ON THE EFFECTIVENESS OF RESAMPLING FOR SUPERVISED GRAPH ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

When does resampling help supervised graph anomaly detection? This empirical study compares resampling across 17 benchmarks and two tree-ensemble learners, holding graph-derived features and labelled splits fixed. We use allocation ($\mathcal{A}$), transformation ($\mathcal{T}$), and the weight rule ($\mathcal{M}$) to distinguish where resampling gains arise. The clearest gains occur on injected anomalies: simple feature perturbation improves test AUPRC over class weighting by 5.22 points with XGBoost and 4.40 with Random Forest; its smaller organic gains have confidence intervals including zero. Under scarce labels, labelled-split variation exceeds strategy variation in the organic panel: a descriptive decomposition assigns 72% versus 10% to these factors for XGBoost, and 68% versus 16% for Random Forest. Adaptive controls establish a small gain from residual-guided allocation and weighting, while the additional benefit of structural transformation remains unresolved. These findings motivate separating anomaly provenance, repeating paired labelled splits, and testing transformations against unchanged copies with matched allocation and class mass.

1 INTRODUCTION

Graph anomaly detection (GAD) identifies unusual nodes, edges, or substructures in applications such as fraud detection and cybersecurity. We study supervised node anomaly detection when few anomalies are labelled. Our contribution is an empirical study of when resampling helps and which of its mechanisms account for the observed gains. In this setting, GADBench (Tang et al., 2023) found that tree ensembles trained on graph-derived features can match or outperform specialized graph neural networks at lower computational cost. LightGAD (Liu et al., 2026) provides further evidence for simple detectors. These findings motivate studying how to use scarce labels more effectively.

Resampling changes training observations or their weights. Methods range from duplication and feature perturbation to graph generation and structural views. Their benefits are hard to isolate when studies also change the detector, representation, and evaluation protocol. We ask: *When does resampling improve supervised graph anomaly detection?*

We compare eight organic benchmarks with nine benchmarks containing injected anomalies, using common graph-derived features, learner settings, and paired labelled splits. Both unweighted and class-weighted training serve as references. A six-dataset budget study and full-supervision evaluations complement the main scarce-label setting.

Our organizing lens is **allocation, transformation, and weight rule** ($\mathcal{A}/\mathcal{T}/\mathcal{M}$): *which observations receive the resampling budget, how their representations change, and how much training weight they receive*. In adaptive resampling, this lens yields matched controls that distinguish selecting and reweighting difficult examples from creating new representations. The evaluated samplers instantiate these choices and provide evidence for three empirical findings:

1. **Resampling helps most clearly on injected anomalies.** Simple feature perturbation gains 5.22 and 4.40 AUPRC points over class weighting with XGBGraph and RFGraph. Its smaller organic gains remain uncertain against both references (Section 4.1).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2. **Under scarce labels, split composition matters more than strategy choice on average.** In the organic panel, labelled splits account for 72% of within-dataset AUPRC variation versus 10% for strategy with XGBGraph, and 68% versus 16% with RFGraph (Section 4.1).
3. **Adaptive allocation and weighting yield a small gain; structural transformation adds no established benefit.** Residual weighting improves organic and injected AUPRC by 0.20 and 0.33 points, respectively. Replacing matched copies with structural views leaves the additional gain unresolved (Section 4.2).

Source code, configurations and the complete per-trial result archive are available at `https://anonymous.4open.science/r/ICLR40454/`.

2 Related Work

Supervised graph anomaly detection. Recent supervised graph anomaly detectors have primarily relied on graph neural networks to model the heterophilous and highly imbalanced structure of fraud and security graphs, with BWGNN (Tang et al., 2022) being a representative example. However, the large-scale benchmark GADBench (Tang et al., 2023) showed that gradient-boosted trees trained on graph-derived representations often match or outperform specialized GNNs while requiring substantially less computation and hyperparameter tuning. These methods construct graph-enhanced features $\Phi = [X \,\|\, PX \,\|\, P^2X]$, where P averages incoming neighbours, before fitting a standard tree ensemble. LightGAD (Liu et al., 2026) follows the same philosophy with adaptive feature aggregation. We therefore adopt a common graph-derived representation to compare resampling strategies under controlled downstream learners.

Graph resampling under class imbalance. Class imbalance is commonly addressed either by modifying the learning objective (e.g., Balanced Random Forest (Chen et al., 2004)) or by resampling the training data. Existing tabular resampling spans a spectrum from exact duplication (ROS), to perturbation (ROSE (Menardi & Torelli, 2014)), to interpolation (SMOTE (Chawla et al., 2002) and ADASYN (He et al., 2008)). Although highly effective on tabular data, these methods ignore graph structure. We evaluate this complete spectrum to determine how well these resampling philosophies transfer to supervised graph anomaly detection.

Graph-aware and adaptive resampling. Graph-aware methods explicitly exploit graph structure during resampling. GraphSMOTE (Zhao et al., 2021), DAGAD (Liu et al., 2022a), GODM (Liu et al., 2024), and GenGA (Meng et al., 2023) generate synthetic graph observations, whereas ShaDow-GNN (Zeng et al., 2021) and Forest Fire Sampling (Leskovec & Faloutsos, 2006) instead modify the graph context of existing nodes. Resampling also differs in *allocation*, i.e., which observations receive the available budget. ADASYN (He et al., 2008) allocates more synthetic samples to locally difficult minority observations; GOSS (Ke et al., 2017) prioritizes observations using gradient information; and graph methods such as GraphSHA (Li et al., 2023) and GRAPES (Younesian et al., 2025) adapt sample construction or selection using difficulty or structural information. GraphENS (Park et al., 2022) synthesizes ego-networks to address minority-node and neighbour memorization in GNNs. Other methods modify the learning objective: TAM (Song et al., 2022) adjusts margins using local topology, whereas ReNode (Chen et al., 2021) reweights labelled nodes according to their structural positions relative to class boundaries. These approaches address graph imbalance beyond the resampling operators evaluated here.

Existing approaches typically couple sample selection and transformation, making it difficult to isolate whether gains arise from *which* labelled observations are selected, *how* they are transformed, or changes in their effective training mass. Our framework addresses this gap through controlled evaluation and, for adaptive resampling, separates *allocation*, *transformation*, and *weight redistribution*.

3 Allocation, Transformation, and Weight: An Empirical Lens

The $\mathcal{A}/\mathcal{T}/\mathcal{M}$ lens separates source selection, representation changes, and training weight. To compare these choices, we hold canonical graph-derived features and the downstream learner fixed, then vary the resampling operator.

3.1 Common Resampling Interface

Let G be a graph with node set V and node features X, and let $V_L \subseteq V$ be the labelled training nodes, with $y_i \in \{0, 1\}$ (1 denotes an anomaly). Write Φ for the matrix of graph-derived features, ϕ_i for its row at node i, and Φ_L and y_L for the labelled rows and labels. The labelled training data are (Φ_L, y_L, w), with one nonnegative weight w_i per labelled node. A tilde denotes the resulting training rows, labels, and weights after resampling; their number may differ from $|V_L|$. A graph resampling method is an operator $\mathcal{R}$ that changes the training supervision. Structural operators also use G and X. Learner-guided allocation additionally uses a predictor, instantiated by the warm-up model in the residual-guided controls. Static operators do not use this predictor. Sampler-only contrasts hold the learner configuration fixed.

Within the framework, a graph resampling configuration is defined by $(\Phi, f, \mathcal{R})$, where Φ denotes the graph-derived representation, f the downstream learner, and $\mathcal{R}$ the resampling strategy. The corresponding learning pipeline is

$$(G, X) \longrightarrow \Phi \longrightarrow (\Phi_L, y_L, w) \xrightarrow{\mathcal{R}} (\widetilde{\Phi}, \widetilde{y}, \widetilde{w}) \longrightarrow f.$$

Holding the representation and learner fixed while varying the resampling strategy isolates its effect, which is the main objective of this paper. Holding the representation and resampling strategy fixed while varying the learner tests whether that effect is learner-dependent.

3.2 Common Graph-Derived Representation

The tree-based comparisons use canonical features computed on the original graph, following GAD-Bench, $\Phi = [X \| PX \| P^2X]$. Here P uniformly averages incoming neighbours in the graph used for feature computation, with zero rows for empty neighbourhoods, and $\|$ denotes column concatenation. The feature construction and canonical rows Φ stay fixed. Resampling changes training weights or rows. Island views recompute only the added rows on sampled graph contexts, using the same feature construction. Validation and test nodes retain their canonical rows.

3.3 Reference Graph Detectors

The graph-derived representation is coupled with a supervised learner. For node i, its predicted anomaly probability is $p_i = f(\phi_i)$, where f denotes the fitted tabular classifier returning the probability of class 1. Throughout this work we instantiate two reference detectors: (1) **XGBGraph** (XGBoost (Chen & Guestrin, 2016) trained on the graph-derived representation Φ); and (2) **RF-Graph** (Random Forest (Breiman, 2001) trained on the same graph-derived representation). Both detectors share the same features, allowing us to test compatible resampling strategies for learner dependence. We additionally evaluate two configurations of XGBGraph: a reference configuration with 100 trees and learning rate $\eta = 0.3$, and an alternative with 200 trees and $\eta = 0.05$, both at depth six. We measure each resampling gain against its configuration's class-weighted control. Because tree construction and refresh timing also change, this comparison assesses sensitivity to the overall detector configuration (Appendix A.1).

3.4 Graph Resampling Strategies

We instantiate $\mathcal{R}$ through the following families of changes to the labelled training distribution.

Identity operator (No resampling). The original labelled rows, labels, and sample weights are left unchanged. With $w_i = 1$, this corresponds to unweighted supervised learning without resampling (the *No resampling* control).

Weighting operators (Class weighting). The labelled observations remain unchanged while their sample weights are modified: $(\widetilde{\Phi}, \widetilde{y}) = (\Phi_L, y_L)$, with class-balanced weights $\widetilde{w}$. Minority examples therefore contribute more strongly to the learning objective without creating additional observations. Our implementation uses class-balanced sample weighting: normal nodes receive weight 1 and anomalies the ratio of labelled normal nodes to labelled anomalies. Under Scarce_{20}, these weights are 1 and 4, respectively, giving total weight 80 per class.

Duplication operators. These operators rebalance the empirical distribution by replicating labelled minority observations while leaving their representations unchanged. We evaluate Random Over-Sampling (ROS).

Perturbation and interpolation operators. These operators generate synthetic observations directly in the common graph-derived representation space. ROSE produces kernel-based perturbations around labelled minority examples (Menardi & Torelli, 2014), whereas SMOTE-Φ interpolates between neighbouring minority representations (Chawla et al., 2002). ADASYN-Φ (He et al., 2008) performs a related interpolation but allocates the synthetic budget non-uniformly according to the local majority-class density around each labelled anomaly.

Generative operators. These methods learn a generative model of anomalous samples and draw new observations from it. We evaluate GODM (Liu et al., 2024).

Structural resampling operators. For a selected labelled node v, an *island view* is the representation $\phi(v; G_v)$ recomputed on a sampled connected induced subgraph G_v containing v, using the same feature construction as the canonical representation $\phi_v = \phi(v; G)$. Each view forms an additional training row with the source node's label; only its graph context changes. The subgraph is sampled by a seeded forest-fire burn (Leskovec & Faloutsos, 2006). Adaptive views use the bounded Bernoulli variant ISLANDBURN (Appendix A.2) and are sampled and encoded on a loop-free bidirected copy of G. Static views use the unbounded repository sampler and retain the original edge directions for feature aggregation.

3.5 MATCHED CONTROLS FOR $\mathcal{A}$, $\mathcal{T}$, AND $\mathcal{M}$

Most strategies above resample before training, independently of the learner. Adaptive resampling instead uses the learner to guide allocation. We organize its analysis around three choices: *allocation* $\mathcal{A}$ selects source observations, *transformation* $\mathcal{T}$ constructs training rows, and the *weight rule* $\mathcal{M}$ assigns their weights. This framework asks whether gains come from which observations are selected, how they are transformed, or how much weight they receive.

Allocation ($\mathcal{A}$). An XGBoost warm-up predictor supplies probabilities $p_i^{(\mathrm{w})} = f_{\mathrm{w}}(\phi_i)$ and residual hardness $|y_i - p_i^{(\mathrm{w})}|$. We freeze these scores after warm-up and use class-wise normalization and constrained sampling to select sources at each refresh. Freezing the signal holds allocation fixed when comparing transformations, avoiding feedback from previously generated rows.

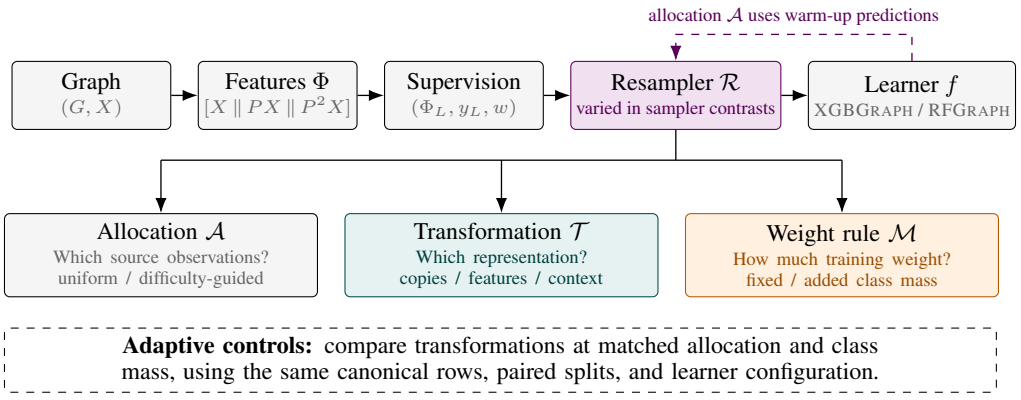
Transformation ($\mathcal{T}$). Residual-guided duplication adds unchanged copies of the selected sources. Residual-guided SMOTE-Φ instead generates $\widetilde{\phi} = \phi_i + \lambda(\phi_j - \phi_i)$ for a selected source i, where j is drawn uniformly from up to five nearest labelled training neighbours of the same class in canonical Φ, excluding i, and $\lambda \sim \mathcal{U}(0, 1)$. The generated row retains y_i. This interpolation applies to selected sources in either class and samples no graph context; if no same-class neighbour exists, it duplicates the source. Residual-guided structural resampling generates island views using ISLANDBURN. Comparing views with copies under the same source allocation and weight rule measures the contribution of the structural transformation. The static controls similarly compare duplication with feature perturbation.

Weight rule and class mass ($\mathcal{M}$). Class mass is the sum of observation weights within a class. Our primary rule redistributes this mass across original and generated rows:

$$\sum_{j:\widetilde{y}_j=c} \widetilde{w}_j = \sum_{i\in V_L:y_i=c} w_i, \qquad c \in \{0, 1\}.$$

Here j includes all resulting training rows. The added-mass control instead retains original weights and gives each generated row its source's full weight. Both rules use observation weights; they differ in whether augmentation increases total class mass.

The three choices define experimental controls, not independent components of every resampling method. Our adaptive comparisons share sources, resampling budgets, and boosting schedules. Validation and test nodes retain their canonical representations Φ. Implementation details are in Appendix A.3.



(a) The $\mathcal{A}/\mathcal{T}/\mathcal{M}$ lens: source selection, representation, and training weight.

Comparison	Datasets	Learner configuration	What varies	RQ	Evidence
Static samplers	17 datasets	XGBGRAPH, RFGRAPH; GADBench defaults	static $\mathcal{R}$ only	1	Table 1
XGB configuration	17 datasets	XGB; 100 / 200 trees	configuration and resampling	1	Table 1
Variance decomposition	8 organic	XGBGRAPH, RFGRAPH	split × strategy	1	Table 3
End-to-end incl. GNNs	17 datasets	different learners	everything (descriptive)	1	Table 6
$\mathcal{A}/\mathcal{T}/\mathcal{M}$ decomposition	17 datasets	XGB; 200 trees; frozen residuals & sources	one mechanism per contrast	2	Table 2
Label budget	6-dataset subset	XGBGRAPH, RFGRAPH	$k \in \{5, 10, 20, 50\}$	3	Fig. 2
Full supervision	17 datasets	XGBGRAPH, RFGRAPH	official masks	3	Tables 14, 15

(b) Which experiment answers which question. The six-dataset subset contains three organic and three injected datasets. Resampling gains use a reference with the same learner configuration; end-to-end comparisons are descriptive.

Figure 1: **Experimental map.** (a) Allocation, transformation, and weight rule organize the mechanism controls; (b) the controlled comparisons, research questions, and supporting evidence.

4 Main Results

We evaluate eight organic and nine injected datasets. Following GADBench, our main setting, Scarce_{20}, uses 20 labelled anomalies and 80 labelled normal nodes. The budget study varies anomalies over $k \in \{5, 10, 20, 50\}$ while retaining 80 normals, reaching a ratio of 5:80 (1:16). FullSup uses the official training masks (Appendix B.2). RQ1 establishes the provenance and split-variation findings; RQ2 uses $\mathcal{A}/\mathcal{T}/\mathcal{M}$ to examine the adaptive mechanisms. RQ3 complements these findings by varying the supervision budget.

Evaluation protocol. Each resampling contrast uses ten paired labelled splits per dataset: candidate and reference share training labels, evaluation nodes, and learner settings. We report test AUPRC in points ($\times 100$). Gains are computed within each split, averaged within each dataset, then averaged equally across datasets of the same provenance. Thus, larger graphs do not dominate the estimate. The 95% confidence intervals use a hierarchical bootstrap that resamples datasets, then paired splits within them. We compare against both unweighted training and class weighting, since beating one reference need not imply beating the other. Intervals spanning zero leave the direction of a gain unresolved. Appendix B.3 details the calculations.

4.1 RQ1: When does graph resampling improve supervised anomaly detection?

Table 1 compares resampling strategies with XGBGraph and RFGraph under Scarce_{20}, relative to class weighting. Each contrast holds the learner configuration and ten paired labelled splits fixed. Table 16 also reports paired gains and confidence intervals against unweighted training.

Static graph resampling shows no consistent improvement on organic anomalies. ROSE gains $+0.94$ AUPRC points with XGBGraph and $+0.99$ with RFGraph against class weighting. Against unweighted training, the gains are $+1.10$ and -0.25, respectively. All four 95% intervals include zero. The RFGraph comparison illustrates why both references matter: class weighting scores 41.52, below unweighted training at 42.76.

Table 1: **Resampling effects by anomaly provenance (Scarce_{20}).** Test AUPRC and gain Δ over class weighting with the same learner configuration (both $\times 100$). Scores average seeds, then datasets. Parentheses count improved datasets; bold gains have hierarchical-bootstrap 95% intervals excluding zero.

Method	XGBGraph				RFGraph			
	Organic (8)		Injected (9)		Organic (8)		Injected (9)	
	AUPRC	Δ (wins)	AUPRC	Δ (wins)	AUPRC	Δ (wins)	AUPRC	Δ (wins)
Class weighting	41.94	—	20.74	—	41.52	—	25.84	—
Static graph resampling								
No resampling	41.78	−0.16 (3)	21.54	+0.80 (6)	42.76	+1.24 (4)	26.86	+1.02 (6)
Random oversampling (ROS)	41.92	−0.02 (5)	20.25	−0.49 (3)	41.74	+0.23 (5)	25.90	+0.06 (5)
Kernel perturbation (ROSE)	42.88	+0.94 (5)	25.96	**+5.22** (8)	42.51	+0.99 (4)	30.24	**+4.40** (8)
Feature interpolation (SMOTE-Φ)	42.10	+0.16 (4)	24.18	**+3.44** (8)	42.04	+0.52 (4)	27.60	**+1.76** (7)
GraphSMOTE-inspired	40.18	−1.76 (4)	23.10	**+2.36** (7)	39.45	−2.07 (3)	25.62	−0.21 (4)
Density-adaptive interpolation (ADASYN-Φ)	41.38	−0.55 (5)	23.99	**+3.25** (9)	41.83	+0.31 (3)	27.05	+1.21 (5)
Island-view resampling	41.36	−0.58 (4)	19.26	−1.48 (1)	40.77	−0.75 (2)	22.91	−2.93 (4)
Graph generation (GODM)†	40.47	−1.47 (2)	23.28	**+2.58** (7)	40.71	−0.81 (1)	27.45	+1.64 (7)
Gradient sampling (100-tree refit after a 30-tree pilot)								
GOSS-inspired	39.77	**−2.17** (0)	16.86	**−3.88** (0)		—		
Adaptive graph resampling								
Residual-guided SMOTE-Φ	41.83	−0.11 (4)	21.74	**+1.00** (9)		—		
Alternative XGBGraph configuration: 200 trees, $\eta = 0.05$								
Class weighting (200 trees)	42.07	—	21.07	—		—		
No resampling	42.26	+0.19 (4)	22.00	**+0.93** (7)		—		
Static ROSE (control)	43.39	+1.32 (5)	26.59	**+5.53** (8)		—		
Residual-guided SMOTE-Φ	42.54	**+0.46** (6)	22.40	**+1.33** (9)		—		

XGBGraph uses 100 trees, $\eta = 0.3$, and `gpu_hist`, except in the final block (200 trees, $\eta = 0.05$, `hist`); both use depth 6. All methods share canonical Φ and ten paired splits; adaptive gates are disabled. Full settings: Appendix A.1. GraphSMOTE-inspired is a local interpolation adaptation, not the published GraphSMOTE model. GOSS-inspired adds a separate 30-tree pilot to its 100-tree fit: its total training budget is not matched to the reference. Neither row evaluates the corresponding original method. †GODM: 169/170 trials per learner; failed Flickr seed 2 is excluded from both sides of its contrasts. Dashes in the RFGraph columns denote configurations not evaluated.

Simple perturbation provides strong gains on injected anomalies. ROSE delivers the largest mean gains among the evaluated static methods: $+5.22$ points with XGBGraph and $+4.40$ with RFGraph, improving eight of nine injected datasets with each learner. Both 95% intervals exclude zero. Against unweighted training, the gains are $+4.42$ and $+3.38$ points; both intervals also exclude zero.

Residual-guided interpolation provides a small gain over class weighting. With the 200-tree configuration, residual-guided SMOTE-Φ gains $+0.46$ points over class weighting (95% interval $[+0.06, +0.86]$). Against unweighted training, its gain is $+0.27$ ($[-0.30, +0.95]$). This modest benefit does not establish superiority over static resampling: ROSE reaches 43.39 AUPRC, against 42.54 for residual-guided SMOTE-Φ.

Sensitivity to the XGBoost configuration. The estimated organic gain of residual-guided SMOTE-Φ changes from -0.11 to $+0.46$ points against each configuration's class-weighted reference. Four settings change together: 100 to 200 trees, $\eta = 0.3$ to 0.05, `gpu_hist` to `hist`, and 25 warm-up trees plus five 15-tree blocks to 50 plus five 30-tree blocks. Depth remains six. This is a sensitivity check of the whole configuration. It neither isolates one setting's effect nor establishes a reversal from a negative to a positive effect, since the first estimate's interval includes zero.

Labelled splits explain more variation than strategy choice. For each organic dataset and learner, we decompose AUPRC variation across ten paired splits and five strategies into split, strategy, and a remainder that combines interaction and noise. Averaging these shares across datasets gives 72% for splits versus 10% for strategies with XGBGraph, and 68% versus 16% with RFGraph (Table 3). Weibo is the exception: strategy variation exceeds split variation for both learners. Excluding Weibo raises the mean split share to 78% and 74%, respectively; we retain it in the main estimate. These are descriptive shares for the evaluated strategies, not causal effects or population variance estimates. Including unweighted training as a sixth strategy preserves the pattern (72% versus 9% for XGBGraph; 65% versus 18% for RFGraph). Appendix B.3 gives the calculation and per-dataset results.

4.2 RQ2: How beneficial is each mechanism in graph resampling?

We now use $\mathcal{A}$, $\mathcal{T}$, and $\mathcal{M}$ to interpret resampling gains within a fixed learner configuration. Table 2 compares weighting, duplication, transformation, and added class mass in the static and adaptive families. These controls were run after the sealed evaluation was opened and are diagnostic (Appendix B.2).

Residual allocation and weighting provide a small gain. With 200-tree XGBoost, residual weights improve AUPRC over class weighting by $+0.20$ points on organic datasets (95% CI $[+0.02, +0.41]$) and $+0.33$ on injected datasets ($[+0.05, +0.59]$). Physical copies with the same source weights give practically identical results (Appendix A.3).

Structural transformation adds no resolved benefit. Holding allocation and the weight rule fixed, replacing residual-guided copies with structural views adds $+0.28$ organic AUPRC points ($[-0.18, +0.94]$) and -0.32 injected points ($[-1.40, +0.59]$). Both intervals include zero. This comparison isolates what structural views add beyond selecting and reweighting their source observations.

Added class mass also has no resolved benefit for adaptive views. The static ROSE control raises positive mass from 80 to 320 while retaining normal mass 80, on identical generated rows. The adaptive control instead compares redistributing 10% of each class's mass to generated views with adding their full source weights. Within each family, allocation, generated rows, splits, and learner settings are fixed. For example, a class with total weight 80 retains 72 on original rows and transfers 8 to generated views: its total remains 80. For adaptive views, added-minus-fixed mass gives $+0.02$ points on organic datasets ($[-0.29, +0.40]$) and -0.09 on injected datasets ($[-0.68, +0.50]$). Together, these controls motivate testing new transformations against duplication with matched sources and class mass.

Table 2: **Resampling mechanisms (Scarce_{20}).** Test AUPRC and gain Δ over class weighting within each family and learner (both $\times 100$). Scores average ten paired seeds, then datasets; parentheses count improved datasets.

Mechanism	Variant	XGBGraph				RFGraph			
		Organic (8)		Injected (9)		Organic (8)		Injected (9)	
		AUPRC	Δ (wins)	AUPRC	Δ (wins)	AUPRC	Δ (wins)	AUPRC	Δ (wins)
Static: ROSE controls									
Control	No resampling	41.78	−0.16 (3)	21.54	+0.80 (6)	42.76	+1.24 (4)	26.86	+1.02 (6)
Row weights	Class weighting	41.94	—	20.74	—	41.52	—	25.84	—
	Matched source weights	41.92	−0.02 (5)	20.25	−0.49 (3)	41.57	+0.06 (3)	25.97	+0.13 (5)
Duplication	Copies (ROS)	41.92	−0.02 (5)	20.25	−0.49 (3)	41.74	+0.23 (5)	25.90	+0.06 (5)
Transformation	Fixed perturbation	42.30	+0.36 (5)	25.19	+4.45 (8)	42.59	+1.07 (4)	29.42	+3.58 (8)
	Independent noise (ROSE)	42.88	+0.94 (5)	25.96	+5.22 (8)	42.51	+0.99 (4)	30.24	+4.40 (8)
Mass $\mathcal{M}$	ROSE, added mass	42.50	+0.57 (5)	25.80	+5.06 (8)	41.16	−0.35 (3)	28.37	+2.53 (5)
Adaptive: residual-guided controls									
Control	No resampling	42.26	+0.19 (4)	22.00	+0.93 (7)		—		
Row weights	Class weighting	42.07	—	21.07	—		—		
	Residual weights	42.27	+0.20 (6)	21.39	+0.33 (7)		—		
Duplication	Residual copies	42.27	+0.20 (6)	21.39	+0.33 (7)		—		
Transformation	Structural views, fixed mass	42.55	+0.48 (4)	21.08	+0.01 (5)		—		
Mass $\mathcal{M}$	Structural views, added mass	42.57	+0.50 (4)	20.99	−0.08 (5)		—		

Static XGB: 100 trees, $\eta = 0.3$; adaptive XGB: 200 trees, $\eta = 0.05$, with residuals frozen after warm-up. Mechanisms are compared within each block; Δ is relative to class weighting, not the preceding row. *No resampling* uses unit weights; corrected variants preserve mass 80 per class except the added-mass controls described in Section 4.2. Fixed perturbation repeats one perturbed row per source; ROSE draws independent noise. Win counts are descriptive. Adaptive added-minus-fixed mass: $+0.02$ $[-0.29, +0.40]$ organic; -0.09 $[-0.68, +0.50]$ injected (hierarchical-bootstrap 95% intervals). Adaptive RF was not evaluated.

4.3 RQ3: How do graph resampling gains vary with the supervision budget?

We vary the number of labelled anomalies over $k \in \{5, 10, 20, 50\}$ while keeping 80 labelled normal nodes, on a panel of three organic and three injected datasets. This varies both the amount of positive

Table 3: **Sources of organic AUPRC variation.** Mean within-dataset shares (%) across ten splits and five strategies at Scarce_{20}. The remainder combines interaction and noise.

Panel	Learner	Split	Strategy	Remainder
All eight datasets	XGBGraph	72	10	18
	RFGraph	68	16	16
Weibo only	XGBGraph	29	35	36
	RFGraph	27	47	26

supervision and the training imbalance ratio. Figure 2 shows ROSE gains across these budgets. FullSup appears in a separate panel because it uses different test masks.

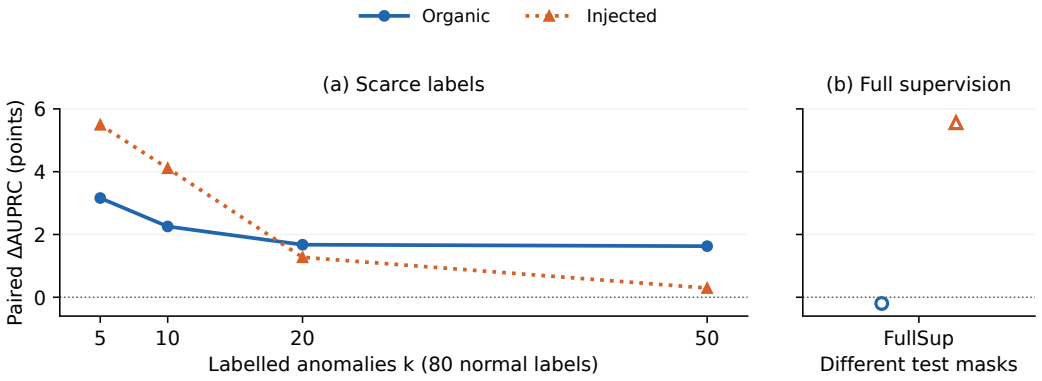


Figure 2: **Resampling gains across supervision budgets.** Paired ΔAUPRC ($\times 100$) on Weibo, FDCompCN and T-Finance (organic) and Cora, Flickr and Coauthor-CS (injected). Points average ten paired seeds within datasets, then the three datasets per provenance. ROSE equally averages XGBGraph and RFGraph gains against their respective class-weighted references. Panel (a) varies the scarce-label budget; panel (b) reports FullSup on the same six datasets using its own test masks. Both panels share the vertical scale and show descriptive means.

Some gains are concentrated at the smallest label budgets. For ROSE, the mean organic gain across the two learners falls from $+3.16$ AUPRC points at $k = 5$ to $+1.67$ at $k = 20$ and $+1.63$ at $k = 50$. The corresponding injected gains are $+5.49$, $+1.27$, and $+0.29$. These descriptive patterns apply to the selected panel and methods (Table 7); their generality remains to be tested.

Averaging learners hides different budget responses. Across all six datasets, ROSE gains over class weighting are $+5.27$ points for XGBGraph and $+3.38$ for RFGraph at $k = 5$. At $k = 50$, they are $+2.46$ and -0.54, respectively (Table 7). These averages combine both provenances, unlike the separate curves in Figure 2. XGBGraph's gain also rises from $+1.93$ at $k = 20$ to $+2.46$ at $k = 50$: the response is not uniformly decreasing. Against unweighted training, the $k = 50$ gains are $+2.19$ and -0.86. These descriptive contrasts support reporting budget curves for each learner and reference, rather than interpreting the cross-learner mean as a common response to additional labels.

Full supervision remains a separate evaluation setting. On the same six-dataset panel, FullSup ROSE gains average -0.20 points on organic datasets and $+5.56$ on injected datasets across the two learners, showing persistent gains on injected benchmarks. Comparing FullSup with Scarce_k combines changes in label counts and training and test masks (Tables 14 and 15).

Limits and interpretation. Our controlled resampling comparisons use two tree ensembles with a common canonical representation Φ. GNNs serve as contextual references (Table 6), but are not included in these controlled comparisons; the findings therefore do not establish how resampling interacts with jointly learned representations. We omit GraphENS and TAM because their published evaluations couple ego-network synthesis or topology-dependent margins with GNN training; adapting them to trees would require separate implementation and validation. ReNode is model-agnostic: its topology-based weights could be adapted to our learners, but this comparison remains unevalu-

ated. We therefore make no claim of superiority over these methods (Park et al., 2022; Song et al., 2022; Chen et al., 2021).

The organic bootstrap resamples only eight datasets; its intervals have limited precision at the dataset level and may be sensitive to panel composition. The organic and injected panels also use different graphs, so their contrast does not isolate a causal effect of anomaly provenance. BOND (Liu et al., 2022b) already highlighted the limited coverage of evaluations based on a few synthetic anomaly types. Here, contextual injection replaces attributes with those of a distant node, while structural injection creates cliques (Appendix B.1). One hypothesis is that ROSE's feature perturbations suit the geometry of these injected anomalies. Testing this explanation requires separating contextual and structural anomalies and varying the injection process. Our budget study covers only three datasets per provenance and changes positive supervision and training imbalance together.

Recommendations for future benchmarks. We encourage the community to make four comparisons standard.

First, establish the simplest competitive reference: report both unweighted and class-weighted training, and include inexpensive duplication and perturbation before attributing gains to a more complex generator. ROSE's strong injected results make it a useful control, while its uncertain organic gains rule out a general recommendation to resample.

Second, test the proposed mechanism: match source selection, class mass, and learner configuration when comparing new representations with copies. Report the additional training cost separately. A gain over a weak reference does not identify which component helped.

Third, expose variation: retain paired labelled splits, report dataset-level results, and separate organic from injected benchmarks. An interval containing zero warrants more precise evaluation, not a claim that two methods are equivalent.

Fourth, evaluate across label budgets and learners: keep evaluation nodes fixed across scarce-label budgets and report each learner's response. To isolate the effect of acquiring labels, vary total supervision and class ratio separately. Extending these controls to GNN training and topology-based reweighting would test whether the present findings generalize beyond fixed canonical representations and tree ensembles.

5 Conclusion

This empirical study of resampling for supervised graph anomaly detection finds the clearest gains on injected anomalies, while organic gains remain uncertain and depend on the reference. Under scarce labels, labelled-split variation exceeds strategy variation on average in the organic panel. The $\mathcal{A}/\mathcal{T}/\mathcal{M}$ controls distinguish source allocation and weighting from new representations: residual weighting provides a small gain, while the additional contributions of structural views and added mass remain unresolved.

These findings support comparing resampling with both unweighted training and class weighting, reporting organic and injected results separately, and evaluating methods on repeated paired labelled splits. New transformations should also be compared with duplication under matched source allocation and class mass. The six-dataset budget study suggests larger perturbation gains at the smallest budget, motivating further work on how label acquisition and resampling interact.

REFERENCES

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

Chao Chen, Andy Liaw, and Leo Breiman. Using random forest to learn imbalanced data. Technical Report 666, Department of Statistics, University of California, Berkeley, 2004.

Deli Chen, Yankai Lin, Guangxiang Zhao, Xuancheng Ren, Peng Li, Jie Zhou, and Xu Sun. Topology-imbalance learning for semi-supervised node classification. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/fa7cdfad1a5aaf8370ebeda47a1ff1c3-Abstract.html.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

Kaize Ding, Jundong Li, Rohit Bhanushali, and Huan Liu. Deep anomaly detection on attributed networks. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*, 2019.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive Representation Learning on Large Graphs. In *NIPS*, pp. 1024–1034, 2017.

Haibo He, Yang Bai, Edwardo A. Garcia, and Shutao Li. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *IEEE International Joint Conference on Neural Networks (IJCNN)*, 2008.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

Jure Leskovec and Christos Faloutsos. Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pp. 631–636, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933395. doi: 10.1145/1150402.1150479. URL https://doi.org/10.1145/1150402.1150479.

Wen-Zhi Li, Chang-Dong Wang, Hui Xiong, and Jian-Huang Lai. GraphSHA: Synthesizing harder samples for class-imbalanced node classification. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2023.

Fanzhen Liu, Xiaoxiao Ma, Jia Wu, Jian Yang, Shan Xue, Amin Beheshti, Chuan Zhou, Hao Peng, Quan Z. Sheng, and Charu C. Aggarwal. DAGAD: Data augmentation for graph anomaly detection. In *IEEE International Conference on Data Mining (ICDM)*, 2022a.

Kay Liu, Yingtong Dou, Yue Zhao, Xueying Ding, Xiyang Hu, Ruitong Zhang, Kaize Ding, Canyu Chen, Hao Peng, Kai Shu, Lichao Sun, Jundong Li, George H. Chen, Zhihao Jia, and Philip S. Yu. BOND: Benchmarking unsupervised outlier node detection on static attributed graphs. In *Advances in Neural Information Processing Systems*, volume 35, 2022b. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/acc1ec4a9c780006c9aafd595104816b-Abstract.html.

Kay Liu, Hengrui Zhang, Ziqing Hu, Fangxin Wang, and Philip S. Yu. Data augmentation for supervised graph outlier detection with latent diffusion models. In *Proceedings of the Third Learning on Graphs Conference (LoG)*, volume 269 of *Proceedings of Machine Learning Research*, 2024. URL https://proceedings.mlr.press/v269/liu25b.html.

Xiaodong Liu et al. LightGAD: Lightweight graph anomaly detection with adaptive homophily-aware aggregation and gradient boosting. *Knowledge-Based Systems*, 2026. doi: 10.1016/j.knosys.2026.115126. Code: https://github.com/lxdgogogo/LIGHTGAD.

Yixin Liu, Zhao Li, Shirui Pan, Chen Gong, Chuan Zhou, and George Karypis. Anomaly detection on attributed networks via contrastive self-supervised learning. *IEEE Transactions on Neural Networks and Learning Systems*, 33(6):2378–2392, 2022c.

Giovanna Menardi and Nicola Torelli. Training and assessing classification rules with imbalanced data. *Data Mining and Knowledge Discovery*, 28(1):92–122, 2014. doi: 10.1007/s10618-012-0295-5.

Lin Meng, Hesham Mostafa, Marcel Nassar, Xiaonan Zhang, and Jiawei Zhang. Generative graph augmentation for minority class in fraud detection. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM)*, pp. 4200–4204, 2023.

Joonhyung Park, Jaeyun Song, and Eunho Yang. GraphENS: Neighbor-aware ego network synthesis for class-imbalanced node classification. In *International Conference on Learning Representations (ICLR)*, 2022.

Jaeyun Song, Joonhyung Park, and Eunho Yang. TAM: Topology-aware margin loss for class-imbalanced node classification. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 20369–20383, 2022. URL https://proceedings.mlr.press/v162/song22a.html.

Jianheng Tang, Jiajin Li, Ziqi Gao, and Jia Li. Rethinking graph neural networks for anomaly detection. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, 2022.

Jianheng Tang, Fengrui Hua, Ziqi Gao, Peilin Zhao, and Jia Li. Gadbench: Revisiting and benchmarking supervised graph anomaly detection, 2023. URL https://arxiv.org/abs/2306.12251.

Taraneh Younesian, Daniel Daza, Emile van Krieken, Thiviyan Thanapalasingam, and Peter Bloem. Grapes: Learning to sample graphs for scalable graph neural networks, 2025. URL https://arxiv.org/abs/2310.03399.

Hanqing Zeng, Muhan Zhang, Yinglong Xia, Ajitesh Srivastava, Andrey Malevich, Rajgopal Kannan, Viktor Prasanna, Long Jin, and Ren Chen. Decoupling the depth and scope of graph neural networks. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=d0MtHWY0NZ.

Tianxiang Zhao, Xiang Zhang, and Suhang Wang. GraphSMOTE: Imbalanced node classification on graphs with graph neural networks. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining (WSDM)*, 2021.

A Implementation Details

A.1 Detector configurations

The two XGBGraph configurations in Table 1 use depth six and row and column subsampling ratios of one:

Trees	Learning rate	Tree method	Warm-up trees	Refresh blocks
100	0.3	gpu_hist	25	5×15 trees
200	0.05	hist	50	5×30 trees

The refresh schedules apply to residual-guided resampling; the mechanism controls use the 200-tree configuration. Representation and context gates are disabled. A 170-fit check of blockwise training

with 100 trees reproduces the static reference predictions exactly. The 200-tree unit-weight reference uses the same block schedule without generating rows; its static ROSE comparison includes all 90 injected trials. Each contrast uses its own configuration's reference, so the sensitivity comparison does not isolate any single hyperparameter.

A.2 Structural views

Static island views select labelled anomalies uniformly and use the unbounded sampler described in Section 3.4. The bounded residual-guided sampler uses at most 20 nodes, acceptance probability 0.6, depth cap two, and fan-out four. The mechanism controls use connected re-ignition and eligible-first expansion: an exhausted frontier restarts from a sampled node with unexplored neighbours, and expansion draws from unvisited neighbours.

A.3 Residual-guided mechanism controls

Warm-up hardness scores (Section 3.5) are winsorized at the 5th and 95th percentiles within each class, then rescaled to $[0, 1]$. The generation budget is $\mathrm{round}(0.5|V_L|)$, distributed across refreshes with class-proportional quotas, exploration share 0.3, anomaly exploitation share 0.5, a one-refresh cooldown, and at most two uses per source. Residual-guided SMOTE-Φ draws neighbours only from the original labelled rows, with pools fixed across refreshes. The fixed-mass rule transfers 10% of each represented class's weight from original to generated rows in proportion to source weights; a class with no generated rows retains its original weights. Direct reweighting and physical duplication differ by at most 3.1×10^{-5} AUPRC points over 170 paired trials at the default setting.

A.4 Other operators and contextual references

ADASYN-Φ uses `imbalanced-learn`, five neighbours and class parity, as does SMOTE-Φ. Per-anchor rounding requires trimming or repeating generated rows to match the 60-row ROS budget in 290/340 Scarce_{20} trials; the reported mean deviation before adjustment is 0.2 rows. GODM uses the published implementation and default hyperparameters, with the same generated rows for both learners and the same positive-row budget as ROS. A missing pre-trained checkpoint prevents evaluation of Flickr seed 2: paired GODM contrasts exclude it from both methods, leaving 169 trials per learner.

GraphSMOTE-inspired uses topology-anchored interpolation, five generated rows per labelled anomaly, forest-fire neighbours, and rebalanced class weights. GOSS-inspired uses gradient-based normal-node sampling ($a = 0.2, b = 0.1$): a discarded 30-tree pilot selects observations for an independent 100-tree fit. These are local adaptations, not reproductions of GraphSMOTE or LightGBM GOSS; the latter also exceeds its reference's training budget by the pilot cost.

The contextual references in Table 6 include Balanced Random Forest (Chen et al., 2004), GraphSAGE (Hamilton et al., 2017), BWGNN (Tang et al., 2022), and LightGAD+ (Liu et al., 2026). LightGAD+ uses its official implementation and recommended settings where available; other datasets are configuration-transfer evaluations. Weighting conventions and the label-free Isolation Forest reference are specified with that table.

Computational cost and execution audit. For n_{aug} generated rows of dimension d_Φ, ROS adds $O(n_{\mathrm{aug}})$ work and ROSE $O(n_{\mathrm{aug}} d_\Phi)$. Neighbour search costs $O(n_+^2 d_\Phi)$ for SMOTE-Φ and $O(n_+ n_L d_\Phi)$ for ADASYN-Φ, where n_+ and n_L count labelled anomalies and all labelled nodes. Island views add one burn and local feature computation per row; residual guidance also requires refreshes. GODM additionally trains a VGAE and diffusion model offline. At Scarce_{20}, mean generation times are 36.3 ms for ROS, 602.9 ms for ROSE, 39.3 ms for SMOTE-Φ, 113.7 ms for ADASYN-Φ, and 35.5 ms for static island views. GODM's 3.3 ms measures sampling only, excluding offline training. ADASYN training times were measured under a different machine load and are not compared. In the original campaign, SMOTE reduced its neighbourhood at $k = 5$; ADASYN required no such fallback at $k = 20$. Three external LightGAD crashes and the missing GODM checkpoint were recorded without reruns. The full historical timing audit, including aggregates over multiple label budgets, is retained in the released archive.

B Experiment Details

B.1 Datasets

Table 4 separates eight datasets with observed anomaly labels from nine injected benchmarks. Injected anomalies follow the DOMINANT/CoLA recipe (Ding et al., 2019; Liu et al., 2022c): half are structural clique anomalies and half are contextual anomalies obtained by replacing attributes with those of a distant feature-space node. The same recipe applies to the six legacy and three additional graph supports.

Table 4: Statistics of the 17-dataset panel, recomputed from the loaded DGL graphs. *Edges* are the directed edges of the loaded graph, self-loops removed; *Avg. deg.* is the mean out-degree; *Homoph.* is edge-label homophily (fraction of edges whose endpoints share a label).

Dataset	Nodes	Edges	Feat.	Anom. %	Avg. deg.	Homoph.
Organic anomalies (real labels)						
Amazon	11,944	8,835,152	25	6.9	739.7	0.95
FDCompCN	5,317	7,268	57	10.5	1.4	0.85
Questions	48,921	153,540	301	3.0	3.1	0.84
Reddit	10,984	157,032	64	3.3	14.3	0.95
T-Finance	39,357	42,445,086	10	4.6	1078.5	0.97
Tolokers	11,758	519,000	10	21.8	44.1	0.59
Weibo	8,405	407,963	400	10.3	48.5	0.98
YelpChi	45,954	7,693,958	32	14.5	167.4	0.77
Injected anomalies, legacy supports						
ACM	16,484	148,146	8,337	3.6	9.0	0.93
BlogCatalog	5,196	345,566	8,189	5.8	66.5	0.89
Citeseer	3,327	10,154	3,703	4.5	3.1	0.93
Cora	2,708	11,606	1,433	5.5	4.3	0.91
Flickr	7,575	482,608	12,047	5.9	63.7	0.90
Pubmed	19,717	92,846	500	3.0	4.7	0.94
Injected anomalies, additional benchmark graphs						
Amazon-Computers	13,752	496,332	767	5.0	36.1	0.91
Amazon-Photo	7,650	240,672	745	5.0	31.5	0.91
Coauthor-CS	18,333	170,084	6,805	5.0	9.3	0.90

B.2 Evaluation Protocol and Reproducibility

Evaluation is transductive: methods access the complete graph and attributes, but only training labels for fitting. Methods share ten paired splits, with $\texttt{seed_value} = 3407 + 10 \cdot \text{trial}$ for trial $\in \{0, \ldots, 9\}$. Within each resampling contrast, canonical features and learner settings are fixed (Figure 1).

Label budgets and evaluation nodes. Scarce_{20} uses twenty labelled anomalies and eighty labelled normal nodes, with a disjoint validation set of the same class proportions. The budget study varies $k \in \{5, 10, 20, 50\}$ anomalies with eighty normals on Cora, Flickr, Coauthor-CS, Weibo, FDCompCN, and T-Finance. Training budgets are nested; validation and test nodes are reserved for the largest budget and fixed across k. Unused training anomalies never return to evaluation. Thus both positive supervision and training imbalance vary. FullSup instead uses official GADBench masks and different test nodes. For Amazon and YelpChi, the official mask columns repeat one fixed split: deterministic XGBoost without resampling therefore has zero seed variation, whereas stochastic resamplers can vary on those same masks.

Selection and diagnostic status. The original configurations were selected on validation data; candidates, comparisons, analysis and decision rules were frozen before opening the sealed test results, without external preregistration. Baseline results were reused without further model development. The reported residual-guided reruns and mechanism controls were performed after unsealing, using cached splits, and are diagnostic rather than independent confirmation.

Reproduction. The environment is Python 3.10.20, numpy 2.2.6, scikit-learn 1.7.2, xgboost 1.7.6, and imbalanced-learn 0.14.2. The released repository contains configurations, run manifests and results. Per-trial records identify the dataset, split, label budget, anomaly provenance, strategy, fallback, and validation and test metrics.

B.3 Metrics and Statistical Analysis

AUPRC is the primary metric; AUROC and Recall@K (K is the test-set anomaly count) are complementary discrimination measures, and the Brier score is a calibration diagnostic. The paired difference is $\Delta = \mathrm{Metric}_{\mathrm{candidate}} - \mathrm{Metric}_{\mathrm{reference}}$. We average seeds within each dataset, then datasets equally within the stated panel, and count datasets with positive mean differences as wins. AUPRC scores and differences are multiplied by 100. The hierarchical bootstrap uses 20,000 replicates, resampling datasets and then paired seeds; 95% intervals quantify estimation uncertainty rather than equivalence.

Descriptive variance decomposition. For each organic dataset and learner, let x_{ij} denote Scarce_{20} test AUPRC for split $i \in \{1, \ldots, 10\}$ and strategy $j \in \{1, \ldots, 5\}$: class weighting, ROS, ROSE, SMOTE-Φ, and uniform island views. The complete, unreplicated 10×5 layout gives

$$\underbrace{\textstyle\sum_{ij}(x_{ij} - \bar{x})^2}_{SS_{\mathrm{tot}}} = \underbrace{5\textstyle\sum_i(\bar{x}_{i\cdot} - \bar{x})^2}_{SS_{\mathrm{split}}} + \underbrace{10\textstyle\sum_j(\bar{x}_{\cdot j} - \bar{x})^2}_{SS_{\mathrm{strategy}}} + SS_{\mathrm{resid}}.$$

The residual pools interaction and noise, which cannot be separated with one observation per cell. Shares $SS_\bullet/SS_{\mathrm{tot}}$ are computed within each dataset and averaged equally across the eight organic datasets, not from a pooled cross-dataset layout (Table 5). This is a descriptive decomposition, not an inferential test. Dropping FDCompCN gives split/strategy shares of $69/11\%$ for XGBGraph and $66/18\%$ for RFGraph; using AUROC gives $73/8\%$ and $66/18\%$. Adding unweighted training as a sixth strategy gives $72/9\%$ and $65/18\%$. Split selection remains the larger source in all three variants.

Table 5: **Per-dataset descriptive variance decomposition.** Within-dataset AUPRC shares (%) across ten splits and five strategies at Scarce_{20}; rows sum to 100 up to rounding. "Resid." combines interaction and noise. Mean gives equal weight to the eight datasets.

	XGBGraph			RFGraph		
Dataset	Split	Strategy	Resid.	Split	Strategy	Resid.
Amazon	84	2	13	73	4	22
FDCompCN	91	2	8	86	1	12
Questions	84	3	13	86	5	10
Reddit	75	5	20	71	17	12
T-Finance	55	13	32	73	14	12
Tolokers	75	6	18	56	21	23
Weibo	29	35	36	27	47	26
YelpChi	81	12	6	74	16	10
Mean	**72**	**10**	**18**	**68**	**16**	**16**

B.4 Additional Experimental Results

Table 6 provides contextual detector comparisons. Tables 8–9 give dataset-level AUPRC, including ADASYN and GODM; Tables 10–13 report AUROC and Recall@K for the original static panel. Tables 14–15 cover FullSup, and Table 7 covers the label-budget study.

ADASYN versus uniform interpolation. Paired ADASYN-minus-SMOTE AUPRC differences are -0.71 $[-1.83, +0.05]$ organic and -0.20 $[-1.08, +0.60]$ injected for XGBGraph, and -0.21 $[-0.86, +0.34]$ and -0.55 $[-1.90, +0.52]$ for RFGraph (hierarchical-bootstrap 95% intervals). ADASYN's mean AUPRC is below ROSE's in both strata for both learners, by 0.68–3.19 points.

Table 6: Graph-derived tree ensembles versus raw-feature trees, graph neural networks, and additional reference methods (test AUPRC $\times 100$, mean over ten paired seeds, Scarce_{20}). Best result in each row is shown in bold.

	Raw X		Graph-derived trees			Other detectors		
Dataset	XGB	RF	XGBGraph	RFGraph	Bal. RF	GraphSAGE	BWGNN	LightGAD+
Organic anomalies (real labels)								
Reddit	3.78	3.80	3.81	4.08	4.13	3.91	**4.65**	3.92
Weibo	64.69	78.58	78.87	**80.87**	77.01	57.08	76.78	60.99
Amazon	83.46	83.78	83.32	69.95	68.18	48.50	80.67	**84.98**
YelpChi	26.17	**27.37**	23.56	22.32	21.77	20.39	23.87	25.36
T-Finance	69.64	74.71	78.46	**82.29**	81.98	10.23	63.76	50.69
Tolokers	30.26	31.31	36.08	36.61	**36.63**	33.98	35.86	30.41
Questions	8.65	**10.65**	7.13	10.03	9.57	6.70	6.58	8.30
FDCompCN	26.77	**28.94**	24.27	25.99	25.44	10.90	10.91	26.57
Injected benchmarks (DOMINANT/CoLA protocol)								
Cora	4.50	6.80	42.22	46.69	**47.97**	13.57	10.62	6.54
Citeseer	4.69	6.72	42.16	**50.63**	50.33	14.57	9.31	6.07
BlogCatalog	21.36	28.70	6.69	6.90	6.09	**28.74**	28.17	13.75
Flickr	13.62	25.66	11.07	15.84	13.26	**32.34**	30.34	8.74
ACM	4.25	7.53	9.41	**16.73**	14.75	10.60	9.83	4.57
Pubmed	3.98	4.29	20.89	**34.54**	31.27	4.78	5.34	6.05
Amazon-Photo	16.96	21.57	18.61	20.12	16.69	30.66	**44.70**	14.36
Amazon-Computers	16.29	20.02	12.83	13.37	12.56	26.27	**36.36**	13.89
Coauthor-CS	15.82	28.49	22.80	27.73	21.73	25.15	**30.21**	16.88
Macro avg.	24.41	28.76	30.72	**33.22**	31.73	22.26	29.88	22.47
Organic (8)	39.18	42.39	41.94	41.52	40.59	23.96	37.89	36.40
Injected (9)	11.27	16.64	20.74	**25.84**	23.85	20.74	22.76	10.09

Isolation Forest on Φ: macro AUPRC 10.91 (organic 15.45; injected 6.87). Raw trees are unweighted; graph-derived trees use class weighting; Balanced RF and GNNs retain their own training objectives. With unit weights on both sides, replacing X by Φ adds $+2.60/+0.37$ organic points (XGBGraph/RFGraph), improving 3/8 datasets each; excluding Weibo and T-Finance gives $-0.43/-2.49$. Injected gains are $+10.3/+10.2$ points.

Table 7: **Label-budget analysis.** Paired ΔAUPRC ($\times 100$) against class-weighted and unweighted references, averaged over ten seeds and the six datasets of Appendix B.2. Parentheses count improved datasets; final rows give absolute reference AUPRC. Bold marks the largest gain per column and learner.

	Δ vs class weighting				Δ vs *No resampling*			
Method	$k=5$	$k=10$	$k=20$	$k=50$	$k=5$	$k=10$	$k=20$	$k=50$
XGBGRAPH								
No resampling	+3.43 (5)	+1.31 (5)	+0.17 (4)	+0.28 (4)	—	—	—	—
Class weighting	—	—	—	—	−3.43 (1)	−1.31 (1)	−0.17 (2)	−0.28 (2)
ROS	−0.10 (2)	−0.18 (3)	−0.58 (2)	+0.24 (5)	−3.53 (1)	−1.49 (1)	−0.76 (2)	−0.03 (3)
ROSE	**+5.27** (5)	**+3.89** (5)	**+1.93** (5)	**+2.46** (4)	**+1.84** (5)	**+2.57** (5)	**+1.76** (4)	**+2.19** (6)
SMOTE-Φ	+1.26 (5)	+1.79 (4)	+0.81 (3)	+1.22 (4)	−2.17 (4)	+0.48 (2)	+0.64 (3)	+0.95 (6)
Island view	+1.95 (6)	+0.06 (4)	−1.74 (2)	−0.34 (4)	−1.48 (3)	−1.25 (4)	−1.91 (1)	−0.61 (3)
Reference (absolute)	28.29	36.96	42.95	48.35	31.72	38.27	43.12	48.63
RFGRAPH								
No resampling	+1.76 (5)	+1.43 (4)	**+1.19** (3)	**+0.32** (4)	—	—	—	—
Class weighting	—	—	—	—	−1.76 (1)	−1.43 (2)	−1.19 (3)	−0.32 (2)
ROS	−0.60 (4)	+0.40 (4)	+0.35 (3)	−0.08 (3)	−2.36 (1)	−1.03 (3)	−0.84 (2)	−0.41 (2)
ROSE	**+3.38** (5)	**+2.48** (4)	+1.01 (4)	−0.54 (3)	**+1.62** (4)	**+1.05** (4)	−0.18 (2)	−0.86 (2)
SMOTE-Φ	+1.21 (3)	+1.01 (4)	+1.06 (3)	+0.20 (3)	−0.55 (2)	−0.42 (3)	**−0.13** (2)	**−0.12** (2)
Island view	−2.14 (3)	−2.88 (2)	−2.91 (1)	−1.58 (3)	−3.90 (0)	−4.30 (1)	−4.10 (0)	−1.91 (2)
Reference (absolute)	36.16	41.88	46.57	52.31	37.92	43.30	47.76	52.63

Table 8: **Absolute AUPRC of graph resampling with XGBGRAPH** (Scarce_{20}, test AUPRC $\times 100$). Entries are means with standard deviations in superscript over ten seeds; GODM uses nine on Flickr. NR: no resampling; CW: class weighting. Bold marks the largest mean in each row. Paired GODM contrasts exclude Flickr seed 2 from both methods (Table 16); other columns retain all ten seeds.

Dataset	NR	CW	ROS	ROSE	SMOTE-Φ	ADASYN-Φ	Island	GODM
Organic anomalies (real labels)								
Reddit	$3.73^{\pm 0.28}$	$3.81^{\pm 0.30}$	$3.87^{\pm 0.39}$	$3.75^{\pm 0.47}$	$4.01^{\pm 0.49}$	$\mathbf{4.02}^{\pm 0.38}$	$3.81^{\pm 0.33}$	$3.67^{\pm 0.43}$
Weibo	$79.00^{\pm 5.54}$	$78.87^{\pm 5.39}$	$79.48^{\pm 4.37}$	$\mathbf{82.49}^{\pm 2.35}$	$80.30^{\pm 3.71}$	$79.14^{\pm 3.63}$	$72.59^{\pm 6.62}$	$80.62^{\pm 3.20}$
Amazon	$81.85^{\pm 5.37}$	$83.32^{\pm 2.72}$	$83.33^{\pm 2.72}$	$82.67^{\pm 2.67}$	$\mathbf{83.93}^{\pm 3.68}$	$83.35^{\pm 2.48}$	$82.73^{\pm 4.23}$	$74.17^{\pm 9.15}$
YelpChi	$23.26^{\pm 3.78}$	$23.56^{\pm 4.07}$	$23.23^{\pm 4.24}$	$\mathbf{27.70}^{\pm 4.56}$	$25.83^{\pm 5.20}$	$25.65^{\pm 4.91}$	$25.20^{\pm 4.63}$	$26.17^{\pm 3.21}$
T-Finance	$78.76^{\pm 2.52}$	$78.46^{\pm 2.54}$	$78.01^{\pm 2.88}$	$78.47^{\pm 3.51}$	$76.27^{\pm 3.89}$	$72.41^{\pm 6.35}$	$\mathbf{79.89}^{\pm 2.65}$	$77.94^{\pm 2.92}$
Tolokers	$35.79^{\pm 3.65}$	$36.08^{\pm 3.08}$	$\mathbf{36.15}^{\pm 3.16}$	$34.89^{\pm 2.74}$	$35.80^{\pm 2.50}$	$35.90^{\pm 2.66}$	$34.41^{\pm 2.45}$	$34.01^{\pm 2.08}$
Questions	$6.85^{\pm 2.07}$	$7.13^{\pm 2.10}$	$7.23^{\pm 2.44}$	$7.41^{\pm 2.43}$	$6.61^{\pm 1.73}$	$6.25^{\pm 1.31}$	$\mathbf{7.85}^{\pm 2.74}$	$4.50^{\pm 1.21}$
FDCompCN	$25.01^{\pm 3.76}$	$24.27^{\pm 4.47}$	$24.08^{\pm 4.78}$	$\mathbf{25.67}^{\pm 4.93}$	$24.05^{\pm 5.13}$	$24.36^{\pm 4.59}$	$24.36^{\pm 4.44}$	$22.67^{\pm 2.75}$
Injected benchmarks (legacy)								
Cora	$41.77^{\pm 8.68}$	$42.22^{\pm 8.81}$	$39.17^{\pm 9.19}$	$47.22^{\pm 5.40}$	$48.83^{\pm 5.02}$	$\mathbf{49.18}^{\pm 6.02}$	$40.43^{\pm 10.23}$	$42.09^{\pm 7.24}$
Citeseer	$42.97^{\pm 5.84}$	$42.16^{\pm 6.25}$	$42.05^{\pm 6.15}$	$\mathbf{50.28}^{\pm 5.12}$	$47.07^{\pm 5.32}$	$47.03^{\pm 5.53}$	$44.25^{\pm 6.91}$	$47.37^{\pm 4.13}$
BlogCatalog	$9.28^{\pm 2.35}$	$6.69^{\pm 1.23}$	$6.95^{\pm 1.27}$	$\mathbf{15.11}^{\pm 5.25}$	$9.71^{\pm 2.63}$	$8.65^{\pm 1.67}$	$6.22^{\pm 1.26}$	$12.04^{\pm 4.83}$
Flickr	$12.83^{\pm 3.40}$	$11.07^{\pm 2.65}$	$11.41^{\pm 2.61}$	$\mathbf{15.24}^{\pm 3.58}$	$11.13^{\pm 2.64}$	$12.00^{\pm 3.04}$	$10.10^{\pm 1.22}$	$12.54^{\pm 3.83}$
ACM	$10.39^{\pm 2.52}$	$9.41^{\pm 2.82}$	$9.00^{\pm 2.75}$	$12.84^{\pm 2.60}$	$12.69^{\pm 2.08}$	$\mathbf{13.51}^{\pm 2.32}$	$3.76^{\pm 1.24}$	$11.36^{\pm 2.84}$
Pubmed	$24.16^{\pm 7.75}$	$20.89^{\pm 7.70}$	$21.11^{\pm 7.74}$	$28.47^{\pm 6.04}$	$\mathbf{29.60}^{\pm 5.52}$	$29.45^{\pm 3.50}$	$20.10^{\pm 9.77}$	$25.25^{\pm 6.93}$
Injected benchmarks (new supports)								
Amazon-Photo	$19.17^{\pm 6.83}$	$18.61^{\pm 6.46}$	$18.11^{\pm 6.37}$	$\mathbf{26.46}^{\pm 7.28}$	$21.19^{\pm 5.08}$	$19.09^{\pm 4.53}$	$17.93^{\pm 5.40}$	$23.87^{\pm 7.26}$
Amazon-Computers	$11.94^{\pm 4.84}$	$12.83^{\pm 4.79}$	$12.39^{\pm 4.80}$	$\mathbf{17.81}^{\pm 5.79}$	$15.47^{\pm 4.97}$	$14.08^{\pm 4.23}$	$10.67^{\pm 3.20}$	$15.50^{\pm 5.26}$
Coauthor-CS	$21.36^{\pm 4.43}$	$22.80^{\pm 4.63}$	$22.05^{\pm 4.70}$	$20.21^{\pm 3.55}$	$21.98^{\pm 3.54}$	$\mathbf{22.91}^{\pm 4.06}$	$19.90^{\pm 3.18}$	$19.54^{\pm 2.26}$
Macro avg.	31.07	30.72	30.45	**33.92**	32.62	32.17	29.66	31.37

Table 9: **Absolute AUPRC of graph resampling with RFGRAPH** (Scarce_{20}, test AUPRC $\times 100$). Notation and seed counts follow Table 8.

Dataset	NR	CW	ROS	ROSE	SMOTE-Φ	ADASYN-Φ	Island	GODM
Organic anomalies (real labels)								
Reddit	$3.75^{\pm 0.32}$	$4.08^{\pm 0.45}$	$4.09^{\pm 0.46}$	$3.84^{\pm 0.50}$	$\mathbf{4.42}^{\pm 0.40}$	$4.30^{\pm 0.43}$	$3.93^{\pm 0.49}$	$3.64^{\pm 0.54}$
Weibo	$\mathbf{89.03}^{\pm 0.96}$	$80.87^{\pm 6.12}$	$81.96^{\pm 5.11}$	$87.86^{\pm 1.79}$	$85.45^{\pm 4.45}$	$85.00^{\pm 3.36}$	$74.30^{\pm 6.89}$	$88.63^{\pm 2.37}$
Amazon	$\mathbf{73.41}^{\pm 4.80}$	$69.95^{\pm 6.49}$	$71.06^{\pm 5.78}$	$70.96^{\pm 5.09}$	$70.18^{\pm 4.65}$	$69.94^{\pm 5.79}$	$73.14^{\pm 5.47}$	$65.00^{\pm 9.26}$
YelpChi	$22.63^{\pm 2.68}$	$22.32^{\pm 3.67}$	$22.86^{\pm 4.14}$	$\mathbf{27.28}^{\pm 4.34}$	$24.76^{\pm 5.27}$	$25.16^{\pm 5.54}$	$23.86^{\pm 3.48}$	$22.28^{\pm 1.95}$
T-Finance	$82.14^{\pm 2.25}$	$\mathbf{82.29}^{\pm 1.66}$	$81.77^{\pm 2.47}$	$79.45^{\pm 2.56}$	$80.52^{\pm 2.83}$	$78.98^{\pm 4.13}$	$81.03^{\pm 3.06}$	$78.67^{\pm 3.81}$
Tolokers	$36.09^{\pm 2.54}$	$\mathbf{36.61}^{\pm 1.94}$	$36.45^{\pm 2.14}$	$35.15^{\pm 2.44}$	$36.21^{\pm 1.49}$	$36.43^{\pm 1.66}$	$34.03^{\pm 1.90}$	$34.41^{\pm 2.22}$
Questions	$7.98^{\pm 2.16}$	$10.03^{\pm 3.02}$	$\mathbf{10.10}^{\pm 3.15}$	$8.64^{\pm 2.09}$	$9.43^{\pm 2.54}$	$9.62^{\pm 2.82}$	$10.00^{\pm 2.40}$	$7.20^{\pm 2.28}$
FDCompCN	$\mathbf{27.03}^{\pm 3.65}$	$25.99^{\pm 3.69}$	$25.66^{\pm 4.54}$	$26.86^{\pm 4.76}$	$25.33^{\pm 4.85}$	$25.17^{\pm 4.88}$	$25.85^{\pm 4.33}$	$25.86^{\pm 3.69}$
Injected benchmarks (legacy)								
Cora	$50.87^{\pm 5.69}$	$46.69^{\pm 6.37}$	$45.56^{\pm 7.50}$	$51.19^{\pm 5.26}$	$51.74^{\pm 4.27}$	$\mathbf{52.27}^{\pm 4.11}$	$50.08^{\pm 6.35}$	$50.20^{\pm 5.42}$
Citeseer	$50.93^{\pm 4.29}$	$50.63^{\pm 3.54}$	$48.82^{\pm 5.41}$	$50.87^{\pm 4.07}$	$49.79^{\pm 6.14}$	$50.05^{\pm 5.10}$	$\mathbf{51.93}^{\pm 4.28}$	$51.40^{\pm 4.34}$
BlogCatalog	$12.37^{\pm 3.31}$	$6.90^{\pm 1.32}$	$6.90^{\pm 1.15}$	$\mathbf{22.29}^{\pm 5.06}$	$9.62^{\pm 2.15}$	$10.37^{\pm 3.16}$	$7.31^{\pm 0.89}$	$13.43^{\pm 4.38}$
Flickr	$15.25^{\pm 4.08}$	$15.84^{\pm 4.72}$	$16.27^{\pm 5.32}$	$\mathbf{18.68}^{\pm 3.74}$	$14.14^{\pm 3.87}$	$14.95^{\pm 4.29}$	$10.15^{\pm 1.59}$	$15.16^{\pm 5.00}$
ACM	$19.54^{\pm 4.63}$	$16.73^{\pm 5.99}$	$18.49^{\pm 5.92}$	$20.10^{\pm 3.86}$	$18.93^{\pm 3.39}$	$19.56^{\pm 2.77}$	$3.30^{\pm 0.61}$	$\mathbf{20.66}^{\pm 5.50}$
Pubmed	$\mathbf{37.07}^{\pm 7.13}$	$34.54^{\pm 8.05}$	$32.92^{\pm 8.53}$	$34.91^{\pm 7.17}$	$36.36^{\pm 3.85}$	$36.42^{\pm 3.42}$	$32.12^{\pm 11.92}$	$35.09^{\pm 7.59}$
Injected benchmarks (new supports)								
Amazon-Photo	$19.26^{\pm 7.07}$	$20.12^{\pm 5.66}$	$19.77^{\pm 7.92}$	$\mathbf{31.40}^{\pm 4.66}$	$23.02^{\pm 4.94}$	$18.63^{\pm 4.65}$	$20.31^{\pm 8.05}$	$22.94^{\pm 10.58}$
Amazon-Computers	$14.20^{\pm 6.26}$	$13.37^{\pm 3.48}$	$14.08^{\pm 3.96}$	$\mathbf{21.25}^{\pm 5.63}$	$16.17^{\pm 4.25}$	$13.29^{\pm 2.31}$	$10.47^{\pm 4.03}$	$17.68^{\pm 7.98}$
Coauthor-CS	$22.23^{\pm 3.38}$	$27.73^{\pm 3.68}$	$\mathbf{30.31}^{\pm 5.15}$	$21.45^{\pm 2.95}$	$28.60^{\pm 3.14}$	$27.87^{\pm 3.64}$	$20.53^{\pm 6.21}$	$20.52^{\pm 2.67}$
Macro avg.	34.34	33.22	33.36	**36.01**	34.39	34.00	31.31	33.69

Table 10: **Absolute AUROC of graph resampling methods with XGBGRAPH** (Scarce_{20}, test AUROC $\times 100$, mean $\pm$ standard deviation over ten paired seeds). Best result in each row is shown in bold.

Dataset	No resampling	Class weighting	ROS	ROSE	SMOTE-Φ	Island view
Organic anomalies (real labels)						
Reddit	59.01 ± 1.67	59.77 ± 1.86	59.80 ± 1.68	59.45 ± 2.29	$\mathbf{60.48 \pm 1.80}$	59.34 ± 1.55
Weibo	96.99 ± 0.58	96.76 ± 0.78	96.86 ± 0.69	96.92 ± 0.69	$\mathbf{97.02 \pm 0.38}$	96.06 ± 1.04
Amazon	94.01 ± 1.89	94.31 ± 1.71	94.23 ± 1.64	95.19 ± 1.18	$\mathbf{95.41 \pm 1.41}$	94.14 ± 1.89
YelpChi	62.66 ± 4.56	62.72 ± 4.80	62.46 ± 4.65	$\mathbf{66.40 \pm 3.59}$	65.71 ± 5.44	65.58 ± 5.02
T-Finance	94.30 ± 1.19	94.26 ± 1.13	94.16 ± 1.04	94.25 ± 0.87	93.76 ± 1.72	$\mathbf{94.49 \pm 1.20}$
Tolokers	69.59 ± 3.36	$\mathbf{70.14 \pm 2.74}$	69.86 ± 2.59	69.38 ± 2.41	69.68 ± 2.35	68.90 ± 2.48
Questions	60.51 ± 3.07	60.88 ± 3.82	60.96 ± 3.87	59.79 ± 2.86	$\mathbf{62.58 \pm 2.69}$	60.33 ± 3.99
FDCompCN	71.86 ± 2.51	71.33 ± 3.54	71.23 ± 3.33	$\mathbf{72.03 \pm 3.15}$	71.32 ± 3.74	71.77 ± 2.86
Injected benchmarks (legacy)						
Cora	78.85 ± 2.97	78.83 ± 3.40	78.39 ± 3.44	79.93 ± 2.23	79.76 ± 2.16	$\mathbf{80.28 \pm 4.54}$
Citeseer	79.62 ± 4.71	78.99 ± 5.24	79.48 ± 4.86	$\mathbf{81.63 \pm 3.12}$	80.20 ± 3.51	80.79 ± 3.97
BlogCatalog	61.93 ± 4.67	57.25 ± 5.11	57.05 ± 5.18	$\mathbf{68.63 \pm 3.61}$	64.66 ± 4.59	57.80 ± 3.86
Flickr	62.10 ± 4.74	60.83 ± 3.75	61.32 ± 4.12	$\mathbf{66.44 \pm 3.55}$	62.44 ± 4.56	61.84 ± 3.24
ACM	66.05 ± 3.09	64.49 ± 5.07	64.45 ± 4.90	$\mathbf{69.22 \pm 2.98}$	68.25 ± 3.41	52.22 ± 3.29
Pubmed	73.13 ± 3.55	72.99 ± 3.92	73.50 ± 3.58	74.24 ± 2.36	$\mathbf{74.75 \pm 2.31}$	72.17 ± 4.77
Injected benchmarks (new supports)						
Amazon-Photo	68.60 ± 3.27	$\mathbf{70.95 \pm 3.52}$	70.61 ± 3.69	69.81 ± 4.28	69.84 ± 3.72	70.85 ± 2.89
Amazon-Computers	62.53 ± 5.49	64.86 ± 5.92	64.11 ± 5.21	66.65 ± 3.35	$\mathbf{67.02 \pm 2.42}$	64.83 ± 2.66
Coauthor-CS	79.48 ± 2.56	79.83 ± 2.68	79.20 ± 3.40	$\mathbf{81.08 \pm 2.63}$	80.70 ± 2.23	79.59 ± 2.21
Macro average	73.01	72.89	72.80	**74.77**	74.33	72.41

Table 11: **Absolute AUROC of graph resampling methods with RFGRAPH** (Scarce_{20}, test AUROC $\times 100$, mean $\pm$ standard deviation over ten paired seeds). Best result in each row is shown in bold.

Dataset	No resampling	Class weighting	ROS	ROSE	SMOTE-Φ	Island view
Organic anomalies (real labels)						
Reddit	59.31 ± 1.61	61.50 ± 2.06	61.19 ± 2.36	59.41 ± 2.19	$\mathbf{62.40 \pm 1.79}$	60.70 ± 2.52
Weibo	$\mathbf{98.30 \pm 0.26}$	97.26 ± 0.87	97.38 ± 0.76	97.82 ± 0.64	97.84 ± 0.69	96.42 ± 0.98
Amazon	93.25 ± 1.59	92.70 ± 1.73	92.76 ± 1.62	$\mathbf{93.43 \pm 1.15}$	93.14 ± 1.36	93.03 ± 1.76
YelpChi	62.39 ± 2.78	61.60 ± 3.62	62.55 ± 4.37	$\mathbf{66.19 \pm 2.87}$	64.82 ± 4.95	63.90 ± 3.60
T-Finance	94.79 ± 0.76	$\mathbf{94.92 \pm 0.59}$	94.75 ± 0.94	94.40 ± 1.05	94.54 ± 1.14	94.66 ± 0.83
Tolokers	70.71 ± 1.96	$\mathbf{71.68 \pm 1.88}$	71.20 ± 1.63	69.99 ± 2.55	71.11 ± 1.58	69.64 ± 2.03
Questions	59.59 ± 2.22	65.37 ± 3.91	65.52 ± 3.52	58.25 ± 1.96	$\mathbf{66.71 \pm 1.84}$	61.87 ± 3.62
FDCompCN	71.75 ± 2.13	72.83 ± 2.38	72.37 ± 2.72	71.58 ± 2.72	$\mathbf{73.25 \pm 2.96}$	72.68 ± 2.34
Injected benchmarks (legacy)						
Cora	81.08 ± 1.66	80.18 ± 2.37	80.35 ± 2.28	81.33 ± 1.30	81.25 ± 2.29	$\mathbf{83.03 \pm 2.74}$
Citeseer	81.66 ± 2.20	81.03 ± 1.74	81.86 ± 1.99	81.63 ± 2.20	81.05 ± 2.28	$\mathbf{83.95 \pm 2.24}$
BlogCatalog	66.02 ± 1.50	58.41 ± 2.61	58.50 ± 2.34	$\mathbf{73.01 \pm 1.80}$	66.49 ± 2.50	62.17 ± 2.36
Flickr	67.93 ± 3.30	63.53 ± 4.31	64.20 ± 5.10	$\mathbf{71.52 \pm 2.29}$	67.56 ± 4.48	65.60 ± 3.09
ACM	71.65 ± 2.38	68.14 ± 4.89	69.12 ± 4.85	$\mathbf{73.56 \pm 1.39}$	72.84 ± 0.95	50.02 ± 3.80
Pubmed	$\mathbf{77.33 \pm 1.73}$	76.31 ± 1.27	75.93 ± 1.99	75.86 ± 1.23	76.49 ± 0.94	76.24 ± 5.04
Injected benchmarks (new supports)						
Amazon-Photo	65.90 ± 3.67	$\mathbf{74.22 \pm 3.01}$	73.24 ± 3.30	67.59 ± 2.77	73.43 ± 3.94	72.99 ± 3.32
Amazon-Computers	62.14 ± 3.81	69.50 ± 4.80	69.39 ± 4.80	64.80 ± 3.29	$\mathbf{70.14 \pm 3.81}$	62.08 ± 5.19
Coauthor-CS	81.44 ± 2.13	81.21 ± 1.10	81.69 ± 2.15	82.72 ± 1.49	$\mathbf{82.97 \pm 1.63}$	77.28 ± 5.77
Macro average	74.43	74.73	74.82	75.48	**76.24**	73.31

Table 12: **Absolute Recall@K of graph resampling methods with XGBGRAPH** (Scarce_{20}, test Recall@K (K denotes the number of anomalies in the test set; $\times 100$), mean $\pm$ standard deviation over ten paired seeds). Best result in each row is shown in bold.

Dataset	No resampling	Class weighting	ROS	ROSE	SMOTE-Φ	Island view
Organic anomalies (real labels)						
Reddit	4.56 ± 1.22	5.17 ± 1.27	$\mathbf{5.47 \pm 1.40}$	4.73 ± 1.49	5.24 ± 1.51	4.53 ± 1.28
Weibo	71.45 ± 5.87	70.96 ± 5.74	72.04 ± 4.97	$\mathbf{74.91 \pm 2.42}$	73.52 ± 3.72	64.62 ± 6.06
Amazon	76.15 ± 5.11	77.32 ± 3.08	77.59 ± 3.25	76.98 ± 3.71	$\mathbf{78.96 \pm 3.09}$	76.79 ± 4.11
YelpChi	25.67 ± 4.61	25.98 ± 4.72	25.98 ± 4.98	$\mathbf{30.27 \pm 4.09}$	28.76 ± 5.39	27.88 ± 4.99
T-Finance	73.06 ± 2.32	72.56 ± 3.13	72.19 ± 3.37	73.26 ± 2.24	71.11 ± 3.49	$\mathbf{74.39 \pm 2.60}$
Tolokers	37.99 ± 3.98	$\mathbf{38.51 \pm 3.25}$	38.39 ± 3.43	37.09 ± 2.95	37.97 ± 2.57	37.18 ± 3.03
Questions	10.03 ± 2.51	9.64 ± 3.02	9.75 ± 3.08	10.53 ± 3.18	9.95 ± 3.27	$\mathbf{10.90 \pm 2.21}$
FDCompCN	31.19 ± 5.28	30.33 ± 6.78	30.20 ± 6.77	$\mathbf{31.84 \pm 5.34}$	29.53 ± 7.27	29.90 ± 5.70
Injected benchmarks (legacy)						
Cora	43.62 ± 7.23	43.50 ± 7.24	42.50 ± 7.95	47.88 ± 6.10	$\mathbf{48.50 \pm 6.53}$	40.00 ± 8.90
Citeseer	42.00 ± 6.04	41.38 ± 6.88	41.25 ± 6.43	$\mathbf{48.38 \pm 5.71}$	45.12 ± 4.77	41.25 ± 7.34
BlogCatalog	12.78 ± 3.79	9.04 ± 2.03	8.35 ± 2.52	$\mathbf{20.04 \pm 6.09}$	13.04 ± 4.04	7.22 ± 3.47
Flickr	15.79 ± 4.37	13.82 ± 4.37	14.47 ± 3.95	$\mathbf{18.03 \pm 4.47}$	15.05 ± 4.09	15.05 ± 1.77
ACM	15.01 ± 3.72	14.65 ± 4.36	14.00 ± 4.37	$\mathbf{19.01 \pm 4.34}$	17.15 ± 3.05	3.21 ± 2.03
Pubmed	27.25 ± 7.77	26.09 ± 8.84	25.96 ± 8.47	32.85 ± 5.55	$\mathbf{33.57 \pm 4.48}$	23.15 ± 9.42
Injected benchmarks (new supports)						
Amazon-Photo	23.65 ± 7.54	23.27 ± 7.30	22.95 ± 7.11	$\mathbf{30.29 \pm 5.93}$	27.08 ± 4.91	24.26 ± 5.36
Amazon-Computers	17.12 ± 8.01	17.85 ± 6.78	18.19 ± 7.09	$\mathbf{25.66 \pm 6.01}$	22.07 ± 6.50	16.44 ± 5.79
Coauthor-CS	25.14 ± 4.33	$\mathbf{26.56 \pm 3.78}$	25.89 ± 4.99	25.04 ± 3.64	25.66 ± 3.79	24.09 ± 3.89
Macro average	32.50	32.16	32.07	**35.69**	34.25	30.64

Table 13: **Absolute Recall@K of graph resampling methods with RFGRAPH** (Scarce_{20}, test Recall@K (K denotes the number of anomalies in the test set; $\times 100$), mean $\pm$ standard deviation over ten paired seeds). Best result in each row is shown in bold.

Dataset	No resampling	Class weighting	ROS	ROSE	SMOTE-Φ	Island view
Organic anomalies (real labels)						
Reddit	4.66 ± 1.61	5.51 ± 1.58	5.88 ± 1.51	4.90 ± 1.93	$\mathbf{6.39 \pm 1.40}$	5.10 ± 1.83
Weibo	$\mathbf{81.25 \pm 1.91}$	71.40 ± 6.51	72.87 ± 6.47	80.94 ± 1.89	77.08 ± 6.37	65.30 ± 7.16
Amazon	66.67 ± 3.89	63.02 ± 5.01	63.54 ± 4.02	63.74 ± 4.41	63.54 ± 3.41	$\mathbf{66.95 \pm 3.91}$
YelpChi	24.38 ± 2.81	24.34 ± 3.41	24.82 ± 4.51	$\mathbf{29.22 \pm 4.36}$	27.56 ± 5.32	25.67 ± 3.83
T-Finance	$\mathbf{76.55 \pm 2.80}$	76.26 ± 2.17	76.15 ± 2.37	73.95 ± 2.54	75.16 ± 2.55	75.71 ± 3.32
Tolokers	38.37 ± 3.27	$\mathbf{39.03 \pm 2.20}$	38.65 ± 2.52	37.74 ± 2.76	37.86 ± 1.90	37.00 ± 2.70
Questions	10.24 ± 1.82	$\mathbf{13.68 \pm 3.06}$	13.39 ± 4.26	10.64 ± 1.65	13.52 ± 2.50	12.03 ± 2.98
FDCompCN	$\mathbf{33.44 \pm 4.41}$	32.35 ± 4.69	31.78 ± 5.71	33.09 ± 4.82	30.39 ± 5.92	31.64 ± 5.43
Injected benchmarks (legacy)						
Cora	48.25 ± 6.38	46.62 ± 6.62	46.62 ± 6.62	49.12 ± 5.37	$\mathbf{50.12 \pm 5.35}$	48.50 ± 6.97
Citeseer	47.88 ± 4.75	47.50 ± 4.68	45.62 ± 5.93	48.25 ± 4.46	46.12 ± 5.91	$\mathbf{48.38 \pm 4.29}$
BlogCatalog	16.61 ± 4.35	9.52 ± 3.52	8.52 ± 2.15	$\mathbf{26.61 \pm 7.45}$	13.96 ± 3.67	9.87 ± 2.57
Flickr	18.71 ± 4.60	17.63 ± 5.73	18.95 ± 6.70	$\mathbf{21.34 \pm 4.35}$	17.39 ± 4.79	15.55 ± 3.24
ACM	21.06 ± 4.49	19.72 ± 5.95	21.56 ± 4.75	$\mathbf{23.42 \pm 4.12}$	22.14 ± 4.80	2.24 ± 2.11
Pubmed	40.00 ± 6.01	38.66 ± 7.76	36.83 ± 8.46	40.17 ± 7.35	$\mathbf{42.43 \pm 3.95}$	33.98 ± 11.92
Injected benchmarks (new supports)						
Amazon-Photo	23.88 ± 7.65	26.19 ± 6.28	25.54 ± 8.52	$\mathbf{34.90 \pm 2.82}$	28.94 ± 4.04	26.22 ± 8.78
Amazon-Computers	20.10 ± 8.07	19.77 ± 5.67	20.55 ± 5.29	$\mathbf{28.93 \pm 5.87}$	23.43 ± 5.76	17.06 ± 8.80
Coauthor-CS	23.97 ± 4.32	29.92 ± 3.60	$\mathbf{32.30 \pm 5.27}$	24.94 ± 3.54	29.79 ± 3.80	24.04 ± 5.68
Macro average	35.06	34.18	34.33	**37.17**	35.64	32.07

Table 14: **Absolute AUPRC of graph resampling methods with XGBGRAPH under full supervision.** Official GADBench training masks (FullSup), test AUPRC ($\times 100$), mean $\pm$ standard deviation over ten paired seeds. FullSup uses its own train/test split and should not be compared directly with Scarce_k. Best result in each row is shown in bold. Zero standard deviations on repeated masks are explained in Appendix B.2.

Dataset	No resampling	Class weighting	ROS	ROSE	SMOTE-Φ	Island view
Organic anomalies (real labels)						
Reddit	6.49 ± 0.34	5.61 ± 0.54	5.60 ± 0.50	6.24 ± 0.65	$\mathbf{6.81 \pm 0.77}$	6.08 ± 0.66
Weibo	97.49 ± 0.74	97.55 ± 0.59	97.48 ± 0.56	$\mathbf{97.67 \pm 0.57}$	97.05 ± 0.73	97.63 ± 0.35
Amazon	$\mathbf{92.93 \pm 0.00}$	92.61 ± 0.00	92.70 ± 0.26	92.89 ± 0.34	92.25 ± 0.41	92.62 ± 0.37
YelpChi	$\mathbf{88.29 \pm 0.00}$	87.25 ± 0.00	86.79 ± 0.38	87.42 ± 0.38	85.82 ± 0.48	86.78 ± 0.48
T-Finance	89.54 ± 0.98	$\mathbf{89.58 \pm 0.95}$	89.54 ± 0.97	88.96 ± 0.93	89.11 ± 1.06	88.64 ± 0.95
Tolokers	49.53 ± 1.44	49.06 ± 1.22	48.57 ± 2.22	$\mathbf{49.68 \pm 1.75}$	47.58 ± 1.73	44.00 ± 1.19
Questions	18.19 ± 1.40	15.28 ± 0.71	15.66 ± 1.13	$\mathbf{18.55 \pm 1.19}$	17.10 ± 0.90	15.49 ± 0.93
FDCompCN	$\mathbf{39.96 \pm 3.06}$	39.01 ± 2.29	39.18 ± 2.46	39.63 ± 2.95	37.68 ± 3.01	38.85 ± 2.17
Injected benchmarks (legacy)						
Cora	58.36 ± 3.76	58.46 ± 3.57	58.49 ± 3.77	$\mathbf{60.43 \pm 3.65}$	59.54 ± 3.26	59.76 ± 4.39
Citeseer	62.80 ± 6.29	61.90 ± 5.98	61.93 ± 5.89	65.43 ± 6.03	63.51 ± 5.57	$\mathbf{65.51 \pm 3.91}$
BlogCatalog	64.66 ± 5.33	57.26 ± 5.46	58.68 ± 5.88	$\mathbf{66.05 \pm 4.85}$	59.35 ± 5.72	58.60 ± 5.21
Flickr	71.18 ± 2.03	66.41 ± 2.03	66.45 ± 1.77	$\mathbf{74.17 \pm 2.32}$	70.50 ± 2.52	62.23 ± 3.09
ACM	56.04 ± 2.28	54.34 ± 2.62	54.67 ± 1.77	$\mathbf{58.37 \pm 2.39}$	55.92 ± 2.52	40.00 ± 2.59
Pubmed	69.77 ± 2.95	67.44 ± 2.42	67.66 ± 2.36	$\mathbf{70.09 \pm 2.44}$	66.84 ± 2.66	68.16 ± 2.92
Injected benchmarks (new supports)						
Amazon-Photo	72.65 ± 2.65	$\mathbf{73.66 \pm 3.93}$	73.45 ± 4.22	69.60 ± 4.26	70.81 ± 4.39	63.15 ± 3.79
Amazon-Computers	$\mathbf{67.72 \pm 1.91}$	67.63 ± 2.45	66.96 ± 2.26	66.86 ± 1.87	66.06 ± 1.26	53.94 ± 2.66
Coauthor-CS	$\mathbf{93.39 \pm 1.07}$	91.91 ± 1.20	92.18 ± 1.02	93.27 ± 1.46	90.06 ± 1.38	87.57 ± 1.63
Macro average	64.65	63.23	63.29	**65.02**	63.29	60.53

Table 15: **Absolute AUPRC of graph resampling methods with RFGRAPH under full supervision.** Official GADBench training masks (FullSup), test AUPRC ($\times 100$), mean $\pm$ standard deviation over ten paired seeds. FullSup uses its own train/test split and should not be compared directly with Scarce_k. Best result in each row is shown in bold.

Dataset	No resampling	Class weighting	ROS	ROSE	SMOTE-Φ	Island view
Organic anomalies (real labels)						
Reddit	6.23 ± 0.54	5.42 ± 0.44	5.75 ± 0.31	5.45 ± 0.53	$\mathbf{6.80 \pm 1.13}$	5.55 ± 0.34
Weibo	$\mathbf{97.26 \pm 0.31}$	96.98 ± 0.27	97.12 ± 0.35	96.67 ± 0.41	96.86 ± 0.40	97.21 ± 0.43
Amazon	90.74 ± 0.33	91.29 ± 0.14	$\mathbf{91.73 \pm 0.32}$	91.30 ± 0.18	91.08 ± 0.42	89.79 ± 0.33
YelpChi	76.57 ± 0.44	76.57 ± 0.30	$\mathbf{77.73 \pm 0.35}$	76.38 ± 0.37	75.45 ± 0.38	72.14 ± 0.47
T-Finance	88.91 ± 1.00	89.00 ± 0.95	$\mathbf{89.20 \pm 0.95}$	87.42 ± 1.04	88.98 ± 0.99	87.19 ± 1.01
Tolokers	$\mathbf{51.07 \pm 1.99}$	49.36 ± 1.72	49.07 ± 1.93	50.91 ± 2.08	48.37 ± 1.70	42.14 ± 1.73
Questions	$\mathbf{15.36 \pm 0.88}$	14.23 ± 1.25	12.87 ± 1.29	14.05 ± 0.95	15.35 ± 1.25	12.75 ± 1.21
FDCompCN	$\mathbf{39.90 \pm 2.59}$	38.99 ± 2.36	39.06 ± 3.17	39.57 ± 2.75	35.33 ± 2.45	38.56 ± 2.53
Injected benchmarks (legacy)						
Cora	$\mathbf{59.91 \pm 3.84}$	58.00 ± 3.48	57.28 ± 3.70	59.84 ± 3.80	59.69 ± 3.86	59.05 ± 4.19
Citeseer	62.12 ± 4.02	60.35 ± 5.07	61.06 ± 3.83	61.91 ± 4.49	61.72 ± 4.90	$\mathbf{63.94 \pm 3.57}$
BlogCatalog	$\mathbf{59.34 \pm 5.76}$	43.27 ± 7.51	51.28 ± 7.80	57.43 ± 6.30	50.44 ± 5.66	55.05 ± 5.89
Flickr	73.90 ± 1.59	61.76 ± 4.09	69.88 ± 2.67	73.09 ± 2.16	$\mathbf{74.03 \pm 2.28}$	68.21 ± 2.13
ACM	65.12 ± 2.87	54.21 ± 2.35	54.85 ± 2.11	$\mathbf{65.83 \pm 2.51}$	61.86 ± 2.59	29.14 ± 3.29
Pubmed	70.03 ± 2.83	60.09 ± 2.42	63.10 ± 2.58	$\mathbf{70.22 \pm 2.78}$	61.47 ± 1.82	64.47 ± 2.56
Injected benchmarks (new supports)						
Amazon-Photo	53.85 ± 3.94	69.18 ± 4.04	$\mathbf{71.34 \pm 3.44}$	54.06 ± 4.21	57.95 ± 5.01	52.31 ± 4.75
Amazon-Computers	48.07 ± 2.39	55.73 ± 2.70	$\mathbf{59.10 \pm 2.94}$	47.70 ± 2.40	48.18 ± 2.29	33.68 ± 1.15
Coauthor-CS	$\mathbf{92.10 \pm 1.29}$	81.97 ± 1.04	83.63 ± 1.60	91.07 ± 1.65	80.44 ± 1.59	69.18 ± 2.83
Macro average	**61.79**	59.20	60.83	61.35	59.65	55.31

C Paired Comparisons Against Both References

Table 16: **Resampling against two references (Scarce_{20}).** Test AUPRC and paired gains (all $\times 100$), averaging seeds then datasets. CW: class weighting; NR: unweighted training without resampling. Brackets give hierarchical-bootstrap 95% CIs (20,000 replicates). Bold gains have intervals excluding zero.

	Organic (8)			Injected (9)		
Method	AUPRC	Δ vs CW	Δ vs NR	AUPRC	Δ vs CW	Δ vs NR
XGBGraph: 100 trees, $\eta = 0.3$						
No resampling	41.78	-0.16 $[-0.80, +0.41]$	—	21.54	$+0.80$ $[-0.27, +1.87]$	—
Class weighting	41.94	—	$+0.16$ $[-0.40, +0.79]$	20.74	—	-0.80 $[-1.89, +0.28]$
ROS	41.92	-0.02 $[-0.32, +0.33]$	$+0.14$ $[-0.52, +0.86]$	20.25	-0.49 $[-1.34, +0.19]$	$\mathbf{-1.29}$ $[-2.25, -0.32]$
ROSE	42.88	$+0.94$ $[-0.34, +2.45]$	$+1.10$ $[-0.14, +2.54]$	25.96	$\mathbf{+5.22}$ $[+2.68, +7.43]$	$\mathbf{+4.42}$ $[+2.41, +6.31]$
SMOTE-Φ	42.10	$+0.16$ $[-0.83, +1.20]$	$+0.32$ $[-0.88, +1.49]$	24.18	$\mathbf{+3.44}$ $[+1.47, +5.58]$	$\mathbf{+2.64}$ $[+0.83, +4.62]$
GraphSMOTE-inspired	40.18	-1.76 $[-4.26, +0.65]$	-1.60 $[-4.20, +0.82]$	23.10	$\mathbf{+2.36}$ $[+0.46, +4.17]$	$+1.56$ $[-0.56, +3.59]$
ADASYN-Φ	41.38	-0.55 $[-2.38, +0.81]$	-0.40 $[-2.40, +1.16]$	23.99	$\mathbf{+3.25}$ $[+1.33, +5.45]$	$\mathbf{+2.44}$ $[+0.62, +4.51]$
Island views	41.36	-0.58 $[-2.52, +0.84]$	-0.43 $[-2.45, +1.03]$	19.26	-1.48 $[-2.97, +0.03]$	$\mathbf{-2.28}$ $[-3.85, -0.71]$
GODM†	40.47	-1.47 $[-4.18, +0.73]$	-1.31 $[-3.63, +0.71]$	23.28	$\mathbf{+2.58}$ $[+0.57, +4.42]$	$\mathbf{+1.78}$ $[+0.28, +3.31]$
GOSS-inspired	39.77	$\mathbf{-2.17}$ $[-3.56, -0.93]$	$\mathbf{-2.01}$ $[-3.51, -0.69]$	16.86	$\mathbf{-3.88}$ $[-5.69, -2.29]$	$\mathbf{-4.68}$ $[-6.21, -3.24]$
Residual SMOTE-Φ	41.83	-0.11 $[-0.64, +0.39]$	$+0.05$ $[-0.85, +0.94]$	21.74	$\mathbf{+1.00}$ $[+0.53, +1.56]$	$+0.20$ $[-1.13, +1.57]$
RFGraph						
No resampling	42.76	$+1.24$ $[-0.54, +3.66]$	—	26.86	$+1.02$ $[-1.18, +3.05]$	—
Class weighting	41.52	—	-1.24 $[-3.65, +0.53]$	25.84	—	-1.02 $[-3.04, +1.18]$
ROS	41.74	$+0.23$ $[-0.23, +0.74]$	-1.01 $[-3.09, +0.55]$	25.90	$+0.06$ $[-1.03, +1.17]$	-0.96 $[-3.45, +1.93]$
ROSE	42.51	$+0.99$ $[-1.06, +3.47]$	-0.25 $[-1.63, +1.43]$	30.24	$\mathbf{+4.40}$ $[+0.48, +8.55]$	$\mathbf{+3.38}$ $[+0.41, +6.80]$
SMOTE-Φ	42.04	$+0.52$ $[-0.76, +2.11]$	-0.72 $[-2.27, +0.74]$	27.60	$\mathbf{+1.76}$ $[+0.22, +3.24]$	$+0.74$ $[-1.07, +2.77]$
GraphSMOTE-inspired	39.45	-2.07 $[-5.68, +1.65]$	$\mathbf{-3.31}$ $[-6.82, -0.40]$	25.62	-0.21 $[-2.64, +2.28]$	-1.23 $[-4.08, +1.95]$
ADASYN-Φ	41.83	$+0.31$ $[-1.23, +1.94]$	-0.93 $[-2.72, +0.78]$	27.05	$+1.21$ $[-0.37, +2.96]$	$+0.19$ $[-1.28, +1.92]$
Island views	40.77	-0.75 $[-2.83, +1.07]$	-1.99 $[-6.03, +0.57]$	22.91	-2.93 $[-6.43, +0.15]$	$\mathbf{-3.95}$ $[-7.64, -1.03]$
GODM†	40.71	-0.81 $[-3.12, +2.12]$	$\mathbf{-2.05}$ $[-4.20, -0.57]$	27.45	$+1.64$ $[-1.09, +4.06]$	$+0.61$ $[-0.72, +2.12]$
XGBGraph: 200 trees, $\eta = 0.05$						
No resampling	42.26	$+0.19$ $[-0.39, +0.78]$	—	22.00	$\mathbf{+0.93}$ $[+0.12, +1.80]$	—
Class weighting	42.07	—	-0.19 $[-0.78, +0.39]$	21.07	—	$\mathbf{-0.93}$ $[-1.80, -0.12]$
ROSE	43.39	$+1.32$ $[-0.06, +2.86]$	$+1.13$ $[-0.06, +2.47]$	26.59	$\mathbf{+5.53}$ $[+2.72, +7.91]$	$\mathbf{+4.60}$ $[+2.15, +6.80]$
Residual SMOTE-Φ	42.54	$\mathbf{+0.46}$ $[+0.06, +0.86]$	$+0.27$ $[-0.30, +0.95]$	22.40	$\mathbf{+1.33}$ $[+0.83, +1.94]$	$+0.40$ $[-0.62, +1.42]$

Contrasts use paired splits and each configuration's own references (Appendix A.1). GraphSMOTE-inspired and GOSS-inspired are local adaptations, not evaluations of the original methods. GOSS adds a 30-tree pilot to its 100-tree fit, so total training cost is not matched. †GODM omits failed Flickr seed 2 from both sides (169 trials per learner); its paired reference differs slightly from the complete row.

D AI USE STATEMENT

Generative AI tools were used to assist with drafting and editing the manuscript to improve its clarity and readability. The authors take full responsibility for the final content of the paper, including all text and claims produced with AI assistance.