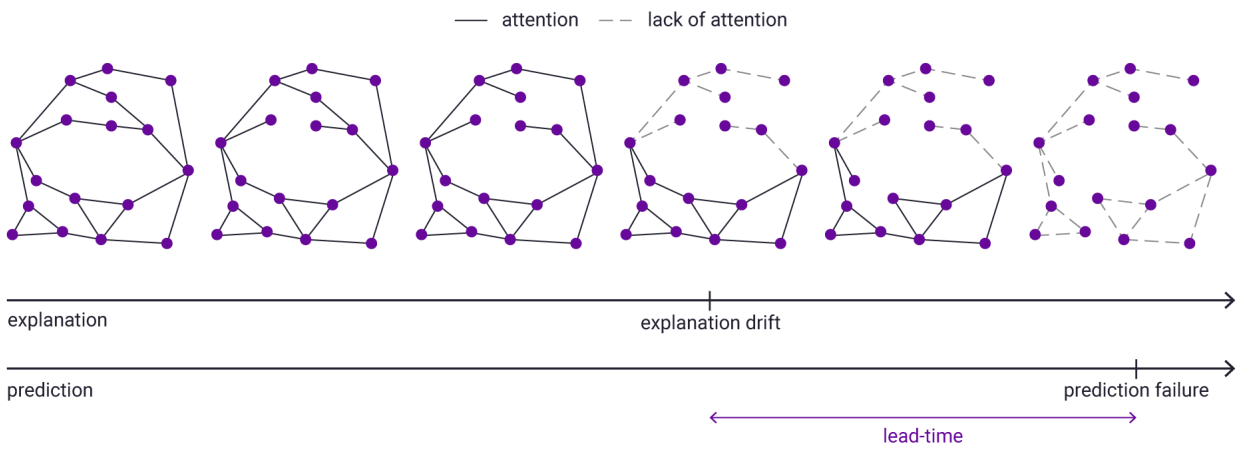

Figure 1: Conceptual illustration of staged motif degradation and the resulting lead-time: the explanation (edge attention) begins to drift while the prediction still holds, and only later does the prediction fail.

3.2 Synthetic data streams

A stream of graph-classification instances is generated over T discrete time steps. The number of nodes n is fixed; no nodes are added or removed. Each step yields a batch of graphs with binary labels. An early reference window $[0, R)$ contains no drift; drift then ramps linearly over $[t_{\text{onset}}, t_{\text{end}}]$.

The primary regime is *redundant structure*. The positive class carries two dense structural motifs, A and B, on disjoint sets of nodes; the negative class carries neither. Node features are uninformative noise, so the only predictive signal is structural and hence visible to an edge-based explainer. Drift degrades motif A *early* and motif B *late*. The intuition is a two-engine system: when A fails, the explanation shifts onto B, yet accuracy holds because B still separates the classes; only when B also fails does accuracy fall. This builds the precedence condition by construction, with a known onset.

We use a *no-drift* control (the same generator with drift switched off) to measure false alarms. Likewise, vary the construction (a single, gradually degrading motif and a feature-buffered variant) to check that any effect is not an artefact of a single design.

3.3 Model and explainer

A graph convolutional network is trained for graph classification on the reference window. Next, a parameterised edge explainer (PGExplainer-style; Luo et al., 2020) is trained on the same window, with the classifier frozen. The explainer is *inductive*: it is trained once and applied at every subsequent step so that we can cheaply track its edge masks over time, without retraining.

3.4 Signals over time

Each step t emits two signals from a held-out evaluation batch. The first one is Accuracy as well as the error, $e(t) = 1 - \text{acc}(t)$.

Next is the attribution drift, averaging the explainer's edge masks into an aggregate importance vector $\mathbf{a}_t$ over the $n(n-1)/2$ candidate edges. A reference vector $\mathbf{a}_{\text{ref}}$ is the mean over a held-out slice of the reference window. The drift at step t is a distance between $\mathbf{a}_{\text{ref}}$ and $\mathbf{a}_t$.

Most edges carry no signal, so a distance over the full vector is swamped by noise. We use a *focused* drift measure as the primary signal, chosen *a priori* on signal-to-noise grounds, not after seeing results. Let $\mathcal{K}$ be the top-k edges by reference importance, the edges the explainer actually relied on. We can then measure

Algorithm 1 Label-free monitoring of graph explanation drift

Require: graph stream $\{\text{batch}_t\}_{t\geq 0}$; reference window $[0, R)$; focus size k; smoothing window w; detector tolerance δ and threshold λ

train the classifier f on $[0, R)$; freeze its weights
train the inductive edge explainer g on $[0, R)$ to mimic f's *predicted* labels; freeze
$\mathbf{a}_{\text{ref}} \leftarrow$ mean of g's upper-triangle edge-mask vectors over $[0, R_0)$, with $R_0 = \lfloor R/2 \rfloor$
$\mathcal{K} \leftarrow$ indices of the top-k entries of $\mathbf{a}_{\text{ref}}$
calibrate μ (median) and σ ($1.4826 \cdot \text{MAD}$) of the w-smoothed D_{focus} series on the held-out slice $[R_0, R)$; initialise the Page–Hinkley accumulator m at R_0
for each step $t \geq R_0$ **do**
  $\mathbf{a}_t \leftarrow$ mean edge mask of g on batch_t {one forward pass; no labels}
  $D_{\text{focus}}(t) \leftarrow \big(\sum_{i\in\mathcal{K}} a_{\text{ref},i} - \sum_{i\in\mathcal{K}} a_{t,i}\big) \big/ \sum_{i\in\mathcal{K}} a_{\text{ref},i}$
  $z(t) \leftarrow \big(\text{smooth}_w(D_{\text{focus}})(t) - \mu\big)/\sigma$
  $m_t \leftarrow m_{t-1} + \big(z(t) - \bar{z}_t - \delta\big)$ {$\bar{z}_t$: running mean of z}
  if $t \geq R$ and $m_t - \min_{s\leq t} m_s > \lambda$ **then**
    alarm: $t_{\text{attr}} \leftarrow t$; flag the model for revalidation and **stop**
  end if
end for

the relative loss of importance mass on those edges (Eq. 1),

$$D_{\text{focus}}(t) \;=\; \frac{\sum_{i\in\mathcal{K}} a_{\text{ref},i} - \sum_{i\in\mathcal{K}} a_{t,i}}{\sum_{i\in\mathcal{K}} a_{\text{ref},i}} \tag{1}$$

Full-vector measures (rank correlation, cosine distance) are recorded for transparency.

3.5 Change-point detection and lead-time

Each series is converted to a single onset time using one detector, applied identically to both.

Applied to D_{focus} this gives the attribution onset t_{attr}; applied to $e(t)$ it gives the accuracy onset t_{acc}. The lead-time is (eq. 2)

$$L \;=\; t_{\text{acc}} - t_{\text{attr}}, \tag{2}$$

where $L > 0$ means the explanation drifted first. The threshold λ is chosen as the smallest value that yields a zero false-alarm rate on the no-drift controls, and the lead-time is reported only at that operating point.

The complete monitoring loop is summarised in Algorithm 1. It runs label-free: the only labelled data it ever touches is the reference window on which the classifier and explainer are trained. For evaluation, the identical detector is applied offline to the error series $e(t)$ to obtain t_{acc}, from which the lead-time L of Eq. 2 is computed.

4 Experiments

- **Precedence (main).** The primary regime over multiple seeds, with no-drift controls. It asks whether $L > 0$ and whether the detector stays silent without drift.
- **Mechanism (dose-response).** Sweep the degradation *staging* from synchronised (both motifs degrade together) to fully staged (A then B). The goal is for L to grow with staging and fall to zero under synchronisation. This falsifiable boundary ties the lead-time to the proposed mechanism rather than to the onset alone.
- **Robustness.** Vary the focus size k, and sweep the detector threshold λ to trace lead-time against false-alarm rate.

Table 1: Main result (redundant-structure regime, 10 seeds). Lead-time $L = t_{\text{acc}} - t_{\text{attr}}$; positive means explanation drift came first. The false-alarm rate is measured on 10 no-drift controls at the chosen operating point (λ giving zero control alarms).

Median L	IQR	Frac. $L > 0$	Wilcoxon p	Detection rate	False-alarm rate
9	[3, 12]	0.80	< 0.01	10/10	0/10

- **Generality.** We repeat the main test across the alternative regimes to check that the effect is not tied to a particular construction.
- **Baselines (for context).** Finally, run input-feature, embedding and degree-distribution maximum mean discrepancy, and predictive entropy, through the *same* calibrated detector, and compare their lead-times to attribution drift.

4.1 Metrics and goals

Each experiment reports seeds, the median lead-time and its interquartile range, the fraction of seeds with $L > 0$, and a Wilcoxon signed-rank test of $L > 0$. In addition, the detection rate (how often a true drift is caught) and the false-alarm rate (how often the detector fires without drift) are reported. The goals are: (i) to show a positive, significant lead-time at a zero-false-alarm operating point; (ii) to show, via the dose–response sweep, that the lead-time follows the staged-redundancy mechanism and vanishes when it should; and (iii) to show the result is stable across k and λ, and consistent across regimes.

5 Results

5.1 Explanation drift precedes accuracy degradation

Across the ten drift seeds, attribution drift was detected in every run, and in most it was detected before accuracy fell. The median lead-time was 9 steps (IQR $[3, 12]$); the lead-time was positive in 8/10 runs, and a Wilcoxon signed-rank test rejected $L \leq 0$ at $p < 0.01$. On the ten no-drift controls, the detector never fired (false-alarm rate 0/10). Thus, the signal is both timely and silent when nothing is wrong. These figures are summarised in Table 1, and a representative run is shown in Figure 2, where the focused attribution-drift curve rises shortly after the drift onset while accuracy holds, and only later collapses.

5.2 The lead-time follows the staged-redundancy mechanism

To test whether the lead-time reflects the proposed mechanism rather than the drift onset alone, the degradation *staging* was swept from synchronised to fully staged (Figure 3). When both motifs degrade together (synchronised), no precedence was observed: the median lead-time was negative and not significant. As staging increased, the lead-time rose monotonically and became significant toward the fully-staged end. This dose-response behaviour, including the vanishing lead under synchronisation, indicates that the precedence is driven by staged degradation of redundant structure, not by the detector simply firing at onset.

5.3 Robustness

Two free parameters govern the measurement — the focus size k and the detector threshold λ — and the result is stable across both. Varying k over $\{5, 10, 20, 30, 50, 80, 120\}$ (of 300), the median lead-time was stable between 7.5 and 9 steps and significant for $k \geq 20$ (Wilcoxon $p < 0.05$, false-alarm rate 0); at the two narrowest settings ($k = 5, 10$) the focused metric became too sparse to reach significance. See Figure 4. Sweeping λ traced a clear operating region: the control false-alarm rate falls to 0 for $\lambda \geq 8$, and across that zero-false-alarm plateau the median lead-time stays positive throughout, rising from 6.5 steps at $\lambda = 8$ to 11.5 at $\lambda = 30$. The reported operating point ($\lambda = 16$, median lead 9) sits inside this plateau, where the detector is silent on every no-drift control (Figure 5).

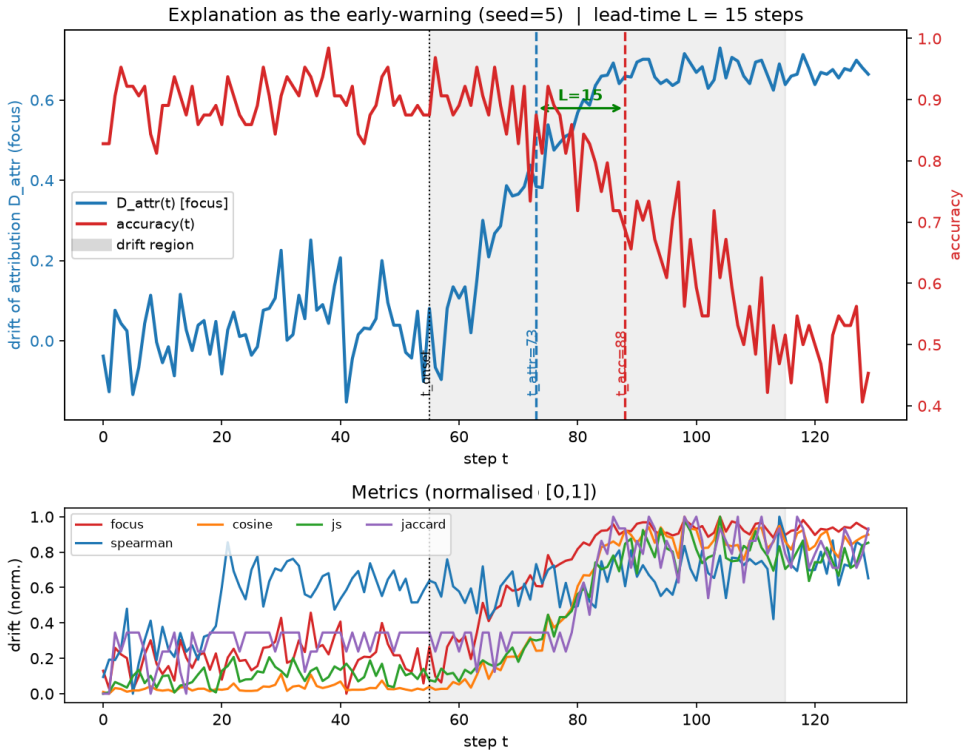

Figure 2: A single run (seed 5; its lead-time of 15 steps lies above the median of 9 but within the observed range, and is shown for legibility). The focused attribution-drift signal (top, blue) departs from its reference band well before accuracy (top, red) declines; the dashed lines mark the detected onsets t_{attr} and t_{acc}. The lower panel shows the soft detection of each metric for the same run.

Table 2: Lead-time by signal (same detector, 10 seeds). Reported for context; structural discrepancy leads at least as early as attribution drift in this regime. Input-feature MMD is a pure-noise signal here (the features are non-informative), so it never crosses the detector and its lead-time is undefined (—).

Signal	Median L	IQR	Frac. $L > 0$	Wilcoxon p	FA (no-drift)
Attribution drift (ours)	9.0	[3, 12]	0.80	< 0.01	0/10
Degree-distribution MMD	11.0	[4, 18]	0.90	< 0.01	0/10
Embedding MMD	4.5	[−6, 20]	0.60	0.19	2/10
Input-feature MMD	—	—	—	—	0/10
Predictive uncertainty	−2.0	[−4, 3]	0.33	0.77	0/10

5.4 Comparison with drift-detection baselines

For context, four label-free baselines were passed through the identical detector and compared on lead-time (Table 2). Attribution drift was a strong leading indicator, but not the earliest: a structural baseline (maximum mean discrepancy on the degree distribution) led by a larger margin (11 vs. 9 steps), as expected in a regime where the drift is itself structural. Input-feature discrepancy and predictive uncertainty did not lead at all. Explanation drift can be presented as a *valid* early-warning signal rather than the fastest one. Its distinct value is qualitative — it identifies *which* edges the model has stopped relying on, made concrete in Section 5.5 — and it becomes possible to report the baselines in full rather than select among them.

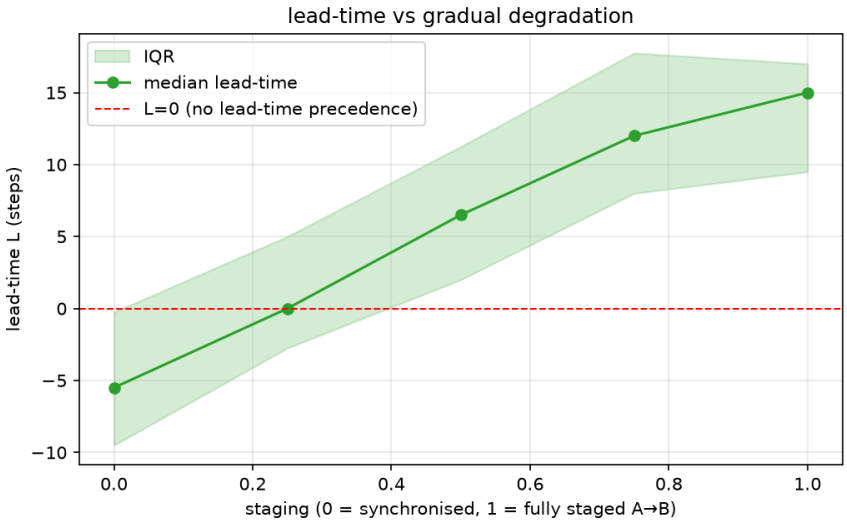

Figure 3: Dose-response, aggregated over seeds (not a single run; 10 seeds per staging value, passed through the same calibrated detector and operating point as Table 1). The line is the median lead-time L and the band its IQR, against the degradation staging parameter (0 = synchronised, both motifs degrade together; 1 = fully staged, A degrades before B; see Appendix A for the exact breakpoint schedule). The dashed line marks $L = 0$ (no precedence). The lead-time increases monotonically with staging and vanishes under synchronisation, as the staged-redundancy mechanism predicts.

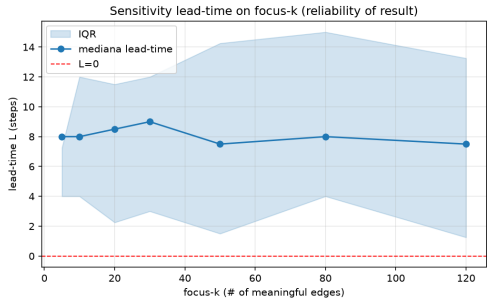

Figure 4: Lead-time is stable across the focus size k.

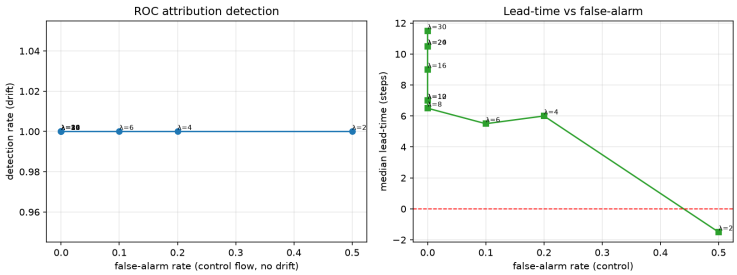

Figure 5: Lead-time and false-alarm rate against the detector threshold λ.

5.5 Where the explanation moves: the attribution hand-off

The lead-time establishes *that* the explanation drifts first, but not *where* it goes. Because the synthetic generator assigns each edge to motif A, motif B, or the uninformative background, the explainer's edge masks can be projected onto these ground-truth groups and the reallocation of importance can be read directly. Figure 6 tracks, over time, the total attribution mass the explainer places on A-edges versus B-edges, alongside accuracy. The two curves are stable and well separated through the reference window; shortly after the onset the mass on A collapses while the mass on B rises to replace it, and this hand-off completes *before* accuracy departs from its plateau. The per-edge strip (bottom panel) makes the same reallocation legible at the level of individual edges: the A-block dims out and the B-block lights up, with the crossover sitting to the left of t_{acc}.

This is the concrete evidence behind the qualitative claim of Section 5.4: the signal does not merely cross a threshold, it reports *which* structure the model has stopped relying on. Crucially, the projection uses

the trained explainer's masks, not the generator's motif labels, so the hand-off is a property of the model's explanation and not a restatement of the induced drift; the ground-truth grouping is used only to read the masks, never to produce them.

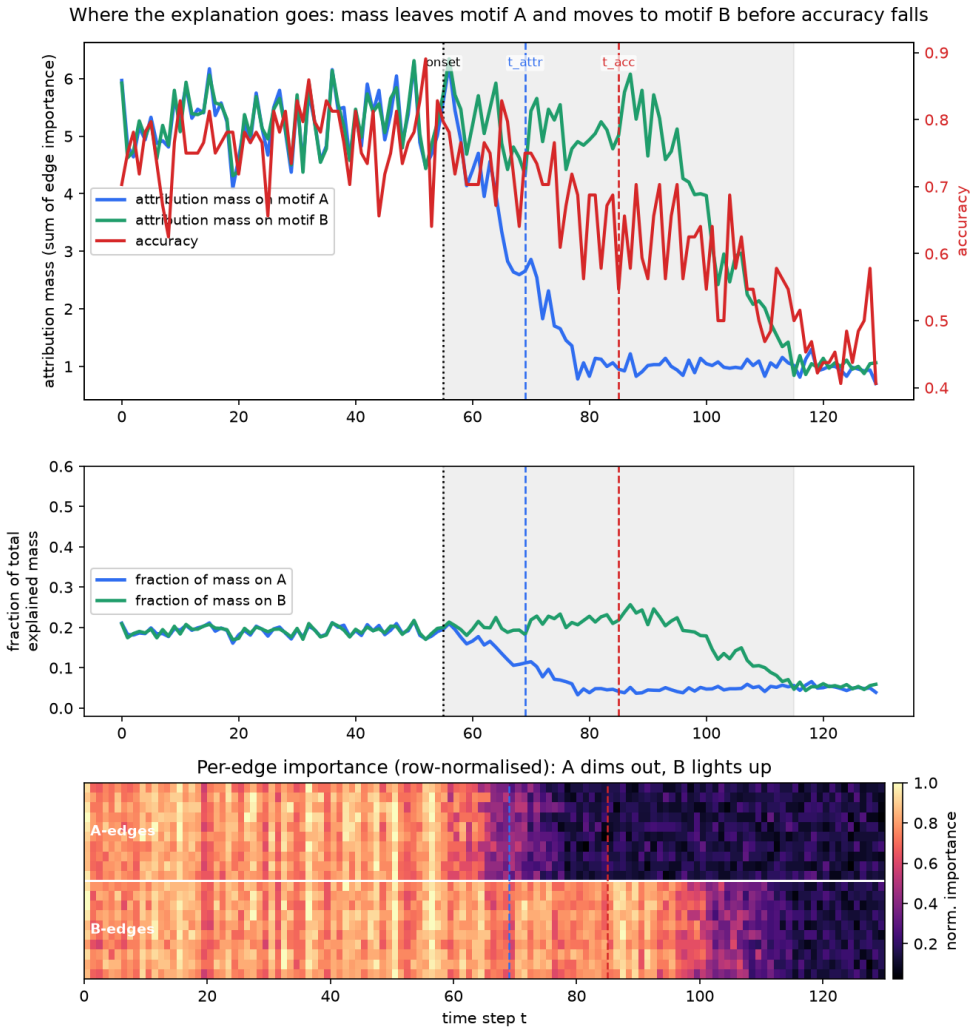

Figure 6: The attribution hand-off (a single representative synthetic run; the aggregate statistics are in Table 1). **Top:** total attribution mass on motif-A edges (blue) and motif-B edges (green), with accuracy (red, right axis). Mass leaves A and moves to B before accuracy falls; dashed lines mark t_{attr} and t_{acc}. **Middle:** the same hand-off as a fraction of total explained mass (scale-free). **Bottom:** per-edge importance (row-normalised), A-edges above the divider and B-edges below — A dims out as B lights up. The mask projection uses the trained explainer, with ground-truth motif membership used only to read the masks.

6 Reaffirmation on IT infrastructure (RCAEval)

IT infrastructure changes and evolves constantly, which makes it a natural setting for this explanation-drift study (Szandała, 2025a). The dependency graph *is* the system: services are nodes and calls are edges, so faults propagate along the very structure an edge explainer inspects. Resilient deployments also include redundancy, replicas, retries, and failover, which absorb degradation until a component saturates. This is exactly the staged-redundancy mechanism studied on synthetic data: when one path fails, the model's reliance shifts (the explanation drifts), while end-to-end accuracy holds on the backup, and only

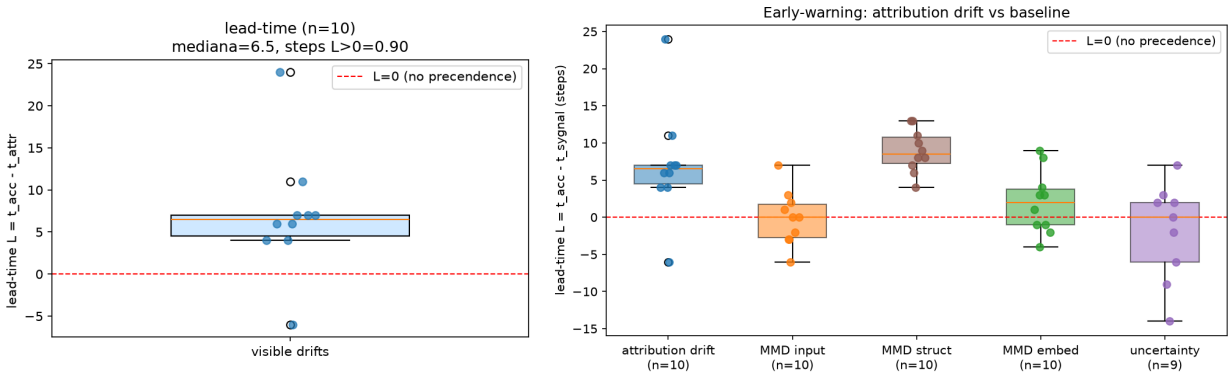

Figure 7: Explanation drift performance on IT infrastructure (RCAEval), 10 drift runs. **Left:** distribution of the attribution-drift lead-time $L = t_{\text{acc}} - t_{\text{attr}}$; the median is 6.5 steps and 90% of runs are positive, so the explanation drifts before diagnostic accuracy falls. **Right:** lead-time by signal, all passed through the same detector. Attribution drift leads clearly; a degree-distribution (structural) baseline leads by a slightly larger margin, while feature, embedding and uncertainty signals do not lead. The lead is smaller and noisier than on synthetic data, as expected for a densely coupled service graph.

later collapses. Since infrastructure also develops continuously, a trained root-cause-analysis (RCA) model drifts out of date; asking whether its explanation moves before its diagnostic accuracy falls is therefore operationally useful, as an early signal that the model should be revalidated before it silently misdiagnoses.

This setting was instantiated using a service call graph from RCAEval (Pham et al., 2025), to which controlled, cumulative topology evolution was applied: a semi-synthetic design combining a realistic topology with a fully known onset. The diagnosis task was kept binary (the region of the root cause), so the pipeline of Section 3 was mostly reused. Precedence was again observed (Figure 7, left): the median lead-time was 6.5 steps, positive in 9/10 runs ($p \approx 0.007$), with no false alarms in the static control. The effect was somewhat smaller than on synthetic data (median 9, positive in 8/10). The reason is structural: real service graphs are densely coupled, so the predictive substructure is not isolated, and message passing mixes unrelated changes into the explanation, leaving a noisier signal. Note the wider spread and the occasional negative run in Figure 7 (left). Among the baselines (Figure 7, right), attribution drift was again a clear leading indicator, while a structural discrepancy baseline led by a slightly larger margin and the input, embedding and uncertainty signals sat near zero. Explanation drift thus carries over to IT infrastructure, with a smaller and less clean margin than in the controlled setting.

The attribution drift also transfers to this substrate. Projecting the explainer's masks onto the diagnostic regions A and B (Figure 8), the importance mass leaves the A-region and moves onto B as the topology evolves, and this reallocation again precedes the accuracy drop, here by 11 steps, consistent with the smaller but positive median lead. The hand-off is noisier than in the controlled setting: dense coupling lets message passing leak importance onto background edges, so the A/B curves separate less cleanly. Nonetheless the qualitative signal survives on a realistic service graph and the explanation localises *which* region the RCA model has stopped relying on, before its diagnoses fail.

7 Discussion

A single, falsifiable claim was evaluated: that on graphs, explanation drift precedes accuracy degradation by a measurable lead-time. The evidence supports it. The lead-time is positive, significant, and silent on no-drift controls; it follows the staged-redundancy mechanism through a dose-response sweep, vanishing exactly where the mechanism is absent. This pattern is also observed on a realistic IT-infrastructure substrate. The contribution is therefore narrow but robust; there are several limitations.

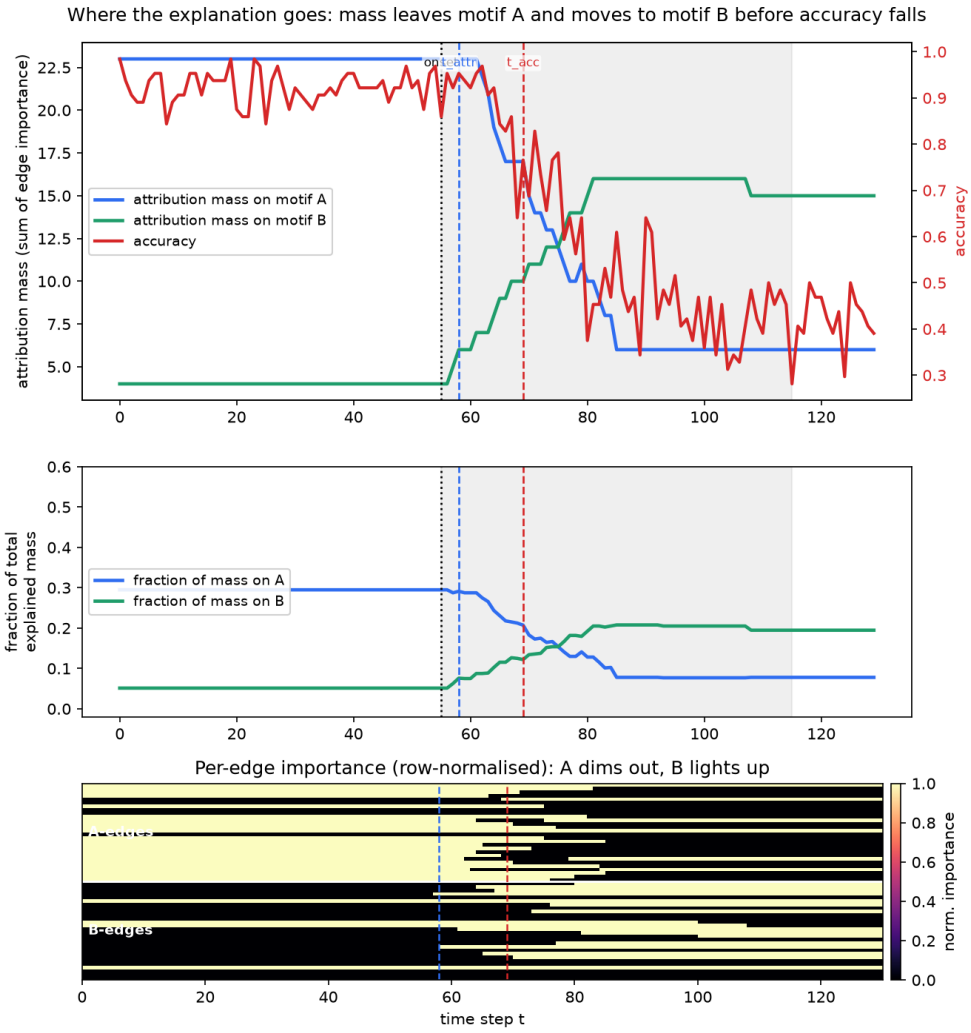

Figure 8: The attribution hand-off on IT infrastructure (RCAEval, a single representative run; the aggregate statistics are in Figure 7). Total attribution mass on region-A edges (blue) versus region-B edges (green), with accuracy (red, right axis). Mass moves from A to B about 11 steps before accuracy falls; the middle panel shows the scale-free fraction and the bottom panel the per-edge reallocation. The separation is noisier than on synthetic data, as expected for a densely coupled service graph.

The primary result rests on synthetic streams in which the drift onset and the redundant structure are built in by design. This is deliberate, as it provides ground-truth timing but limits external validity. The IT-infrastructure study is semi-synthetic: controlled, cumulative topology evolution applied to a real service graph, not a naturally occurring production incident with an organic onset.

The attribution drift leads, but a cheap structural baseline (degree-distribution MMD) leads by a larger margin in both settings. On lead-time alone, the explanation signal does not dominate. Its distinctive value is qualitative; it localises *which* edges or services the model has stopped relying on. This is not merely asserted: the attribution hand-off from motif/region A onto B is shown directly on both substrates (Figures 6 and 8), and in both the reallocation precedes the accuracy drop. A purely structural alarm reports *that* the graph changed; the explanation additionally reports *where* the model's reliance moved, which is the actionable quantity for revalidation.

The signal can only see the drift that flows through the substructure the explainer inspects. Drift arriving through a channel the model does not explain through, for instance, a shift in label priors rather than in structure, would move accuracy without first moving the explanation, and would be missed.

On densely coupled graphs, global message passing mixes unrelated changes into the explanation, shrinking the lead and adding noise, as seen in the smaller RCAEval margin. The method is at its clearest when the predictive substructure is relatively isolated and free of noise.

Operational cost. The monitoring overhead is small by construction. The explainer is inductive: it is trained once, on the reference window, and then frozen, so producing the drift signal at each step costs one forward pass of a small MLP over the candidate edges plus an $O(1)$ incremental update of the Page-Hinkley statistic. No labels, no retraining, and no gradient computation are required during monitoring; the only labelled data ever used is the reference window on which the classifier and explainer are trained.

Choosing λ in deployment. In the experiments, λ is set to the smallest value with zero false alarms on held-out no-drift controls - a luxury of the controlled setting. In deployment, the calibration slice of the reference window plays that role: λ can be chosen as the smallest value that produces no alarms on the reference stream itself, or on bootstrap resamples of it for a less optimistic estimate, reproducing the zero-false-alarm plateau logic without an external clean stream. The robust statistics (μ, σ) and λ should be re-calibrated whenever the model is revalidated or retrained, since each revalidation yields a fresh, verified reference window. Under known seasonality, the reference window should span at least one full seasonal cycle, or one calibration per regime can be maintained. The residual assumption, that the initial reference window is genuinely drift-free, is shared by every reference-based drift detector and is not specific to attribution drift.

Architecture and explainer dependence. The main results use a graph convolutional network with a PGExplainer-style edge explainer, chosen for their established position and stability. We additionally attempted the study on two further encoders, a graph attention network (GAT) and GraphSAGE, and the outcome is itself informative about the method's preconditions. On GraphSAGE the classifier did not settle to a stable reference accuracy (it oscillated by roughly two percentage points across training epochs); because the change-point detector is calibrated on the reference window, this instability produced an accuracy onset *before* the injected drift ($t_{\text{acc}} < t_{\text{onset}}$), which makes the lead-time ill-defined for that encoder. On GAT the classifier reached only about 0.77 reference accuracy and the parameterised explainer's mask degenerated, so no reliable attribution signal was available. In both cases the obstacle is a training pathology of the classifier or explainer on these small dense graphs, not evidence about precedence. Nonetheless, this dependence is a central limitation of the method rather than an implementation footnote: the lead-time signal is only meaningful when a precondition holds that our GCN setup satisfies but these encoders did not — a reference model that converges to a stable accuracy and an explainer that yields a selective mask. Crucially, both parts of the precondition can be verified *before* any monitoring, using only the reference window: (i) *reference stability*, the classifier's accuracy on the held-out calibration slice should be stationary across training epochs (GraphSAGE's oscillation of roughly two percentage points fails this check); and (ii) *mask selectivity*, the share of total mask mass concentrated on the top-k reference edges should clearly exceed the uniform share k/E, where E is the number of candidate edges (GAT's degenerate, near-uniform mask fails it). A practitioner can therefore diagnose whether to trust the lead-time signal at calibration time, before deployment. Characterising which encoder-explainer pairings meet this precondition, and stabilising those that do not, is a concrete direction rather than a mere caveat. A noisier or more robust explainer could likewise blur or sharpen the lead.

7.1 Future work

These limitations point to several directions for future work. Building on the attempt reported above, a natural next step is to extend the study across GNN architectures and explanation methods once their training is stabilised, to determine whether the lead-time is explainer-dependent and whether more stable explainers yield cleaner leads.

IT infrastructure is a natural fit for this finding, yet it is worth seeking a non-synthetic positive on data with organic structural concept drift and known degradation.

A further step is to extend from binary region diagnosis to root-cause *localisation*, and to graphs that grow and shrink as services are added and removed.

It should also be measured whether the drifting subset of edges or services actually points operators to what changed, validated against logged infrastructure modifications.

It may be worth pairing a structural detector (for *when*) with attribution drift (for *where*, and whether it matters), and studying locality-restricted attributions that resist the coupling effect above. And finally, relate existing explanation-stability bounds to lead-time: under what margin and redundancy conditions must the explanation drift first?

8 Conclusion

Under the evaluated structural degradation conditions, the graph model's explanation drifts before its accuracy falls. On synthetic graph streams, the explanation drift preceded the accuracy degradation by a clear, significant lead time, which vanished exactly when the staged-redundancy mechanism was removed. The effect carries over, more weakly, to a realistic IT infrastructure graph.

Explanation drift is not the earliest possible signal; a simple structural baseline fires at least as early. But it is label-free and points to *which* part of the model has gone stale. It is a practical early warning that a graph model, such as one used for root-cause analysis, should be revalidated before it silently fails.

References

Chirag Agarwal, Owen Queen, Himabindu Lakkaraju, and Marinka Zitnik. Evaluating explainability for graph neural networks. *Scientific Data*, 10(1):144, 2023.

Doha Al Bared and Mohamed Nassar. Segmentation fault: A cheap defense against adversarial machine learning. In *2021 3rd IEEE Middle East and North Africa COMMunications Conference (MENACOMM)*, pp. 37–42. IEEE, 2021.

Albert Bifet and Ricard Gavaldà. Learning from time-changing data with adaptive windowing. In *Proceedings of the 2007 SIAM International Conference on Data Mining*, pp. 443–448. SIAM, 2007.

Vitor Cerqueira, Heitor Murilo Gomes, Albert Bifet, and Luis Torgo. STUDD: A student–teacher method for unsupervised concept drift detection. *Machine Learning*, 112(11):4351–4378, 2023.

Jaka Demšar and Zoran Bosnić. Detecting concept drift in data streams using model explanation. *Expert Systems with Applications*, 92:546–559, 2018.

Christopher Duckworth, Francis P Chmiel, Dan K Burns, Zlatko D Zlatev, Neil M White, Thomas WV Daniels, Michael Kiuber, and Michael J Boniface. Using explainable machine learning to characterise data drift and detect emergent health risks for emergency department admissions during covid-19. *Scientific reports*, 11(1):23017, 2021.

João Gama, Pedro Medas, Gladys Castillo, and Pedro Rodrigues. Learning with drift detection. In *Advances in Artificial Intelligence – SBIA 2004*, pp. 286–295. Springer, 2004.

Johannes Haug, Alexander Braun, Stefan Zürn, and Gjergji Kasneci. Change detection for local explainability in evolving data streams. In *Proceedings of the 31st ACM international conference on information & knowledge management*, pp. 706–716, 2022.

Fabian Hinder, Valerie Vaquet, Johannes Brinkrolf, and Barbara Hammer. Model-based explanations of concept drift. *Neurocomputing*, 555:126640, 2023.

Jason LOPATECKI, Aparna Dhinakaran, and Michael Schiff. Data drift impact in a machine learning model, August 6 2024. US Patent 12,056,586.

Ana Lucic, Maartje A Ter Hoeve, Gabriele Tolomei, Maarten De Rijke, and Fabrizio Silvestri. CF-GNNExplainer: Counterfactual explanations for graph neural networks. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 4499–4511. PMLR, 2022.

Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. Parameterized explainer for graph neural network. *Advances in neural information processing systems*, 33: 19620–19631, 2020.

Carlos Mougan, Klaus Broelemann, Gjergji Kasneci, Thanassis Tiropanis, and Steffen Staab. Explanation shift: Detecting distribution shifts on tabular data via the explanation space. *arXiv preprint arXiv:2210.12369*, 2022.

Carlos Mougan, Klaus Broelemann, Gjergji Kasneci, Thanassis Tiropanis, and Steffen Staab. Explanation shift: how did the distribution shift impact the model? *TMLR: Transactions on Machine Learning Research*, 2025.

Ewan S Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954.

Luan Pham, Hongyu Zhang, Huong Ha, Flora Salim, and Xiuzhen Zhang. Rcaeval: a benchmark for root cause analysis of microservice systems with telemetry data. In *Companion Proceedings of the ACM on Web Conference 2025*, pp. 777–780, 2025.

Aalok Shanbhag, Avijit Ghosh, and Josh Rubin. Unified shapley framework to explain prediction drift. *arXiv preprint arXiv:2102.07862*, 2021.

Tomasz Szandała. Aiops for reliability: Evaluating large language models for automated root cause analysis in chaos engineering. In *International Conference on Computational Science*, pp. 323–336. Springer, 2025a.

Tomasz Szandała. Chatgpt vs human expertise in the context of it recruitment. *Expert Systems with Applications*, 264:125868, 2025b.

Mengjiao Yang and Been Kim. Benchmarking attribution methods with relative feature importance. *arXiv preprint arXiv:1907.09701*, 2019.

A Reproducibility details

This appendix specifies the model, explainer, detector, and data generators in enough detail to reproduce every reported number. All randomness is controlled by a single per-run seed; the drift runs use seeds 0–9, the no-drift controls 100–109, and the covariate-only controls 200–229. All results were regenerated under a single pinned software environment (`torch` 2.12, `numpy` 2.4, `scipy` 1.17); absolute lead-times shift slightly across library versions through changes in weight initialisation and reduction order, while the qualitative findings (positive median lead, structural baseline at least as early, weak signals not leading) are stable.

A.1 Model and training

The classifier is a graph convolutional network operating on dense adjacency batches. It uses an input feature dimension F (dataset-dependent), a hidden dimension of 32, and a graph-embedding dimension of 16, followed by a linear head to two classes. It is trained *only* on the reference window $[0, R)$ and then frozen for all later time steps; no weights are updated during monitoring. Training uses Adam (learning rate 10^{-2}, weight decay 5×10^{-4}), a batch size of 128, cross-entropy loss, and 120 epochs. The reference window is $R = 40$ steps in every experiment.

A.2 Explainer

The edge explainer is a PGExplainer-style parameterised model (Luo et al., 2020): a small MLP (hidden dimension 32) that maps a pair of node embeddings from the frozen classifier to an edge-mask logit. It is trained once, on the same reference window, with the classifier frozen, for 60 epochs using Adam (learning rate 10^{-2}) and a batch size of 128. The explainer is trained to reproduce the classifier's *predicted* label (not the ground-truth label), using a concrete relaxation with temperature $\tau = 1.0$ and mask regularisers weighting size (0.05) and entropy (0.1). Because the explainer is inductive, the same trained model is applied at every subsequent step; edge masks are tracked over time without any retraining.

A.3 Signals and drift metrics

At each step t the explainer's edge masks are averaged over the evaluation batch ($B = 64$ graphs) and read off the upper triangle into an importance vector $\mathbf{a}_t \in \mathbb{R}^{n(n-1)/2}$. The reference vector $\mathbf{a}_{\text{ref}}$ is the mean of $\mathbf{a}_t$ over the held-out slice $[0, R_0)$ with $R_0 = \lfloor R/2 \rfloor$; the complementary slice $[R_0, R)$ is reserved for detector calibration, so the reference vector and the calibration statistics never share data. The primary signal is the focused mass-drop D_{focus} of Eq. 1 on the top-k reference edges ($k = 30$ unless swept). For transparency, four full-vector metrics are also recorded: 1−Spearman, cosine distance, Jensen–Shannon divergence, and Jaccard@k ($k = 20$) on the top-k sets.

A.4 Change-point detector

Onsets are produced by a calibrated Page–Hinkley test applied identically to the drift signal and to the error series $e(t) = 1 - \text{acc}(t)$. Each series is first smoothed with a causal moving average (window 3), then standardised using robust statistics from the calibration slice $[R_0, R)$: $z(t) = (x(t) - \mu)/\sigma$, where μ is the median and $\sigma = 1.4826 \cdot \text{MAD}$. The Page–Hinkley accumulator is $m_t = \sum_{s \le t} \big(z(s) - \bar{z}_s - \delta\big)$ with running mean $\bar{z}_s$, and an alarm fires when $m_t - \min_{s \le t} m_s > \lambda$. Both thresholds are in calibrated (standard-deviation) units: tolerance $\delta = 1.5$ and alarm threshold $\lambda = 16$. The accumulator starts at R_0 so the pre-reference region cannot poison the statistic, and alarms are only accepted from $t \ge R$. The first accepted alarm is the onset; if none occurs the onset is undefined and the lead-time is treated as missing. The operating point λ is chosen as the smallest value giving a zero false-alarm rate on the no-drift controls (Figure 5); the reported λ sits inside the resulting plateau.

A.5 Synthetic generator (redundant-structure regime)

Each step $t \in \{0, \ldots, T-1\}$, with $T = 130$, emits a batch of $B = 64$ graphs on a fixed set of $n = 25$ nodes, giving $n(n-1)/2 = 300$ candidate edges per graph; the same batch serves as the held-out evaluation batch

for that step. Node features are $F = 4$-dimensional i.i.d. Gaussian noise, so the only discriminative signal is structural. The positive class plants two dense motifs A and B, each on $K = 5$ nodes, on disjoint node sets ($\binom{5}{2} = 10$ intra-motif node pairs per motif); in the undrifted state an intra-motif edge appears independently with probability $p_{\text{hi}} = 0.95$, while every other node pair — the background, and all pairs in negative-class graphs — appears independently with probability $p_{\text{base}} = 0.06$.

The reference window $[0, R)$ with $R = 40$ is drift-free. The global drift progress then ramps linearly, $\rho(t) = \min\big(1, \max\big(0, (t - t_{\text{onset}})/(t_{\text{end}} - t_{\text{onset}})\big)\big)$ with $[t_{\text{onset}}, t_{\text{end}}] = [55, 115]$. Degradation lowers a motif's intra-motif edge probability linearly from $p_{\text{hi}} = 0.95$ to $p_{\text{lo}} = 0.10$ as that motif's own degradation progress runs from 0 to 1. Staging maps the global progress ρ to per-motif progress through two breakpoints: motif A degrades over $\rho \in [0, \rho_A^{\text{end}}]$ and motif B over $\rho \in [\rho_B^{\text{start}}, 1]$. The fully staged setting uses $(\rho_A^{\text{end}}, \rho_B^{\text{start}}) = (0.4, 0.6)$, so A is fully degraded before B begins to degrade; the synchronised setting uses $(1, 0)$, i.e. both motifs degrade together over the entire ramp. The dose–response sweep (Figure 3) interpolates the breakpoints linearly between these two endpoints.

Controls and variants use the same generator. The no-drift control switches the ramp off entirely. A covariate-only control raises the background density from $p_{\text{base}} = 0.06$ to 0.10 along the ramp while leaving both motifs intact — input drift that should not move accuracy, used to measure false alarms under benign covariate shift. A single-motif variant plants one motif and degrades it gradually over the full ramp. A feature-buffered variant degrades the structure over the first half of the ramp while an informative class-conditional feature signal (mean shift $\mu_{\text{feat}} = 1.6$) ramps in from the second half, so accuracy is buffered by features rather than by a second motif.

A.6 RCA generator (infrastructure evolution)

The IT-infrastructure substrate treats a service dependency graph as the input: services are nodes and calls are edges. The diagnostic task is binary — whether the injected root cause lies in region A or region B — so the whole pipeline is reused unchanged. A failure signature is injected by raising the metric values of the root cause and its neighbourhood, decaying with hop distance along the graph. Infrastructure evolution is applied as a cumulative, monotonic rewiring trajectory: in the `evolve` mode the diagnostically relevant A/B dependencies are rewired (staged, as in the synthetic sweep), so the model goes stale; in the `churn` control only irrelevant decoy dependencies change, so accuracy should hold and the signal should stay silent; the `none` mode applies no evolution and serves as the false-alarm control. The topology can be synthetic (runnable out of the box) or loaded from an RCAEval service graph (Pham et al., 2025); the evolution and labelling logic are identical in both cases, and the onset is fully known, making the design semi-synthetic.

A.7 The attribution hand-off (Figures 6 and 8)

The hand-off figures project the trained explainer's edge masks onto ground-truth groups. Node membership of motif/region A and B is recorded by the generator; an edge is assigned to a group iff *both* endpoints belong to that group, using the same upper-triangle ordering as $\mathbf{a}_t$. For each step the attribution mass on a group is the sum of $\mathbf{a}_t$ over that group's edges; the fractional panel divides by the total explained mass, and the per-edge strip row-normalises each edge to its own maximum so the reallocation is visible per edge regardless of scale. The ground-truth grouping is used only to *read* the masks; the masks themselves come from the trained explainer, so the hand-off is a property of the model's explanation rather than a restatement of the induced drift.

A.8 Notation

Throughout, $L = t_{\text{acc}} - t_{\text{attr}}$ is the lead-time in steps (positive means the explanation drifted first); IQR denotes the interquartile range $[Q_1, Q_3]$; a run's lead-time is missing when either onset is undefined, and such runs are excluded from medians and tests while still counting against the detection rate. Significance is a one-sided Wilcoxon signed-rank test of $L > 0$.