Journal of Mathematical Finance and Risk Management
ISSN: 3080-9983 (Print), ISSN: 3080-9991 (Online) | Vol. 2, No. 1, 2026

Budgeted Class-Balanced Drift Gating for Directional Temporal Graph Fraud Detection

Sabrina Chandra [1], Yuna Klein [2], Diego Kim [3, *]

[1] Department of Computer Science and Engineering, University of California, San Diego, CA, USA

[2] Department of Computer Science, University of Texas at Austin, TX, USA

[3] Paul G. Allen School of Computer Science & Engineering, University of Washington, WA, USA

[*] **Corresponding author**: Diego Kim (Email: diego.kim32@uw.edu)

Abstract: Concept drift can render a graph-based fraud classifier obsolete even when its topology encoder remains useful. This study investigates when a delayed-label alarm should update a classifier under independent class-prior fluctuations and abrupt changes in transaction-flow roles. Building on the seven-channel directed two-hop simplified graph convolution and stable–plastic classifier of ATGNN-SP, we introduce BCG-5: a class-balanced predictive-loss gate, a five-update stream budget, and recent-only plastic fitting. Controlled experiments use reproducibly generated directed transaction graphs, 49 chronological snapshots, and five paired seeds in three regimes; no production transaction data are used. In the mixed regime, BCG-5 achieves average precision 0.3175 ± 0.0366 and F1 0.3413 ± 0.0693, compared with 0.3004 ± 0.0280 and 0.3367 ± 0.0712 for the loss-gated ATGNN-SP update rule, while reducing mean updates from 8.0 to 5.0. A plastic-only alternative achieves higher predictive accuracy at greater update frequency, revealing a meaningful stability–adaptability trade-off.

Keywords: Financial fraud, temporal graph, concept drift, class-balanced risk, online learning, prequential evaluation.

1. Introduction

Fraud detection on transaction networks is affected by evolving behavior, rare positive labels, and time-varying investigative coverage. The Elliptic study demonstrated both the value of relational Bitcoin transaction information and the importance of temporal structure in evaluating anti-money-laundering models [1]. Drift-adaptation research further separates a change in class prevalence from a change in the relation between inputs and the target label [2]. These phenomena may look similar to a detector that simply monitors aggregate predictive loss, yet their implications for retraining are different.

Lightweight graph encoders are attractive when labels arrive in batches. Simplified graph convolution (SGC) amortizes topology propagation into a fixed representation, making classifier refresh less expensive [3]. Yue et al. [4] built on this property in ATGNN-SP: directed self, incoming, outgoing, two-hop and cross-direction SGC channels feed a frozen stable branch and a drift-updated plastic branch with anchor replay, so that the cached graph representation is reused while only a lightweight classifier adapts. Its delayed-label gate is based on ordinary snapshot log loss. We retain its directed propagation operator and stable–plastic scoring sequence and investigate a different decision rule for consuming an update budget.

Our research question is narrow: can class-balanced evidence and a bounded recent-window update policy improve the efficiency of adaptation when a shift in the frequency of illicit transactions is mixed with a genuine reversal of their graph roles? The question matters because an unbalanced loss average is sensitive to the current positive fraction even if class-conditional predictive quality has not changed. Conversely, replaying a well-learned historical concept can slow recognition when the new concept reverses the former fraud signature.

The contributions are: (i) a prior-balanced delayed predictive-risk statistic with a simple invariance argument; (ii) a budgeted recent-only plastic update mechanism built on the directed propagation operator and stable–plastic scorer of ATGNN-SP; (iii) a three-regime controlled study separating isolated prior perturbations, a conditional topology shift, and their combination; and (iv) label-delay and fixed-conditional-score diagnostics, together with per-seed predictions, alarm traces, ablations, and negative results. The study is intended as a method-level controlled evaluation, not an estimate of deployment accuracy on real payment traffic.

2. Related Work

2.1. Concept Drift, Monitoring, And Imbalance

Lu et al. categorize concept drift by whether it alters the input distribution, the label prior, or the decision relationship [5]. ADWIN detects distribution changes using adaptive windows [6], whereas Page's sequential inspection framework motivates accumulation of prediction residuals [7]. The operational choice of an alarm variable is equally consequential: under rare positive labels, precision–recall metrics are often more informative than ROC summaries [8], and conventional imbalance remedies can distort apparent confidence if their training and evaluation roles are conflated [9]. Our monitor is computed on natural-prevalence evaluation labels; oversampling is confined to fitting the MLP.

2.2. Graph Representation Across Time

Graph convolution [10], sampled inductive aggregation [11], and graph attention [12] offer distinct approaches to relational learning. EvolveGCN evolves convolution parameters across graph snapshots [13], while temporal graph networks [14], JODIE [15], and DySAT [16] model time through event memory, trajectory evolution, and joint structural–temporal attention. These approaches are designed

for richer temporal interactions than our coarse, disconnected synthetic snapshots. Fixed directed propagation permits an unusually transparent separation between information available at prediction and labels received afterward.

2.3. Graph-based Financial Risk

CARE-GNN targets relational camouflage [17], and PC-GNN addresses supervised graph imbalance with tailored sampling [18]. Systems including xFraud [19] and BRIGHT [20] develop explanation-oriented and latency-aware architectures at much larger operational scale. Elliptic++ expands Bitcoin forensic graph evidence [21], and label-scarcity analysis highlights the cost of acquiring reliable AML supervision [22]. A broad review of network analytics for AML identifies evaluation protocol and task heterogeneity as recurring barriers to direct comparisons [23]. Graph anomaly surveys [24, 25] also distinguish local structure from collective irregularities. Our controlled generator isolates direction-role reversal rather than claiming to reproduce strategic evasion in any of those real systems.

3. Problem and Method

3.1. Prequential Stream and Evidence Availability

Let $G_t=(V_t, E_t)$ denote a directed transaction graph at time t. A node represents one transaction; an arc $u\to v$ denotes a possible value-flow relation from u to v. Only a fixed fraction of node labels is observable for supervised evaluation. At each step, a model scores the observed topology before its labels are disclosed. The labels are then used solely to compute an alarm and update the plastic classifier for the next step. The initial training period is t=1, …, 29, calibration is t=30, …, 34, and the held-out stream is t=35, …, 49. No test labels enter scaling, threshold selection, or their own predictions.

3.2. Directed Two-Hop Propagation

For adjacency A_t with $A_t[u,v]=1$ for $u\to v$, define $P_i=D_i^{-1}A_t^T$ and $P_o=D_o^{-1}A_t$, treating zero-degree rows as zero. Nineteen structure-only node descriptors encode degree, direction imbalance, source/sink status, predecessor and successor degree summaries, neighbor-flow totals, approximate PageRank and HITS scores, undirected clustering, and core index. Let $X_t\in R^{nx19}$ collect these descriptors. We use the seven-block directed SGC operator of ATGNN-SP [4]:

$$Z_t = [X_t \,\|\, P_iX_t \,\|\, P_oX_t \,\|\, P_i^2X_t \,\|\, P_o^2X_t \,\|\, P_iP_oX_t \,\|\, P_oP_iX_t] \quad (1)$$

Thus $Z_t\in R^{nx133}$. All blocks are computed within their respective snapshots; a node never receives a future-time message. A StandardScaler is estimated on labeled initial-training nodes only, applied to all 133 columns, and its outputs are clipped to ±8. Separating incoming from outgoing aggregation is what lets the cached representation distinguish outbound fan-out from inbound convergence, the two illicit roles exchanged by the concept shift of Section IV-A; the operator is computed with sparse matrix products and shared by every graph-embedding baseline.

3.3. Stable–Plastic Classifier and Comparison Rule

A 48–24 ReLU multilayer perceptron is fitted for 24 partial-fit epochs with fivefold positive oversampling, Adam learning rate 10^{-3}, L2 penalty 10^{-3}, and 256-node minibatches. Two copies share initialization; the stable copy f_s is frozen while f_p is updateable. Both are scored before current labels arrive. Unless otherwise stated, the same equal mixture is used for all stable–plastic variants:

$$p_t(v) = ½\, f_s(Z_t[v]) + ½\, f_p(Z_t[v]) \quad (2)$$

The common comparison alarm uses the mean μ_v and sample standard deviation σ_v of ordinary validation-snapshot loss. It fires if $L_t>\mu_v+0.5\sigma_v$; the plastic network then makes five further partial-fit passes over the most recent four labeled snapshots together with a frozen five-step anchor (t=25–29). This loss+anchor rule is the update mechanism of ATGNN-SP [4]—its delayed log-loss gate, equal stable–plastic mixture and anchor replay—run with the shared 48–24 network and five update epochs so that all variants have the same computational cost.

3.4. Class-balanced Drift Evidence

For a batch containing n_+ positive and n_- negative disclosed labels, write $\ell_{+,t}=-n_+^{-1}\Sigma_{y=1}\log\, p_t(v)$ and $\ell_{-,t}=-n_-^{-1}\Sigma_{y=0}\log(1-p_t(v))$. Ordinary mean log loss is $L_t=\pi_t\ell_{+,t}+(1-\pi_t)\ell_{-,t}$, where $\pi_t=n_+/(n_++n_-)$. We replace it with the class-balanced risk:

$$B_t = ½(\ell_{+,t} + \ell_{-,t}), \quad \text{alarm if } B_t>\mu^B_v+0.5\sigma^B_v \quad (3)$$

Under pure prior shift with unchanged class-conditional score distributions, $E[B_t]=½E[\ell|Y=1]+½E[\ell|Y=0]$ does not depend on π_t; $E[L_t]$ generally does. This is an expectation-level statement, not an assertion that a finite sampled graph is invariant. A snapshot with fewer than five observed positive labels cannot reliably form this statistic and does not trigger an update. The validation set alone fixes μ^B_v and σ^B_v. No test loss is used to retune the control multiplier.

3.5. Variance, Finite Labels, And Interpretation of Drift

The class-balanced transformation addresses only one aspect of drift. Define $c_+=E[-\log\, p(X)|Y=1]$ and $c_-=E[-\log(1-p(X))|Y=0]$, assuming the scorer is held fixed over the comparison. If the only population intervention changes $P(Y=1)=\pi$, ordinary log-risk is $R(\pi)=\pi c_++(1-\pi)c_-$ and its derivative with respect to π equals c_+-c_-. Therefore a classifier with unequal class-conditional errors can exhibit an increased overall loss even if neither conditional error changes. The balanced risk $R^B=(c_++c_-)/2$ eliminates the prior weight; differentiating R^B with respect to π gives zero under the stated invariance assumption. This simple calculation is the reason to monitor balanced rather than unconditional prediction surprise.

The same transformation has a variance cost. Let σ_+^2 and σ_-^2 denote within-class log-loss variances. Conditional on positive and negative label counts, and ignoring dependence caused by graph edges, the approximate sampling variance is $Var(B_t)\approx¼(\sigma_+^2/n_++\sigma_-^2/n_-)$. Thus balancing class means does not magically eliminate uncertainty: at an extreme low-positive snapshot, even a large observed loss need not establish a new concept. The requirement $n_+\geq 5$ is a pragmatic reliability floor, not a formal type-I error guarantee. The threshold 0.5 standard deviations above the validation mean is intentionally comparable with the reference, and no sequential multiple-testing guarantee is attached to it. More conservative control limits are reasonable in high-consequence screening.

There is a second distinction between a change alarm and an inference about fraud tactics. Both the unbalanced and

balanced losses are computed using the model's own probabilities. An alarm may originate from miscalibration, model underfit, a label-collection anomaly, or a real change in the conditional topology. The controlled generator supplies a known t=40 intervention to interpret the alarm response. On real streams an observed alarm would require analyst evidence before it could be attributed to strategic illicit behavior. This restriction applies equally to event-triggered deep learning and traditional sequential detectors.

3.6. BCG-5: Bounded Recent Adaptation

BCG-5 combines (3) with at most five updates across the 15-step stream. On each alarm it trains f_p for five epochs on the current and preceding three disclosed snapshots, again using positive oversampling, but omits historical anchor replay. This omission is purposeful for the abrupt role-reversal setting: retaining outdated positive prototypes might impede learning when former outbound-fanout fraud becomes inbound-convergence fraud. A matched BCG-5+anchor ablation changes only the replay set and otherwise uses the same class-balanced alarm and update budget. The data-dependent choice is whether a gate fires; the cap of five is fixed before evaluation.

3.7. Streaming Update Sequence and Leakage Audit

Each evaluation step executes the same deterministic sequence. First, graph-only descriptors and their seven directional paths are computed without reference to any class label. Second, the two branch probabilities are blended and written to disk as the predictions of record. Third, the newly disclosed subset is used to evaluate ordinary and balanced log loss. Fourth, a method-specific gate decides whether to train the plastic MLP on the available historical examples; that changed MLP is visible only at t+1. Finally, the past-snapshot pointer advances. Because the update set includes the just-evaluated batch, saving probabilities before the update is essential; reversing these operations would create prequential leakage and invalidate the reported F1 and AP.

The method does not train a new graph encoder during adaptation. This division allows the cached structural representation and its direction labels to remain fixed while the classifier changes. It also narrows the experimental question: any observed difference among the neural variants comes from scoring composition, drift gating, or the supervised update history, not from a model-specific edge encoder. The static baseline shares the same network initialization and features; the positive oversampling strategy is used in training and updates but never in evaluation. The chronological scaler and F1 threshold likewise remain frozen after step 34.

Sparse feature propagation costs $O(|E_t|d)$ per directional multiplication; the seven channels require a constant number of multiplications. After caching, scoring costs $O(|V_t|(133\times48+48\times24))$ per branch, while retraining is charged only to alarmed snapshots. No timing claim includes transaction ingestion or full streaming graph construction.

3.8. Two-step Disclosure Lag and Resampling Diagnostic

We conduct two orthogonal robustness checks. In the disclosure-lag experiment, labels from step t become available only after predictions at t+2 have been saved. The plastic model and drift alarm therefore use previously scored batches that have actually matured; unobserved labels cannot enter its recent-window update. Both loss+anchor and BCG-5 retain their original validation-calibrated thresholds and update policies. A zero-lag rerun is checked numerically against the main stored predictions for each seed, providing a regression test of the additional code path.

A second diagnostic makes the prior-invariance premise testable without confounding the graph generator. We hold each seed's static scoring distributions from snapshot 35 fixed conditional on the label and resample 1,200 batches of 400 scores for each positive-prior setting in {0.02, 0.05, 0.10, 0.20, 0.30}. A disjoint simulated control sample at prior 0.10 establishes each statistic's mean-plus-half-standard-deviation alarm threshold. This diagnostic is counterfactual stratified resampling, not another detector trained on a new transaction dataset: conditional scores, their topology dependence, and the classifier are all unchanged by construction.

4. Controlled Experimental Design

4.1. Synthetic Directed Transaction Graph Generator

The experiments use explicitly synthetic, reproducible transactions so that the type and timing of every drift are known; the Elliptic benchmark [1], on which ATGNN-SP was evaluated, provides real labels but no ground-truth drift annotations. Each of 49 snapshots contains 370 nodes. Latent types represent balanced flow, outbound fan-out, and inbound convergence. Outdegree is Poisson with mean 4.2, 11.0, or 1.7 by type; edge destinations are sampled without replacement with corresponding destination attractiveness 4.0, 1.9, or 11.0 and a mild illicit-to-illicit affinity. Only 62% of nodes are selected for label disclosure independently of the target; all nodes remain available for unsupervised topology propagation.

In the initial regime, a licit node is assigned balanced/fan-out/convergence type with probabilities 0.78/0.11/0.11, while an illicit node is fan-out with probability 0.94 and convergence otherwise. At $t\geq40$, the concept-shift generator switches illicit roles to 0.10 fan-out and 0.90 convergence. Its benign role law is unchanged. The baseline illicit probability is 0.10, and prior-perturbation scenarios change it at t={36,37,38,45,46,47} to {0.035,0.23,0.025,0.24,0.018,0.19}, respectively. Edge sampling may cause secondary shifts in neighborhood statistics when priors change; consequently the prior-only condition is an intervention on the generator prior rather than a mathematical perfect-label-shift construction.

Three regimes use identical initial training/validation generators: prior-only (prevalence perturbations, no role reversal), concept-only (reversal at t=40, constant class prior), and mixed (both interventions). Five independent integer seeds {7, 42, 2026, 2027, 2028} control data generation and model initialization. For each seed, models see identical graphs, observed labels, and chronological boundaries, allowing paired contrasts without randomly mixing historical and future nodes.

The directed edge sampler assigns a Poisson number of distinct outgoing links to each transaction and chooses recipient nodes using normalized attractiveness weights. A fraud-originating node doubles the attractiveness of illicit recipients relative to the same type of licit recipients. The induced neighbor distributions are therefore intentionally correlated with labels; they are not arbitrary Erdős–Rényi edges with random fraud tags. Crucially, this data-generating

relation can be learned using only graph-derived predictors. In the initial snapshots, outward fan-out is positive evidence of illicit behavior; after the conditional intervention, inbound convergence becomes the dominant positive signal. The mixture of benign high-flow roles creates false positives, making the task nontrivial instead of perfectly separable.

Every random seed regenerates the graph, label, missing-label indicators, model parameter initialization, and within-batch oversampling indices. A fixed 370-node size makes structural effects easier to attribute to regime interventions instead of changing network size. The 49 individual snapshots are disconnected, which mirrors the computational assumption of snapshot-local propagation without asserting the topology of any real Bitcoin graph. The experiment logs edges and realized class prevalence for every snapshot, not just expected probabilities, so researchers can distinguish generator settings from realized samples.

Table I. Measured synthetic data partitions (five-seed means)

Steps	Nodes	Observed	Positive
1–29	10730	6617.8	654.0
30–34	1850	1149.4	123.8
35–49	5550	3425.6	369.8

4.2. Baselines, Ablations, And Metric Protocol

Baselines include a permanently static directed-graph MLP, the adapted ordinary-loss/anchor method, an always-updated stable–plastic model, a plastic-only branch with ordinary-loss alarms, BCG-5+anchor, and BCG-5. Histogram gradient boosting is trained either on the 133-D directed representation or on the 19 unpropagated node descriptors; its role is to test whether graph structure and online plasticity add value beyond strong non-neural supervised classification. Tree ensembles [26] and gradient boosting [27] are well-established alternatives to learned embeddings. All models are implemented with scikit-learn [28].

A single F1-maximizing decision threshold is selected for each model on validation predictions, then held constant in testing. Snapshot probabilities are saved before their associated labels are disclosed, so cumulative AP, ROC-AUC, F1, precision, and recall represent strict prequential predictions. AP is primary because it integrates ranking quality over the precision–recall curve without relying on one chosen threshold [29]. Results are means and sample standard deviations across five seeds; no inferential significance test is claimed from five related synthetic replications.

The scripts retain actual prediction probabilities, per-step labels, alarms, confusion counts, training/update timings, environment versions, and complete generator settings. In the mixed scenario each seed produces 18,130 synthetic nodes in total; the mean test interval contains 3,425.6 observed labels (369.8 positives), and the source code reconstructs every realization deterministically.

5. Results and Analysis

5.1. Detection Quality Under Distinct Drift Types

Table II presents complete mixed-regime scores. BCG-5 reaches AP 0.3175 ± 0.0366 and F1 0.3413 ± 0.0693, improving the ATGNN-SP update rule (loss+anchor) by 0.0171 AP and 0.0046 F1 in paired mean differences. These improvements occur in all five mixed-regime seeds, although the F1 gain is small relative to between-seed variation. The ATGNN-SP update rule itself raises AP from 0.1958 for the static graph MLP to 0.3004 under this designed role inversion, so most of the adaptation benefit comes from its stable–plastic design and BCG-5 refines when that design updates. The plastic-only detector ranks substantially better than the stable–plastic systems (AP 0.3709, F1 0.4361).

Table II. Mixed regime: held-out prequential scores, five seeds

Method	AP	F1	Updates
Static graph MLP	0.196	0.246	0.0
Graph HGB	0.212	0.257	0.0
Loss+anchor	0.300	0.337	8.0
BCG-5+anchor	0.296	0.324	5.0
BCG-5 (ours)	0.317	0.341	5.0
Plastic-only	0.371	0.436	8.4

5.2. Seed-Level Reproducibility of the Mixed Regime

Table III makes the paired nature of the mixed-regime result visible. An individual model seed is coupled with a distinct graph realization but reused by all candidate methods within that realization. BCG-5 outperforms the loss+anchor alternative in AP for seeds 7, 42, 2026, 2027, and 2028. The AP difference is most pronounced in seed 2026 (+0.029) and smallest in seed 7 (+0.003); the method therefore does not rely solely on its best-performing run. A caution is that the absolute AP ranking depends strongly on the rarity, directed motifs, and random edge placements in each synthetic realization.

Table III. Mixed-regime seed-level AP and update counts

Seed	Anchor	BCG-5	Δ AP	Updates
7	0.286	0.289	+0.003	5/9
42	0.265	0.278	+0.013	5/7
2026	0.341	0.370	+0.029	5/8
2027	0.305	0.332	+0.027	5/8
2028	0.305	0.319	+0.014	5/8

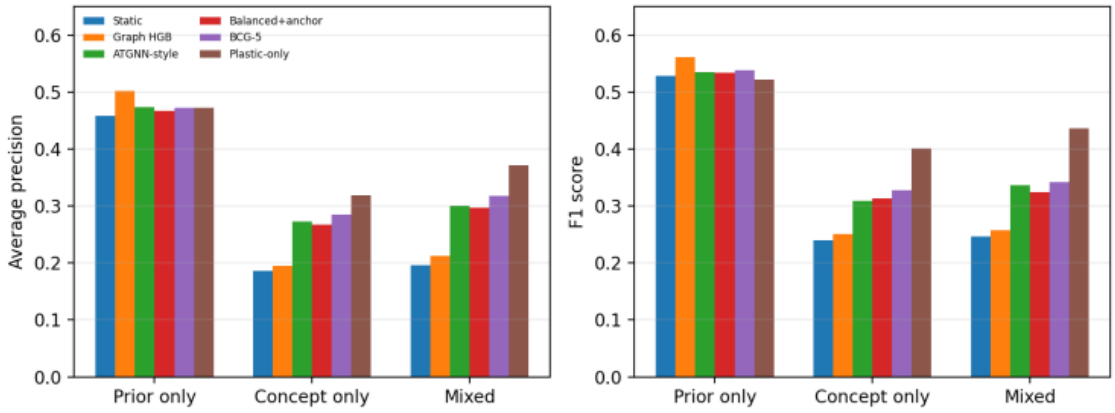


Figure 1. Five-seed mean AP and F1 on three controlled regimes. Error magnitudes are reported in the supplementary results.

The concept-only regime tells a similar but slightly stronger story: BCG-5 improves AP from 0.2726 to 0.2842 and F1 from 0.3092 to 0.3275 against the loss-gated anchor baseline. In the prior-only regime the new method has AP 0.4722 versus 0.4736 for the ordinary-loss baseline, with F1 0.5384 versus 0.5351. Thus the balanced statistic and recent-only update neither reliably win nor meaningfully lose under pure prevalence intervention; their clearest ranking benefit emerges when the fraud-generating graph roles actually reverse.

5.3. Update Budget and Computational Cost

BCG-5 uses 5.0 updates in the mixed setting versus 8.0 for the loss-gated baseline, a 37.5% reduction. Under concept-only drift the corresponding means are 4.8 and 6.0. The mean measured cumulative plastic fitting time is 0.0864 s for BCG-5 and 0.2756 s for the replaying baseline in the mixed setting, a 68.7% reduction on the same hardware. Because the update training sets differ in size, this is a total measured fitting-cost comparison, not a hardware-independent per-transaction latency result.

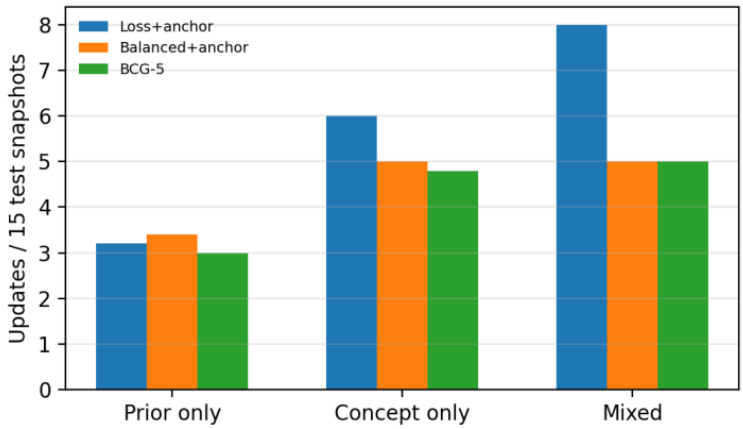


Figure 2. Mean update counts across the 15 evaluated snapshots. BCG-5 is capped at five.

The always-updated model incurs 15 updates with mean mixed AP 0.2963, below BCG-5. The result highlights that unselective additional fitting need not improve a frozen-plus-plastic mixture. In prior-only drift, graph HGB obtains AP 0.5023 and F1 0.5616, outperforming all neural methods; this negative result limits claims that the proposed architecture is universally preferable to non-neural graph-aware classification.

5.4. Ablation: Historic Replay Versus Recent Evidence

BCG-5+anchor reinstates the last five initial-training snapshots while retaining the balanced alarm and update cap. In the mixed setting its AP falls from 0.3175 to 0.2963 and F1 from 0.3413 to 0.3244; in the concept-only setting AP falls from 0.2842 to 0.2670. Although historical replay is sensible when retaining an older concept is valuable, it conflicts with the synthetic reversal in which positive nodes change directed flow roles sharply. By contrast, the long-term stable predictor remains present in the mixture, providing a reference path even when the plastic optimizer trains only on recent data.

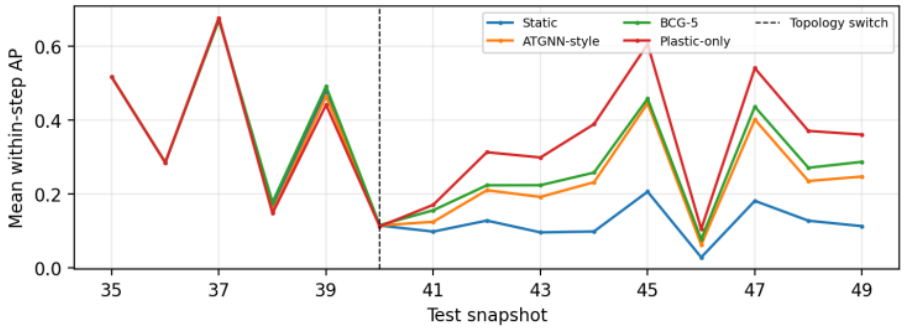


Figure 3. Mean within-snapshot AP in the mixed stream, with the programmed conditional topology switch at step 40.

The plastic-only result is an important counterexample to the assumption that preserving a stable prediction branch always improves minority-class detection. It has fewer constraints after drift and obtains greater recall on the inverted fraud role, at the expense of additional updates and no frozen prediction branch. This comparison suggests that blend weighting is a separate research variable and that the present contribution is primarily update efficiency within a fixed 50:50 stable–plastic architecture.

5.5. Uncertainty and Control Checks

Paired seed-wise AP improvements of BCG-5 over the ordinary-loss baseline are positive in four of five concept-only replications and all five mixed replications. Their mean differences are +0.0115 and +0.0171, with sample standard deviations 0.0120 and 0.0109. These comparisons are descriptive: the five graph generators differ in stochastic realizations but share a common scenario definition. The signal is persuasive for the programmed synthetic topology shift, not sufficient evidence of cross-domain transfer. The synthetic prior perturbations sometimes alter class-conditional neighbor features indirectly, and observed-label sampling introduces finite-batch variance in both loss monitors.

5.6. Temporal Trajectories and Diagnostic Evidence

Within-snapshot AP in Fig. 3 is averaged across the five seeds only after computing AP separately for each step. The curve is intentionally distinct from the global AP of Table II: the latter concatenates all scored nodes, whereas the figure gives each snapshot equal visual weight. An isolated snapshot can have extremely few positive disclosures, so its AP has high variance and should not be read as a precise risk estimate. In particular, the later prior peaks and troughs do not map to monotonic changes in AP because the generator also changes neighbor sampling density as its positive pool varies.

The programmed change at step 40 provides a point of reference for the learning dynamics. Both gated methods initially score with parameters optimized for illicit fan-out; immediately after the inversion the outdated classifier cannot fully recognize illicit convergence. Plastic updates are then allowed to reorient the decision surface, with the stable branch remaining frozen. The observed benefit is distributed across several subsequent snapshots rather than occurring before labels are revealed. No test-step score in Fig. 3 incorporates a model update trained on that same step, a distinction verified by the stored predictions and per-step alarm traces.

We interpret the difference between BCG-5 and BCG-5+anchor as a consequence of conflicting historical targets: an anchor example labeled illicit remains an outbound fan-out node, whereas recent illicit examples increasingly resemble inbound convergence. An optimizer that interleaves both patterns learns a compromise. Omitting replay reduces that direct conflict, but the stable half of the prediction can still vote for earlier relationships. Under recurrent drift, catastrophic forgetting might make recent-only updates worse; this experiment contains one irreversible conditional inversion, not repeated cycles. Thus the ablation supports a conditional design recommendation rather than a universal rejection of anchor replay.

5.7. When Did the Delayed-Label Gates Respond?

The alarm traces give a more diagnostic view than the aggregate update counts. In the mixed regime, the first programmed conditional intervention is step 40. Both the ordinary-loss gate and BCG-5 fire at this step for all five seeds, showing that the class-balanced monitor does not simply suppress true change evidence to save updates. Their subsequent behavior differs because BCG-5 exhausts its budget after at most five confirmed excess-loss events. For instance, in seed 42 the ordinary-loss alarm updates at steps 37, 40, 41, 42, 43, 45, and 47, whereas BCG-5 updates at 37, 38, 40, 41, and 42. In seed 2028 the respective event sets are $\{37, 40, 41, 42, 43, 45, 47, 48\}$ and $\{36, 40, 41, 42, 43\}$. These are recorded predictions-first events, not labels of true change points.

In the concept-only setting, BCG-5 fires at the actual change step 40 for each seed. Ordinary-loss monitoring does so as well. The main observed difference is the lower frequency of fitting after the sudden reversal, not a proven superiority in change-point localization. Under prior-only intervention, BCG-5 still fires 3.0 times on average (versus 3.2 for the loss+anchor baseline), despite the absence of a programmed conditional switch. This is expected because the finite-sample class-balanced loss remains noisy, and graph-neighborhood context may itself respond to the illicit fraction through edge sampling. These concrete traces delimit the efficacy of class balancing: it cancels mixture weights in expectation but cannot fully isolate the nature of a finite graph-stream distribution change.

5.8. Disclosure-lag Robustness

A two-snapshot disclosure lag changes the timing of every eligible alarm and supervised refresh, rather than simply omitting labels from the loss statistic. Table IV shows all five mixed-regime realizations under this stricter information schedule. BCG-5 average precision falls from 0.3175 to 0.2873 when feedback arrives two snapshots later; loss+anchor falls from 0.3004 to 0.2780. The delayed BCG-5 advantage over loss+anchor is 0.0094 AP, while its F1 advantage is only 0.0017. Thus the direction of the AP

contrast persists in these realizations, but both methods become less effective as feedback becomes stale. The delay-free results are reproduced exactly by the new evaluation routine.

Table IV. Mixed regime with two-step label disclosure (five-seed means)

Method	Lag	AP	F1	Updates
Loss+anchor	0	0.300	0.337	8.0
Loss+anchor	2	0.278	0.318	7.8
BCG-5	0	0.317	0.341	5.0
BCG-5	2	0.287	0.319	5.0

The label-lag stress test is limited to a constant two-step delay with full disclosure of the original observed-label subset. In real investigations, delays are heterogeneous and can correlate with suspiciousness, so the effect on class-balanced monitoring may be more severe than suggested here. The saved maturity-step alarm log supports auditing exactly when each supervised update occurred.

5.9. Class-prior Isolation Diagnostic

Figure 4 reports the conditional-resampling results, where the fixed score distributions are sampled under different positive fractions. Ordinary mean log loss rises from 0.2493 at prior 0.02 to 0.3762 at prior 0.30, whereas mean balanced loss remains in the narrow range 0.4664–0.4687. At the reference prior 0.10 the ordinary and balanced alarm frequencies are 0.2947 and 0.2875, respectively. At prior 0.30 they are 0.8517 versus 0.1938. These observations isolate the algebraic advantage of balanced risk under truly fixed class-conditional scores; the residual change in balanced alarm frequency reflects prevalence-dependent estimator variance rather than drift in its expected risk.

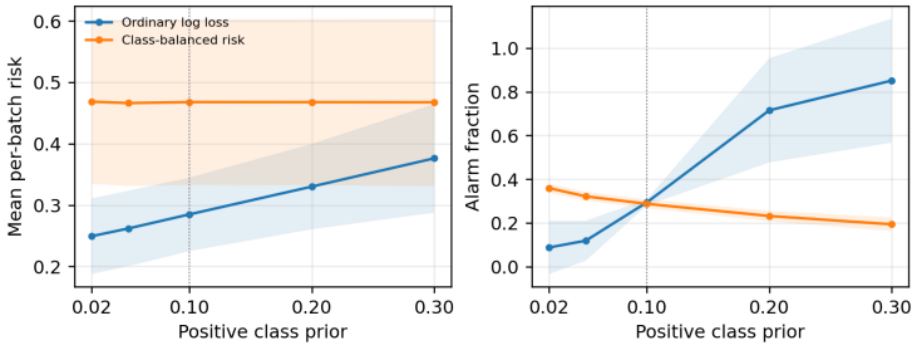


Figure 4. Class-conditional score resampling: empirical mean risk and alarm frequencies. Five seeds, 1,200 draws each per prior; thresholds calibrated at prior 0.10.

This isolation check also explains why class balancing is not a complete solution. When only a handful of positives are available, the minority log-loss estimate fluctuates strongly even if its population mean is fixed. As a result the diagnostic supports mean invariance, not equalized false-alarm probabilities. Estimating uncertainty explicitly or requiring more disclosed positives would be needed to control the finite-sample alarm rate.

5.10.Statistical Uncertainty for Paired Runs

Across the five paired mixed-regime runs, the AP difference is +0.0171, with a nominal 95% Student-t confidence interval [0.0036, 0.0305] for synthetic-generator replications. The exact two-sided paired sign-flip p-value is 0.0625, reflecting the coarse 2^5 permutation resolution. For two-step delayed labels the paired AP difference is +0.0094, with interval [0.0039, 0.0148] and exact p=0.0625. These analyses quantify variability rather than supporting a broad significance claim for real financial data. The two metrics and delay conditions were not preregistered as a confirmatory hypothesis family.

5.11.Meaning of the Three-Way Method Comparison

Three comparisons answer separate scientific questions. Static versus loss+anchor measures the benefit of any selected plasticity while holding the directed encoder fixed. Loss+anchor versus BCG-5+anchor asks what changes when the gate becomes class-balanced and its event budget is capped, but replay remains. Finally BCG-5+anchor versus BCG-5 isolates the change from anchor replay to recent-only adaptation under the same balanced gate and maximum five updates. In the mixed setting, BCG-5+anchor reaches AP 0.2963, below loss+anchor 0.3004, indicating that the gate-plus-cap alone is not sufficient to explain BCG-5's improvement. The substantial advantage emerges with the removal of anchor examples during the programmed role reversal. This factorial reading avoids attributing an observed gain to the wrong algorithmic component.

The choice to include tree baselines is similarly consequential. Histogram boosting on the unpropagated 19 descriptors reaches mixed AP 0.2088, and the propagated 133-dimensional version reaches AP 0.2119; both trail the gated neural approaches under the concept reversal. In prior-only perturbations, however, graph boosting reaches AP 0.5023, confirming that powerful non-neural estimators may generalize well without online neural fitting when the supervised relation remains more stable. The relative ranking depends on whether the experiment changes only how often a fraud signature occurs or changes which directed signature is fraudulent. Real-world evaluation must determine which of these mechanisms dominates a particular financial system.

6. Discussion and Limitations

The experiment directly isolates a methodological issue: when positive prevalence changes sharply, an unbalanced log-loss monitor can move simply because the mixture coefficient π_t changes. Conditional reweighting removes that algebraic dependence in expectation, and the five-update cap makes adaptation cost explicit. BCG-5 furthermore demonstrates that anchor replay need not help under a constructed abrupt role inversion. The central design choice is therefore not merely whether to retrain but which risk statistic, history

horizon, and update budget should govern retraining.

Three limitations are consequential. First, all predictive metrics are from deliberately simplified synthetic flows; the generator does not capture real financial amounts, heterogeneous identities, investigators, or adaptive adversaries. Second, the main protocol discloses labels after each snapshot, with a supplementary two-step lag stress test; all disclosed labels are sampled completely at random, whereas operational investigations can generate prolonged, selective, and biased feedback. Third, full snapshot topology features and clustering are precomputed before scoring, so claimed fitting times exclude graph ingestion and feature extraction. A replication on the real Elliptic benchmark, on which ATGNN-SP has been evaluated, is needed to establish empirical utility outside the controlled mechanism and to compare absolute scores.

The evaluation also contains a deliberately strong challenger: the plastic-only branch substantially improves detection during role reversal, and graph HGB is superior in the prior-only scenario. Future work should allocate stable/plastic weights by validation-calibrated evidence, formalize class-balanced sequential alarms with small-sample guarantees, and test selectively missing and longer-delayed labels on public and permissioned transaction streams. Under those extensions, the same direction-aware encoder and predict-before-update protocol offer a clear starting point for auditable investigation.

These results additionally clarify the operational meaning of an update budget. Limiting retraining events differs from fixing a wall-clock time budget because the number of disclosed labels in the four-snapshot recent window varies. BCG-5 consumes less fitting time partly because it omits the five training anchors, and partly because it fires less often after exhausting the five-event cap. The cap does not select the five globally most useful alarms; it simply accepts the first five that cross the validation limit. A more sophisticated online budget controller might reserve capacity for large later shifts, but such a policy would require an explicit horizon and a prospective optimization rule rather than retroactive selection of favorable alarms.

The use of global threshold maximization is also deliberate. The validation distribution differs from the last test regime, which causes threshold-sensitive F1 to vary much more than AP. Retuning a threshold after each disclosed test batch would be another valid online learning algorithm, but it would introduce a second adaptation mechanism and make the gate ablation harder to interpret. Our F1 estimates therefore assess a fixed operating point selected on time steps 30–34, while AP assesses the ranking across all possible operating points. For deployment, investigator capacity and asymmetric false-positive costs would determine the preferred threshold, and false discovery constraints might dominate either benchmark metric.

7. Conclusion

BCG-5 extends the directed seven-channel SGC and delayed stable–plastic fraud scorer of ATGNN-SP with class-balanced monitoring, recent-only fitting, and a bounded update count. Across five reproducible synthetic realizations of an abrupt directed-flow role reversal, it improves AP over the ATGNN-SP update rule while using fewer updates. It does not dominate all competitors: unconstrained plastic-only learning is more accurate under the designed reversal, and graph boosting is stronger under prior-only perturbations. The supported result is a targeted efficiency–accuracy trade-off for a controlled temporal-graph drift mechanism, ready to be tested—not assumed valid—on genuine financial transaction records.

References

[1] Weber, M., Domeniconi, G., Chen, J., Weidele, D. K. I., Bellei, C., Robinson, T., & Leiserson, C. E. (2019). Anti-money laundering in Bitcoin: Experimenting with graph convolutional networks for financial forensics. arXiv. https://doi.org/10.48550/arXiv.1908.02591

[2] Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. ACM Computing Surveys, 46(4), Article 44. https://doi.org/10.1145/2523813

[3] Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., & Weinberger, K. (2019). Simplifying graph convolutional networks. Proceedings of the 36th International Conference on Machine Learning, 97, 6861–6871.

[4] Yue, X., Yang, J., Zhu, R., & Xu, Q. (2026). Adaptive temporal graph neural networks for real-time financial fraud detection under concept drift. Highlights in Business, Economics and Management, 69, 239–250. https://doi.org/10.54097/c8364y03

[5] Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., & Zhang, G. (2019). Learning under concept drift: A review. IEEE Transactions on Knowledge and Data Engineering, 31(12), 2346–2363. https://doi.org/10.1109/TKDE.2018.2876857

[6] Bifet, A., & Gavaldà, R. (2007). Learning from time-changing data with adaptive windowing. Proceedings of the 7th SIAM International Conference on Data Mining, 443–448. https://doi.org/10.1137/1.9781611972771.42

[7] Page, E. S. (1954). Continuous inspection schemes. Biometrika, 41(1–2), 100–115. https://doi.org/10.1093/biomet/41.1-2.100

[8] Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLOS ONE, 10(3), Article e0118432. https://doi.org/10.1371/journal.pone.0118432

[9] He, H., & Garcia, E. A. (2009). Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, 21(9), 1263–1284. https://doi.org/10.1109/TKDE.2008.239

[10] Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. International Conference on Learning Representations. https://doi.org/10.48550/arXiv.1609.02907

[11] Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Inductive representation learning on large graphs. Advances in Neural Information Processing Systems, 30, 1024–1034.

[12] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph attention networks. International Conference on Learning Representations. https://doi.org/10.48550/arXiv.1710.10903

[13] Pareja, A., Domeniconi, G., Chen, J., Ma, T., Suzuki, T., Kang, H., Kanezashi, H., Kaler, T., Schardl, T. B., & Leiserson, C. E. (2020). EvolveGCN: Evolving graph convolutional networks for dynamic graphs. Proceedings of the AAAI Conference on Artificial Intelligence, 34(4), 5363–5370. https://doi.org/10.1609/aaai.v34i04.5984

[14] Rossi, E., Chamberlain, B., Frasca, F., Eynard, D., Monti, F., & Bronstein, M. (2020). Temporal graph networks for deep learning on dynamic graphs. arXiv. https://doi.org/10.48550/arXiv.2006.10637

[15] Kumar, S., Zhang, X., & Leskovec, J. (2019). Predicting dynamic embedding trajectory in temporal interaction networks. Proceedings of the 25th ACM SIGKDD

International Conference on Knowledge Discovery and Data Mining, 1269–1278. https://doi.org/10.1145/3292500.3330895

[16] Sankar, A., Wu, Y., Gou, L., Zhang, W., & Yang, H. (2020). DySAT: Deep neural representation learning on dynamic graphs via self-attention networks. Proceedings of the 13th ACM International Conference on Web Search and Data Mining, 519–527. https://doi.org/10.1145/3336191.3371845

[17] Dou, Y., Liu, Z., Sun, L., Deng, Y., Peng, H., & Yu, P. S. (2020). Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. Proceedings of the ACM International Conference on Information and Knowledge Management, 315–324. https://doi.org/10.1145/3340531.3411903

[18] Liu, Y., Ao, X., He, Q., Li, Z., Yang, M., & He, X. (2021). Pick and choose: A GNN-based imbalanced learning approach for fraud detection. Proceedings of The Web Conference, 3168–3177. https://doi.org/10.1145/3442381.3449989

[19] Rao, S. X., Zhang, S., He, Z., Lee, K., & Chen, Y. (2022). xFraud: Explainable fraud transaction detection. Proceedings of the VLDB Endowment, 15(3), 427–436. https://doi.org/10.14778/3494124.3494128

[20] Lu, M., Zhang, Y., Zhang, Y., Liu, Y., & Li, J. (2022). BRIGHT—Graph neural networks in real-time fraud detection. Proceedings of the ACM International Conference on Information and Knowledge Management, 3342–3351. https://doi.org/10.1145/3511808.3557136

[21] Elmougy, Y., & Liu, L. (2023). Demystifying fraudulent transactions and illicit nodes in the Bitcoin network for financial forensics. Proceedings of the 29th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 3979–3990. https://doi.org/10.1145/3580305.3599803

[22] Lorenz, J., Silva, M. I., Aparício, D., Ascensão, J. T., & Bizarro, P. (2020). Machine learning methods to detect money laundering in the Bitcoin blockchain in the presence of label scarcity. Proceedings of the ACM International Conference on AI in Finance, 1–8. https://doi.org/10.1145/3383455.3422549

[23] Deprez, B., Vanderschueren, T., Baesens, B., Verdonck, T., & Verbeke, W. (2025). Network analytics for anti-money laundering—A systematic literature review and experimental evaluation. INFORMS Journal on Data Science, 5(2), 119–154. https://doi.org/10.1287/ijds.2024.0042

[24] Akoglu, L., Tong, H., & Koutra, D. (2015). Graph based anomaly detection and description: A survey. Data Mining and Knowledge Discovery, 29, 626–688. https://doi.org/10.1007/s10618-014-0365-y

[25] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys, 41(3), Article 15. https://doi.org/10.1145/1541880.1541882

[26] Breiman, L. (2001). Random forests. Machine Learning, 45, 5–32. https://doi.org/10.1023/A:1010933404324

[27] Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. The Annals of Statistics, 29(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

[28] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

[29] Davis, J., & Goadrich, M. (2006). The relationship between precision-recall and ROC curves. Proceedings of the 23rd International Conference on Machine Learning, 233–240. https://doi.org/10.1145/1143844.1143874